9th Innovations in Theoretical Computer Science

ITCS 2018, January 11–14, 2018, Cambridge, MA, USA

Edited by

Anna R. Karlin

Editor

Anna R. Karlin
Allen School of Computer Science and Engineering
University of Washington
karlin@cs.washington.edu

ACM Classification 1998
F. Theory of Computation, G. Mathematics of Computing

ISBN 978-3-95977-060-6

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/978-3-95977-060-6.

Publication date
January, 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.

Digital Object Identifier: 10.4230/LIPIcs.ITCS.2018.0

LIPIcs – Leibniz International Proceedings in Informatics

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

ISSN 1868-8969

http://www.dagstuhl.de/lipics

Contents

9th Innovations in Theoretical Computer Science (ITCS 2018).
Editor: Anna R. Karlin
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Preface

The papers in this volume were presented at the 9th Innovations in Theoretical Computer Science (ITCS 2018) conference. The conference was held at the Massachusetts Institute of Technology in Cambridge, MA, USA, January 11-14, 2018. ITCS seeks to promote research that carries a strong conceptual message, for instance, introducing a new concept or model, opening a new line of inquiry within traditional or cross-interdisciplinary areas, introducing new techniques, or making novel connections between existing areas and ideas. The conference format is single-session with ample time for discussion, to promote the exchange of ideas between different areas of theoretical computer science and with other disciplines. The call for papers welcomed all submissions, whether aligned with current theory of computation research directions or deviating from them. 181 submissions were received. Of these, the program committee selected 59 papers. I would like to thank the authors of all submissions for their interest in ITCS.

The program committee consisted of 35 members (plus the chair): Shipra Agarwal, Columbia; Zeyuan Allen-Zhu, MSR; Benny Applebaum, Tel Aviv; Paul Beame, U. of Washington; Karl Bringmann, MPI; Bernard Chazelle, Princeton; Jing Chen, Stony Brook; Rachel Cummings, Georgia Tech; Andrew Drucker, U. of Chicago; Faith Ellen, U. of Toronto; Kousha Etessami, U. of Edinburgh; Oded Goldreich, Weizmann; Anupam Gupta, CMU; Zhiyi Huang, U. of Hong Kong; Christian Ikenmeyer, MPI; Yael Kalai, MSR; Robert Kleinberg, Cornell; Tengyu Ma, Stanford; Yury Makarychev, TTIC; Ruta Mehta, UIUC; Raghu Meka, UCLA; Ashley Montanaro, Bristol; Shayan Oveis Gharan, U. of Washington; Christos Papadimitriou, Columbia; Seth Pettie, Michigan; Ronitt Rubinfeld, MIT and Tel Aviv; Atri Rudra, SUNY Buffalo; C. Seshadhri, UC Santa Cruz; Tselil Schramm, Harvard; Roy Schwartz, Technion; Li-Yang Tan, TTIC; Greg Valiant, Stanford; John Watrous, Waterloo; David Woodruff, CMU; and Yuan Zhou, Indiana. I wish to express my sincere thanks to them for agreeing to join the committee and then for investing a great deal of time and effort to evaluate the submissions. I am also grateful to the many subreviewers who assisted with the reviewing process.

The local organizers were Costis Daskalakis (MIT), Yael Kalai (Microsoft Research New England), and Vinod Vaikuntanathan, (MIT). I'd like to thank them profusely for their service. I'm also grateful to Umesh Vazirani, chair of the ITCS Steering Committee, who helped me throughout the process, to Thomas Vidick, who helped with the website among other things, and to Oded Goldreich, Christos Papadimitriou and Madhu Sudan for advice at various stages in the process. Finally, I would like to thank all the presenters and the audience at ITCS; I hope it continues to be a unique experience.

Anna R. Karlin
ITCS 2018 Program Chair
University of Washington
Seattle, WA USA

9th Innovations in Theoretical Computer Science (ITCS 2018).
Editor: Anna R. Karlin
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

ITCS 2018 Conference Organization

Program Chair: Anna Karlin (*University of Washington*)

Local Organization: Costis Daskalakis (*MIT*)
Yael Kalai (*Microsoft Research New England*)
Vinod Vaikuntanathan (*MIT*)

Steering Committee Chair: Umesh Vazirani (*UC Berkeley*)

Steering Committee Sanjeev Arora, Princeton
Manuel Blum, Carnegie Mellon
Bernard Chazelle, Princeton
Irit Dinur, Weizmann
Oded Goldreich, Weizmann
Shafi Goldwasser, MIT and Weizmann
Richard Karp, UC Berkeley
Robert Kleinberg, Cornell University
Ueli Maurer, ETH
Silvio Micali, MIT
Christos Papadimitriou, Columbia
Michael Rabin, Harvard
Omer Reingold, Stanford
Tim Roughgarden, Stanford
Madhu Sudan, Harvard
Leslie Valiant, Harvard
Umesh Vazirani, Berkeley
Thomas Vidick, Caltech
Avi Wigderson, IAS
Andy Yao, Tsinghua

Program Committee: Shipra Agarwal, Columbia University
Zeyuan Allen-Zhu, Microsoft Research
Benny Applebaum, Tel Aviv University
Paul Beame, University of Washington
Karl Bringmann, Max Planck Institute
Bernard Chazelle, Princeton University
Jing Chen, Stony Brook University
Rachel Cummings, Georgia Tech
Andrew Drucker, University of Chicago
Faith Ellen, University of Toronto
Kousha Etessami, University of Edinburgh
Oded Goldreich, Weizmann Institute
Anupam Gupta, Carnegie Mellon University
Zhiyi Huang, University of Hong Kong

Program Committee (continued):

Christian Ikenmeyer, Max Planck Institute
Yael Kalai, Microsoft Research
Anna Karlin, University of Washington
Robert Kleinberg, Cornell University
Tengyu Ma, Stanford
Yury Makarychev, TTIC
Ruta Mehta, UIUC
Raghu Meka, UCLA
Ashley Montanaro, University of Bristol
Shayan Oveis Gharan, University of Washington
Christos Papadimitriou, Columbia
Seth Pettie, University of Michigan
Ronitt Rubinfeld, MIT and Tel Aviv University
Atri Rudra, University at Buffalo, SUNY
C. Seshadhri, UC Santa Cruz
Tselil Schramm, Harvard
Roy Schwartz, Technion
Li-Yang Tan, TTIC
Greg Valiant, Stanford University
John Watrous, University of Waterloo
David Woodruff, Carnegie Mellon University
Yuan Zhou, Indiana University

Additional Reviewers:

Jake Abernethy
Yuqing Ai
Maryam Aliakbarpour
James Aspnes
Pablo Azar
Amos Beimel
Umang Bhaskar
Eric Blais
Ilario Bonacina
Simina Branzei
Jonah Brown-Cohen
Yang Cai
Clement Canonne
Deeparnab Chakrabarty
Eshan Chattopadhyay
Chandra Chekuri
Xi Chen
Richard Cole
Amit Daniely
Ilias Diakonikolas
Bill Fefferman
Amos Fiat
Weihao Kong
Massimo Lauria
James Lee
Yuanzhi Li
Yingyu Liang
Jinyan Liu
Florian Lonsing
Jieming Mao
Ilya Mironov
Cameron Musco
Christopher Musco
Rad Niazadeh
Kobbi Nissim
Alex Olshevsky
Omer Paneth
Yuri Polyanskiy
Christos-Alexandros Psomas
Aditi Raghunathan
Ran Raz
Ilya Razenshteyn
Alireza Rezaei
David Richerby

Additional Reviewers (continued):

Yuval Filmus
Ankit Garg
Vasilis Gkatzelis
Parikshit Gopalan
Mayank Goswami
Joshua Grochow
Nima Haghpanah
Nika Haghtalab
Mohammadtaghi Hajiaghayi
Moritz Hardt
Johan Hastad
Justin Holmgren
Justin Hsu
Fotis Iliopoulos
Ziwei Ji
Frank Kammer
Daniel Kane
Ian Kash
Aggelos Kiayias
Masashi Kiyomi
Saleet Klein
Pavel Kolev
Swastik Kopparty
Pravesh Kothari
Ron Rothblum
He Sun
Xiaoming Sun
Madhur Tulsiani
Jonathan Ullman
Ruth Urner
Peter Van Emde Boas
Santosh Vempala
Thomas Vidick
Emanuele Viola
Adrian Vladu
Mor Weiss
Steven Wu
Xiaowei Wu
Haifeng Xu
Yang Yuan
Hongyang Zhang
Qin Zhang
Yuhao Zhang
Mingfei Zhao
Peilin Zhong
Xue Zhu

Barriers for Rank Methods in Arithmetic Complexity*†

Klim Efremenko[1], Ankit Garg[2], Rafael Oliveira[3], and Avi Wigderson[4]

1 **Ben Gurion University of the Negev, Beer-Sheva, Israel**
klimefrem@gmail.com
2 **Microsoft Research New England, 1 Memorial Dr, Cambridge, MA, USA**
garga@microsoft.com
3 **Department of Computer Science, University of Toronto, 10 King's College Road, Toronto, Canada**
rafael@cs.toronto.edu
4 **Institute for Advanced Study, 1 Einstein Dr, Princeton, USA**
avi@math.ias.edu

Abstract

Arithmetic complexity, the study of the cost of computing polynomials via additions and multiplications, is considered (for many good reasons) simpler to understand than *Boolean complexity*, namely computing Boolean functions via logical gates. And indeed, we seem to have significantly more lower bound techniques and results in arithmetic complexity than in Boolean complexity. Despite many successes and rapid progress, however, foundational challenges, like proving super-polynomial lower bounds on circuit or formula size for explicit polynomials, or super-linear lower bounds on explicit 3-dimensional tensors, remain elusive.

At the same time (and possibly for similar reasons), we have plenty more excuses, in the form of "barrier results" for failing to prove basic lower bounds in Boolean complexity than in arithmetic complexity. Efforts to find barriers to arithmetic lower bound techniques seem harder, and despite some attempts we have no excuses of similar quality for these failures in arithmetic complexity. This paper aims to add to this study.

In this paper we address *rank methods*, which were long recognized as encompassing and abstracting almost all known arithmetic lower bounds to-date, including the most recent impressive successes. Rank methods (under the name of *flattenings*) are also in wide use in algebraic geometry for proving tensor rank and symmetric tensor rank lower bounds. Our main results are barriers to these methods. In particular,

- Rank methods *cannot* prove better than $\Omega_d(n^{\lfloor d/2 \rfloor})$ lower bound on the tensor rank of *any* d-dimensional tensor of side n. (In particular, they cannot prove super-linear, indeed even $> 8n$ tensor rank lower bounds for *any* 3-dimensional tensors.)
- Rank methods *cannot* prove $\Omega_d(n^{\lfloor d/2 \rfloor})$ on the *Waring rank*[1] of any n-variate polynomial of degree d. (In particular, they cannot prove such lower bounds on stronger models, including depth-3 circuits.)

The proofs of these bounds use simple linear-algebraic arguments, leveraging connections between the *symbolic* rank of matrix polynomials and the usual rank of their evaluations. These

* This work was partially supported by NSF grants CCF-1149888, CCF-1523816, CCF-1412958, CAREER award DMS-1451191, European Community's Seventh Framework Programme (FP7/2007- 2013) under grant agreement number 257575, Simons Collaboration on Algorithms and Geometry, Simons Fellowship in Theoretical Computer Science and Siebel Scholarship.

† A full version of the paper is available at [14], https://arxiv.org/abs/1710.09502

[1] A very restricted form of depth-3 circuits

9th Innovations in Theoretical Computer Science Conference (ITCS 2018).
Editor: Anna R. Karlin; Article No. 1; pp. 1:1–1:19

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

techniques can perhaps be extended to barriers for other arithmetic models on which progress has halted.

To see how these barrier results directly inform the state-of-art in arithmetic complexity we note the following. First, the bounds above nearly match the best explicit bounds we know for these models, hence offer an explanations why the rank methods got stuck there. Second, the bounds above are a far cry (quadratically away) from the true complexity (e.g. of random polynomials) in these models, which *if* achieved (by any methods), are known to imply super-polynomial formula lower bounds.

We also explain the relation of our barrier results to other attempts, and in particular how they significantly differ from the recent attempts to find analogues of "natural proofs" for arithmetic complexity. Finally, we discuss the few arithmetic lower bound approaches which fall outside rank methods, and some natural directions our barriers suggest.

1998 ACM Subject Classification F. Theory of Computation – Algebraic complexity theory

Keywords and phrases Lower Bounds, Barriers, Partial Derivatives and Flattenings

Digital Object Identifier 10.4230/LIPIcs.ITCS.2018.1

1 Introduction

Arithmetic complexity theory (often also called algebraic complexity theory) addresses the computation of algebraic objects (like polynomials, matrices, tensors) using the arithmetic field operations (and sometimes other operations like taking roots). Within computational complexity this field is nearly as old as Boolean complexity theory, which addresses the computation of discrete functions via logical operations, but of course mathematicians were interested in arithmetic computation for centuries before computer science was born. Indeed, Euclid's algorithm for computing GCD, Gauss' discovery of the FFT, and Abel's impossibility result for solving quintic equations by radicals are all precursors of arithmetic complexity theory. Today algebraic algorithms pervade mathematics! Extensive surveys of this field are presented in the books [8, 52], and, more focused on the present material are the recent monographs [48, 12], as well as the book [33] which offers an algebro-geometric perspective.

Structurally, the Boolean and arithmetic theories, and especially the quest for lower bounds which we will focus on, progressed almost hand in hand. Shortly after the important discoveries of reductions and completeness leading to the definitions of P, NP, and complete problems for them, Valiant [51] developed the arithmetic analog notions of VP, VNP and complete problems for them. Separating these pairs of classes stand as the long-term challenges of these fields, and their difficulty has led to the study of a large variety of restricted models in both. Definitions, techniques and results have propagated back and forth and inspired progress, but, all in all, we understand the arithmetic models much better. This of course comes as no surprise. In the arithmetic setting (especially over fields that are large, of characteristic zero, or are algebraically closed) the diverse tools of algebra are available, but have no analogs in the Boolean setting. Moreover, as arithmetic computation is mostly *symbolic* it is (essentially) more stringent than the Boolean computation of functions[2]; indeed, it is known that proving (a non-uniform version of) $\mathsf{P} \neq \mathsf{NP}$ implies $\mathsf{VP} \neq \mathsf{VNP}$ when the underlying field is $\mathbb{C}$ [7]. and thus arithmetic lower bounds are also formally easier to prove!

[2] For example, the *polynomial* $x^p - x$ over $\mathbb{F}_p$ is nontrivial to compute, while the (identically zero) function it represents is trivial.

Despite exciting and impressive progress on arithmetic lower bounds (we will detail many later), some of the most basic questions remain open, and this seeming weakness of current techniques begs explanation, which will hopefully lead to new ones. In Boolean complexity there is a rich interplay between the discovery of the power of new techniques, and then their limitations, in the form of *barrier results*. Such results formally encapsulate a set of lower bound methods, and then prove (unconditional, or sometimes conditionally on natural assumptions) that these cannot solve basic questions. Well known barriers to large classes of techniques include the *relativization* barrier of Baker, Gill and Solovay [6], the *natural proof* barrier of Razborov and Rudich [45] and the *algebrization* barrier of Aaronson and Wigderson [2]. But there are many other important barriers, to more concrete lower bound methods, including [42, 43, 38]. Finding analogous barriers for arithmetic complexity has been much harder; while encapsulation of general lower bound techniques exists, e.g. in [20, 16, 21], there are really no proofs of their limitations (we will discuss these in the related works subsection below).

This paper provides, to the best of our knowledge, the first unconditional barrier results on a very general class of methods, capturing many of the known lower bounds, including the very exciting recent ones. We now begin to describe, through examples, the techniques we encompass under *rank methods* and then explain their limitations.

1.1 Sub-Additive Measures, Rank Bounds and Barriers

Throughout, we will discuss the computation of multivariate polynomials over any field, by arithmetic circuits of various forms, in a way that will not necessitate too many specific details; we will give these as needed, and give formal details in the technical sections. The examples we start with below will demonstrate many "cheap" computations may be encompassed by writing the output polynomial as a "short" sum of *simpler* ones. Thus lower bounds on the number of summands can yield (important) complexity lower bounds. We continue with discussing classes of such lower bound techniques, and then barrier results that put a limit on how large lower bounds such classes of techniques can prove.

Sub-additive measures

Let us start with some examples and then generalize them.

- One of the earliest basic results in arithmetic complexity, due to Hyafil [26] states the following: if a homogeneous circuit of size s computes an n-variate polynomial f of degree d, then

  $$f = g_1 + g_2 + \cdots + g_s$$

  where each g_i is *simple*, which here means *highly reducible*: $g_i = p_i \cdot q_i$, where the degrees of p_i, q_i do not exceed $2d/3$. This result was developed towards parallelizing arithmetic computation, but can also be used for lower bounds: if we could find any sub-additive measure μ on polynomials, which is small on all possible g_i but is large on f, we would have a lower bound on the minimum circuit size s of f! In particular, Hyafil's theorem implies that if the ratio of "large" and "small" values of μ is super-polynomial in n, d, this would imply[3] $\mathsf{VP} \neq \mathsf{VNP}$! We note that Hyafil's theorem is today only one example of numerous other decomposition theorems of similar nature used in lower bounds, e.g. [37, 36, 41, 24] to mention a few.

[3] Since homogenous computation can efficiently simulate non-homogeneous one.

- An even simpler example, where a similar decomposition follows directly from the definition, is tensor rank. Assume that a d-dimensional tensor (with n variables in each dimension) has rank s. This means[4] that

  $$f = g_1 + g_2 + \cdots + g_s$$

  where each g_i is *simple*, which here means *of rank 1*: $g_i = \ell_i^{(1)} \otimes \ell_i^{(2)} \cdots \otimes \ell_i^{(d)}$, where $\ell_i^{(j)}$ is a linear form in the variables of dimension j. Again, any sub-additive measure μ on tensors which is small on all possible rank 1 tensors g_i, but is large on f would yield a lower bound on its tensor rank. This question is no less important than the previous one even though tensor rank seems like a more restricted complexity measure: Raz [40] proved that presenting an explicit tensor f of super-constant dimension $d \leq \log n / \log\log n$, with a nearly-tight tensor rank lower bound of $n^{d(1-o(1))}$ (which holds for most tensors) will imply $\mathsf{VP}_e \neq \mathsf{VNP}$ (namely, explicit super-polynomial lower bounds on formulas)! We note that a similar example as tensor rank, where a decomposition suggests itself by definition, is Waring rank, where each g_i is a d-power of a linear form.
- A third set of examples which directly gives such decompositions of computations is when considering bounded-depth circuits. In almost all computations one can assume without loss of generality that the top (output) gate is a plus gate, and so if a polynomial f is computed by a depth-h circuit of size s, then

  $$f = g_1 + g_2 + \cdots + g_s$$

  where each g_i is *simple* in being of depth $h-1$ (and moreover, with a top product gate). Sub-additive measures small on such simple polynomials and large on f were the key to the many successes on remarkably tight lower bounds for depth-3 and then depth-4 circuits [36, 28, 22, 27, 17, 30, 29]. These include the breakthrough of $(nd)^{\sqrt{d}}$ explicit lower bounds [22] on the size of homogeneous depth-4 circuits, which again seem much more restricted than it is: any super-constant improvement of the exponent will imply $\mathsf{VP} \neq \mathsf{VNP}$!

There are many other examples in which obtaining such decompositions as above uses extra tools like approximations, random restrictions, or iterations. Abstracting all these examples and indeed most known lower bounds in arithmetic complexity[5], can be done in a simple way. Let S be a set of *simple* polynomials, and let $\hat{S}$ be their linear span. The S-complexity $c_S(f)$ of a polynomial $f \in \hat{S}$ is simply the smallest number s such that $f = g_1 + g_2 + \cdots + g_s$ and each $g_i \in S$. A *sub-additive* measure μ is a function $\mu : \hat{S} \to \mathbb{R}^+$ such that

$$\mu(g+h) \leq \mu(g) + \mu(h)$$

for any $g, h \in \hat{S}$. Extending μ to sets, denoting $\mu(T) = \max\{\mu(g) : g \in T\}$, we can immediately derive a lower bound on $c_S(f)$ for any polynomial f by

$$c_S(f) \geq \mu(f)/\mu(S).$$

Let Δ_S denote all possible sub-additive measures on $\hat{S}$. It is a triviality that c_S itself is a sub-additive measure in Δ_S, and hence this method can in principle provide tight lower

[4] Directly generalizing matrix rank, which is the case $d = 2$.
[5] The discussion below is quite general and indeed applies to lower bounds and barriers that use sub-additive measures in practically any computational model.

bound on the complexity $c_S(f)$ for every f. However, the difficulty of proving lower bounds precisely means that c_S is hard to understand, and so we try to "approximate it" with simpler measures $\mu \in \Delta$ for some family $\Delta \subseteq \Delta_S$ of sub-additive measures which are hopefully simpler to understand, compute and reason about.

Barriers for sub-additive measures

This brings us to the topic of this paper: barriers, or limits to the power of such class of lower bound methods. A *barrier* result for any such class of sub-additive measures $\Delta \subseteq \Delta_S$ simply asserts that $\mu(f)$ is *small* for *every* $\mu \in \Delta$ and any $f \in \hat{S}$ (whenever $\mu(S)$ is small). The quantity

$$c(\Delta) = \mu(\hat{S})/\mu(S)$$

upper bounds the best lower bound which can be proven using *any* $\mu \in \Delta$ on *any* polynomial $f \in \hat{S}$, simply as $\mu(f) \leq c(\Delta) \cdot \mu(S)$ for all of them.

Of course, concrete lower bounds are obtained using specific measures μ, and there is always hope that a clever variant of such a choice will give even better bounds; indeed, much of the progress in lower bounds is of this nature. The quality of barrier result is in classifying as large as possible a class of measures Δ, which captures many complexity measures, such that either $c(\Delta)$ is close to the best known lower bounds, or it is well separated with a "desired" lower bound (e.g. one that would approach the complexity of a random polynomial, or that would significantly improve the state of art). In this paper we focus on *rank methods*, which we turn to describe now.

Rank methods

The rank function of matrices, is at once extremely well studied and understood in linear algebra, and is sub-additive. This has made numerous (implicit and explicit) choices of sub-additive measures, for a variety of computational models, to be defined via matrix rank, as follows. Fix a field $\mathbb{F}$, and let $\mathrm{Mat}_m(\mathbb{F})$ denote the set of all $m \times m$ matrices over $\mathbb{F}$. Fix the set of (simple) polynomials S, (and thereby also their span $\hat{S}$) as before. Define the class $\Delta_0^S \subseteq \Delta_S$ to be the set of sub-additive measures μ which arise in the following way. Let $L : S \to \mathrm{Mat}_m(\mathbb{F})$ be any *linear* map for some integer m. Namely, for all $g, h \in S$ (and hence also in $\hat{S}$) we have $L(g+h) = L(g) + L(h)$, and that $L(bg) = L(g)$ for any non-zero constant $b \in \mathbb{F}$. Define

$$\mu_L(f) = rank_{\mathbb{F}}(L(f)).$$

Clearly, all these $\mu_L \in \Delta_0^S$ are sub-additive measures on S. We call the elements of Δ_0^S as *rank methods* for S.

As mentioned, rank methods abound in arithmetic (and other) lower bounds. The possibly familiar names including *partial derivatives, shifted partial derivatives, evaluation dimension, coefficient dimension* which are used e.g. in these lower bounds for monotone, non-commutative, homogeneous, multilinear, bounded-depth and other models [37, 49, 42, 36, 28, 22, 27, 15, 17, 30, 29] are all rank methods, and in many of these papers are explicitly stated as such. Moreover, in algebraic geometry, rank methods (usually called *flattenings*) are responsible for almost all tensor rank and symmetric tensor rank lower bounds (see e.g. [33]).

What should be stressed is that rank methods are extremely general. We do not restrict the size m of matrices used in any way (and indeed in some applications, like shifted partial derivatives [22], m grows super exponentially in the basic size parameters n, d). Moreover,

we demand no explicitness in the specification of the linear map L (and indeed, in some applications, like the multilinear formula lower bounds in [39, 41] the map is chosen at random). The barrier results hold for all.

We prove barrier results for two classes of very weak computational models, *tensor rank* and *Waring rank*, which are very special cases of (respectively) multilinear and homogeneous depth-3 circuits (which themselves are the weakest class of circuits studied[6]. As with all barrier results, the weaker the model for which they are proved, the better, as they scale up for stronger models automatically! As discussed above, we will compare our barriers both to the state-of-art lower bounds in these models, as well to the best one can hope for, namely the complexity of random polynomials.

1.2 Main results

Our results below work for infinite fields $\mathbb{F}$.[7] We start with tensor rank, and proceed with Waring rank, which may be viewed as a symmetric version of tensor rank. In both cases, our barrier results nearly match (up to a function of d, the degree[8]) the best explicit lower bounds (obtained by rank methods), and are roughly quadratically away from the (desired) lower bounds that hold for random polynomials.

Tensor rank

Tensors abound in mathematics and physics, and have been studied for centuries. We refer the reader to the book [32] for one good survey. From a computational perspective tensors have been extremely interesting as well, as many problems naturally present themselves in tensor form. In arithmetic complexity they are often called *set-multilinear* polynomials. While 2-dimensional tensors, namely matrices, are very well understood, d-dimensional tensors possess far less structure, and one way this is manifested is that the problem of computing tensor rank of 3-dimensional tensors is already NP-complete [23]. Many special cases, approximations and related decompositions of tensors were studied, especially recently with machine learning applications [11, 35, 5, 25, 18]. Let us define the model and problem formally.

Fix n, d. The family of polynomials of interest here is $\hat{S} = \mathrm{Ten}_{n,d}(\mathbb{F})$, namely degree d polynomials in d sets of n variables (so, total of nd variables), in which each monomial has precisely one variable from each set. The coefficients of a tensor are naturally described by an $[n]^d$ box with entries from $\mathbb{F}$. The simple polynomials S are *rank-1* tensors, namely those which are products of d linear forms, one in each set of variables (equivalently, the coefficients are described by the tensor product of d vectors). The tensor rank of a tensor f is the smallest number of rank-1 tensors which add up to it.

Most tensors have rank about n^{d-1}/d. Explicit lower bounds are way worse. It is trivial to construct an explicit d-dimensional tensor of rank $n^{\lfloor d/2 \rfloor}$, and the best known lower bound is only a factor of 2 larger. Specifically, [4] give an explicit tensor with 0,1 coefficients of tensor rank at least $2n^{\lfloor d/2 \rfloor} + n - d \log n$. Note in particular that the best lower bound for $d = 3$ is about $3n$. Although the lower bounds of [4] are not attained via a rank method, many other lower bounds for tensor rank are attained via a rank method in Δ_0^T (T for

[6] As depth-2 circuits simply represent polynomials trivially, as sums of monomials.

[7] Our results below hold for all large enough fields $\mathbb{F}$ (polynomial in n, m, d), however, in most cases the dimension m of the matrices is exponentially large in the parameters of interest – that is, n, d.

[8] Which is a constant in the very interesting cases where the degree d is a constant!

Tensor), namely using a sub-additive measure in the class of rank methods [34, 31]. Our barrier result proves that no bound better $2^d \cdot n^{\lfloor d/2 \rfloor}$ can be proven by rank methods, and in particular for $d = 3$, they cannot beat $8n$ (a factor $8/3$ away from the best explicit lower bound!).

▶ **Theorem 1** (Statement of Theorem 3). $c(\Delta_0^T) \leq 2^d \cdot n^{\lfloor d/2 \rfloor}$.

Waring rank

The Waring problem has a long history in mathematics, first in its number theoretic form initiated by Waring [53] in 1770 (writing integers as short sums of d-powers of other integers), and then in its algebraic form we care about, initiated by Sylvester [50] in 1851 (writing polynomials as short sums of powers of linear forms). Some of the basic questions (computing this minimum for monomials and for random polynomials) were only very recently resolved, using algebraic geometric techniques [10, 3]. In arithmetic complexity this model is often referred to as *depth-3 powering circuits*. Let us formalize the problem.

Fix n, d. The family of polynomials of interest here is $\hat{S} = \text{poly}_{n,d}$, all n-variate polynomials of total degree d. The simple generating set S we care about here is the set of all d-powers, namely all polynomials of the form ℓ^d, where ℓ is an affine function in the n given variables. So, $c_S(f)$ is the smallest number s such that f can be written as a sum of such d powers.

For most polynomials, the Waring rank was settled by [3], and is about $(n-1)^d$ for d much smaller than n, and is precisely

$$\left\lceil \frac{1}{n} \cdot \binom{n+d-1}{n-1} \right\rceil.$$

It is trivial to find an explicit $f \in \text{poly}_{n,d}$ whose Waring rank is $\Omega(n^{\lfloor d/2 \rfloor})$, and the best known lower bound, due to [19] (again via rank method in Δ_0^W), is only a little better,

$$\binom{n+\lfloor d/2 \rfloor - 1}{\lfloor d/2 \rfloor} + \lfloor n/2 \rfloor - 1.$$

Our barrier result proves that rank methods cannot improve this lower bound even by a factor of roughly d.

▶ **Theorem 2** (Barriers for Waring Rank[9]). $c(\Delta_0^W) \leq (d+1) \cdot \binom{n+\lfloor d/2 \rfloor}{n}$.

1.3 High-level ideas of the proof

As mentioned, the proofs of our barrier results use only simple tools of linear algebra (although their use and combination is a bit subtle). Here are the key ideas of the proof, written abstractly in the general notation established above (again, we believe that they can be applied in other settings beyond the two we consider in this paper).

Consider any simple set S of polynomials, and rank methods Δ_0^S for it. Thus, we need to provide an upper bound on the quantity $c(\Delta_0^S)$, namely on the ratio $\mu_L(f)/\mu_L(S)$ for every $f \in \hat{S}$, and every linear map $L : S \to \text{Mat}_m(\mathbb{F})$. Set $r = \mu_L(S)$.

[9] A proof of this theorem appears in the full version of the paper [14]

- We view linear map L, which gives rise to a sub-additive measure in Δ_0^S, as a matrix polynomial, namely as a polynomial with matrix coefficients, or equivalently as a symbolic matrix whose entries are polynomials. The variables of these polynomials will be the *parameters* of the family of *simple* polynomials S (these parameters are the coefficients of the linear forms appearing in the decompositions in both the tensor rank and Waring rank settings). Call this symbolic matrix $L(S)$.
- Next, the *symbolic* rank of $L(S)$ (over the field of rational functions in these variables) is bounded by the maximum rank of any *evaluation* of this matrix polynomial (this is the only place we use the fact that the field is large enough). By assumption, as these evaluations are all in the image of L on the simple polynomials S, this maximum rank is at most r, and so is the symbolic rank.
- The symbolic rank gives rise to a decomposition $L(S) = KM$ with M, K having dimensions $m \times r$ and $r \times m$ respectively, and their entries are *rational functions* in the variables appearing in $L(S)$. We show that with a small loss in the dimension r, this affords a much nicer decomposition $L(S) = K'M'$, with dimensions $m \times r'$ and $r' \times m$ respectively, but now the entries of K', M' are *polynomial* functions of the variables. Moreover, the polynomials in every column of K' and every row of M' are homogeneous of the same degree. For tensor rank we obtain $r' = r2^d$, and for Waring rank we have $r' = r(d+1)$.
- As all entries in matrix $L(S)$ are polynomials of degree d, we must have for every $i \in [r']$, that either the i'th column of K' or the i'th row of M' have degree at most $\lfloor d/2 \rfloor$. The dimension of the space of (vector) coefficients of these vectors of polynomials is an appropriate function D of n, d (which in both cases we care about is about $n^{\lfloor d/2 \rfloor}$). Each such vector of polynomials generates at most D *constant* vectors of their coefficients.
- Combining what we have, we see that for every $g \in S$, we have a decomposition $L(g) = C(g) + R(g)$, where the columns of $C(g)$ are spanned by at most $r'D$ vectors *independent of* g, and the rows of R are spanned by at most $r'D$ vectors *independent of* g (indeed the total number of these vectors is $r'D$). This gives an upper bound of $r'D$ on the rank of each $L(g)$, which of course is not interesting as we already have an upper bound of r on each.
- The punchline is obtained by using the linearity of L, and the fact that $\hat{S}$ is the linear span of S. Together, these imply that *every* matrix $L(f)$ with $f \in \hat{S}$ is also in the linear span of the matrices $\{L(g) : g \in S\}$, and so the same decomposition holds for them. Thus, the rank of each $L(f)$ is at most $r'D$, which is a bound on $\mu_L(\hat{S})$. Thus, $c(\Delta_0^S) \leq r'D/r$. In the two settings we consider, D is roughly the best known explicit lower bound, and r'/r is a function of d (namely, $d+1$ for Waring rank, and 2^d for tensor rank).

1.4 Related Work

We now mention other attempts to provide barriers to arithmetic circuit lower bounds. We also mention rank lower bounds in Boolean complexity, and barriers for them. As will be evident, our work is very different than both sets.

All barrier results we are aware of in arithmetic complexity theory attempt to find analogs of the *natural proof* barrier in Boolean circuit complexity of Razborov and Rudich [45]. Roughly, a lower bound technique is *natural* if it satisfies three properties: usefulness, constructively, largeness which we will not need to define. They show how many Boolean circuit lower bound techniques satisfy these properties. Now crucially, the barrier results for natural proofs in the Boolean setting are *conditional*: they hold under a computational assumption on the existence of efficient pseudorandom generators. In this setting, this assumption is widely believed, and is known to follow from e.g. the existence of exponentially hard one-way functions (one which the world relies for cryptography and e-commerce).

In several works, starting with [1, 20], and following with the recent [16, 21], it was understood that an analogous framework with the same three properties is simple to describe (replacing the representation of Boolean functions by their truth tables by the representation of low-degree multivariate polynomials by their list of coefficients). And indeed, it captures essentially all arithmetic lower bounds known. Unfortunately, the main difference from the Boolean setting is the non-existence of an analogous pseudo-randomness theory, and a believable complexity assumption. Several suggestions for such an assumption were made in the works above, and as articulated in [16, 21], they all take the form of the existence of *succinct* hitting sets for small arithmetic circuits (indeed, such existence is *equivalent* to a barrier result). This assumption is related to PIT (polynomial identity testing) and GCT (geometric complexity theory), but the confidence in it is still shaky (initial work in [16] shows succinct hitting sets against extremely weak models of arithmetic circuits). But regardless how believable this assumption is, note that this barrier is again, conditional!

As mentioned earlier, our barrier results are completely unconditional, and moreover require no constructivity from the lower bound proof (thus capturing methods which are not strictly natural in the sense above). On the other hand, our framework of rank methods capture only a large subset, but certainly not all of the known lower bound techniques.

It is interesting that rank methods were used not only in arithmetic complexity, but also in Boolean complexity. While not directly related to our arithmetic setting, we mention where it was used, and which barriers were studied. First, Razborov has used the rank of matrices in an essential way for his lower bound on $AC^0[2]$ (although an elegant route around it was soon after devised by Smolensky [49]). In another work, Razborov [44] has shown how rank methods can be used to prove superpolynomial lower bounds on *monotone* Boolean formulas. His methods were recently beautifully extended to other monotone variants of other models including span programs and comparator circuits in [46]. The potential of such methods to proving *non-monotone* lower bounds for Boolean formulas was considered by Razborov [42], where he proves a strong barrier result in this Boolean setting. Observing that rank is a *submodular* function, he presents a barrier for any submodular progress measure on Boolean formulae: *no such method can prove a super-linear lower bound!*. His barrier was recently made more explicit in [38].

1.5 Organization

In Section A we establish the notation that will be used throughout the paper and provide some lemmas which we will need in the later sections. In Section B, we establish the main technical content of our paper: we define three notions of matrix decomposition and relate these new definitions to commutative rank. In Section 2, we apply the new decompostions from Section B to obtain the main results of the paper, which are the limitations of the rank techniques. Finally, in Section 3 we conclude the paper and present some open questions and future directions of this work.

2 Rank Bounds

In this section, we show how the matrix decomposition techniques developed in Section B can be used to establish barriers to rank-based methods used to prove lower bounds for tensor rank.[10]

[10] For our results on barriers for Waring rank and constant depth circuits, please see the full version of the paper [14].

We show that any linear map, denoted here by $L : \mathrm{Ten}_{n,d}(\mathbb{F}) \to \mathrm{Mat}_m(\mathbb{F})$, for which $\mathrm{rank}(L(\mathbf{u}_1 \otimes \cdots \otimes \mathbf{u}_d)) \leq r$ for all rank one tensors has the property that $\mathrm{rank}(L(T)) \leq r \cdot 2^d \cdot n^{\lfloor d/2 \rfloor}$ for any tensor $T \in \mathrm{Ten}_{n,d}(\mathbb{F})$. This in turn, implies that such a technique cannot yield better lower bounds than

$$\mathrm{rank}(T) > 2^d \cdot n^{\lfloor d/2 \rfloor}$$

for any explicit tensor $T \in \mathrm{Ten}_{n,d}(\mathbb{F})$.

To put this matter into perspective, it is very easy to obtain explicit tensors $T \in \mathrm{Ten}_{n,d}(\mathbb{F})$ whose tensor rank is lower bounded by $\mathrm{rank}(T) \geq n^{\lfloor d/2 \rfloor}$. For instance, one can just take a full-rank matrix in $\mathrm{Mat}_{n^{\lfloor d/2 \rfloor}}(\mathbb{F})$. Nevertheless, despite much work on tensor rank lower bounds, the best lower bounds for the rank of explicit tensors are still of the form $\Omega(n^{\lfloor d/2 \rfloor})$, as seen in the works [9, 4, 32].

On the other hand, it is well-known, see for instance [33], that a random tensor has rank on the order of $\frac{n^{d-1}}{d}$. Thus, our paper shows that rank-based methods for proving tensor rank lower bounds will not suffice to prove strong tensor lower bounds. We now state the main theorem of this section.

▶ **Theorem 3** (Tensor Rank Upper Bounds). *Let $m, n \in \mathbb{N}$ be positive integers and $L : \mathrm{Ten}_{n,d}(\mathbb{F}) \to \mathrm{Mat}_m(\mathbb{F})$ be a linear map such that each rank one tensor $\mathbf{u}_1 \otimes \cdots \otimes \mathbf{u}_d$ is mapped into a matrix $L(\mathbf{u}_1 \otimes \cdots \otimes \mathbf{u}_d)$ such that*

$$\mathrm{rank}(L(\mathbf{u}_1 \otimes \cdots \otimes \mathbf{u}_d)) \leq r.$$

Then it holds that

$$\mathrm{rank}(L(f)) \leq r \cdot 2^d \cdot n^{\lfloor d/2 \rfloor}$$

for any tensor $f \in \mathrm{Ten}_{n,d}(\mathbb{F})$.

Proof. Let $\mathbf{x}_1 \otimes \cdots \otimes \mathbf{x}_d$ be a generic rank one tensor, where $\mathbf{x}_i = (x_{i1}, \ldots, x_{in})$, with x_{ij} being variables which take values from $\mathbb{F}$, for all $i \in [d]$. Additionally, let $\mathbf{x} = (\mathbf{x}_1, \ldots, \mathbf{x}_d)$, that is, $\mathbf{x}$ is the set of all variables involved, taking into account the partitions of the variables. As the map $L : \mathrm{Ten}_{n,d}(\mathbb{F}) \to \mathrm{Mat}_m(\mathbb{F})$ is a linear map, we must have that

$$L(\mathbf{x}_1 \otimes \cdots \otimes \mathbf{x}_d) = \sum_{i_1, i_2, \ldots, i_d = 1}^{n} A_{i_1, i_2, \ldots, i_d} \prod_{j=1}^{d} x_{j i_j}$$

where each $A_{i_1, i_2, \ldots, i_d} \in \mathrm{Mat}_m(\mathbb{F})$ is a complex $m \times m$ matrix.[11] Hence, $M(\mathbf{x}) = L(\mathbf{x}_1 \otimes \cdots \otimes \mathbf{x}_d)$ is a matrix with set-multilinear polynomial entries, where each polynomial is set-multilinear over the sets of variables $\mathbf{x}_1, \ldots, \mathbf{x}_d$.

By Lemma 10 and the assumption that $\mathrm{rank}(L(\mathbf{u}_1 \otimes \cdots \otimes \mathbf{u}_d)) \leq r$ for any multiset of vectors $\mathbf{u}_i \in \mathbb{F}^n$, we have that

$$\mathrm{rank}_{\mathbb{F}(\mathbf{x})}(L(\mathbf{x}_1 \otimes \cdots \otimes \mathbf{x}_d)) \leq r.$$

In this case, the conditions of Lemma 17 apply and therefore there exist $R \leq r \cdot 2^d$ vectors of homogeneous set-multilinear polynomials $\mathbf{f}_i(\mathbf{x}), \mathbf{g}_i(\mathbf{x}) \in \mathbb{F}[\mathbf{x}]$ for which

$$M(\mathbf{x}) = \sum_{i=1}^{R} \mathbf{f}_i(\mathbf{x}) \otimes \mathbf{g}_i(\mathbf{x}).$$

[11] One can see this by looking at the standard basis of the space $\mathrm{Ten}_{n,d}(\mathbb{F})$ given by tensoring the standard basis vectors $\mathbf{e}_{i_1} \otimes \cdots \otimes \mathbf{e}_{i_d}$.

Moreover, for all $i \in [R]$, there exists a set S_i such that $\mathbf{f}_i(\mathbf{x})$ is set-multilinear with respect to the partition $(\mathbf{x}_j)_{j \in S_i}$ and $\mathbf{g}_i(\mathbf{x})$ is set-multilinear with respect to the partition $(\mathbf{x}_j)_{j \in [d] \setminus S_i}$. Thus, $\deg(\mathbf{f}_i) + \deg(\mathbf{g}_i) \leq d$, which implies that $\min(\deg(\mathbf{f}_i), \deg(\mathbf{g}_i)) \leq \lfloor d/2 \rfloor$, for each $i \in [R]$. This bound on the minimum degree, combined with Corollary 14 and the fact that $\mathbf{f}_i(\mathbf{x})$ and $\mathbf{g}_i(\mathbf{x})$ are set-multilinear, yield

$$\operatorname{rank}(\mathcal{C}(\mathbf{f}_i(\mathbf{x}) \otimes \mathbf{g}_i(\mathbf{x}))) \leq n^{\lfloor d/2 \rfloor}.$$

As $\operatorname{rank}(\mathcal{C}(M(\mathbf{x}))) \leq \sum_{i=1}^{R} \operatorname{rank}(\mathcal{C}(\mathbf{f}_i(\mathbf{x}) \otimes \mathbf{g}_i(\mathbf{x})))$, we have that

$$\operatorname{rank}(\mathcal{C}(M(\mathbf{x}))) \leq R \cdot n^{\lfloor d/2 \rfloor}.$$

To finish the proof, it is enough to show that $L(f) \in \mathcal{C}(M(\mathbf{x}))$, for any $f \in \operatorname{Ten}_{n,d}(\mathbb{F})$.

For any rank one tensor $\mathbf{u}_1 \otimes \cdots \otimes \mathbf{u}_d$, we have that $L(\mathbf{u}_1 \otimes \cdots \otimes \mathbf{u}_d) \in \mathcal{C}(M(\mathbf{x}))$, as $L(\mathbf{u}_1 \otimes \cdots \otimes \mathbf{u}_d) = M(\mathbf{u})$. As any element $f \in \operatorname{Ten}_{n,d}(\mathbb{F})$ can be written as a linear combination of rank one tensors and by linearity of L, we have that

$$L(f) \in \operatorname{span}\{L(\mathbf{u}_1 \otimes \cdots \mathbf{u}_d) \mid \mathbf{u}_1, \ldots, \mathbf{u}_d \in \mathbb{F}^n\} \subseteq \mathcal{C}(M(\mathbf{x})).$$

Thus, $L(f) \in \mathcal{C}(M(\mathbf{x}))$ and we have that

$$\operatorname{rank}(L(f)) \leq \operatorname{rank}(\mathcal{C}(M(\mathbf{x}))) \leq R \cdot n^{\lfloor d/2 \rfloor},$$

as we wanted. ◀

The theorem above implies the following barrier on rank-based techniques.

▶ **Corollary 4.** *Let $m, n \in \mathbb{N}$ be positive integers and $L : \operatorname{Ten}_{n,d}(\mathbb{F}) \to \operatorname{Mat}_m(\mathbb{F})$ be a linear map (i.e., a flattening). Then, any rank methods which use this linear map cannot prove lower bounds better than*

$$\operatorname{rank}(f) > 2^d \cdot n^{\lfloor d/2 \rfloor}$$

for any tensor $f \in \operatorname{Ten}_{n,d}(\mathbb{F})$.

3 Conclusion and Open Problems

In this paper, we prove the first unconditional barrier for a wide class of lower bound techniques for tensor rank as well as the Waring rank of a polynomial. In particular, for 3-dimensional tensor rank, we show for the first time that a wide class of techniques cannot improve a known linear lower bound (of $2n$) even beyond $8n$. Additionally, we provide an explicit instantiation of the rank method for depth-3 circuits, suggesting it will either help prove better lower bounds, or help develop a barrier for this model that explains the difficulty of proving better lower bounds.

We now provide a list of interesting directions for further research, both on the computational side as well as on the mathematical side.

1. Expand the set of methods for which *unconditional* barrier results be proven in arithmetic complexity theory, beyond the rank methods we study in this paper. In particular, can they be expanded to the use of *non-linear* mappings L, possibly of low degree?
2. Expand the set of arithmetic models for which barriers can be established for rank methods, beyond the two models studied here.
3. In some sense, rank methods "flatten" polynomials of degree $d > 2$ into matrices (in 2 dimensions), in a similar fashion flattening methods in algebraic geometry are used (for very similar purposes). Can this connection be further formalized and used?

References

1 Scott Aaronson and Andrew Drucker. Arithmetic natural proofs theory is sought. *Blog post*, 2008.

2 Scott Aaronson and Avi Wigderson. Algebrization: A new barrier in complexity theory. *ACM Transactions on Computation Theory (TOCT)*, 1(1):2, 2009.

3 James Alexander and André Hirschowitz. Polynomial interpolation in several variables. *Journal of Algebraic Geometry*, 4(2):201–222, 1995.

4 Boris Alexeev, Michael A Forbes, and Jacob Tsimerman. Tensor rank: Some lower and upper bounds. In *Computational Complexity (CCC), 2011 IEEE 26th Annual Conference on*, pages 283–291. IEEE, 2011.

5 Anima Anandkumar, Dean P Foster, Daniel J Hsu, Sham M Kakade, and Yi-Kai Liu. A spectral algorithm for latent dirichlet allocation. In *Advances in Neural Information Processing Systems*, pages 917–925, 2012.

6 Theodore Baker, John Gill, and Robert Solovay. Relativizations of the p=?np question. *SIAM Journal on computing*, 4(4):431–442, 1975.

7 Peter Bürgisser. *Completeness and reduction in algebraic complexity theory*, volume 7. Springer Science & Business Media, 2013.

8 Peter Bürgisser, Michael Clausen, and Amin Shokrollahi. *Algebraic complexity theory*, volume 315. Springer Science & Business Media, 2013.

9 Peter Bürgisser and Christian Ikenmeyer. Geometric complexity theory and tensor rank. In *Proceedings of the forty-third annual ACM symposium on Theory of computing*, pages 509–518. ACM, 2011.

10 Enrico Carlini, Maria Virginia Catalisano, and Anthony V Geramita. The solution to the waring problem for monomials and the sum of coprime monomials. *Journal of algebra*, 370:5–14, 2012.

11 Joseph T Chang. Full reconstruction of markov models on evolutionary trees: identifiability and consistency. *Mathematical biosciences*, 137(1):51–73, 1996.

12 Xi Chen, Neeraj Kayal, and Avi Wigderson. *Partial derivatives in arithmetic complexity and beyond*. Now Publishers Inc, 2011.

13 Richard DeMillo and Richard Lipton. A probabilistic remark on algebraic program testing. *Information Processing Letters*, 7(4):193–195, 1978.

14 Klim Efremenko, Ankit Garg, Rafael Oliveira, and Avi Wigderson. Barriers for rank methods in arithmetic complexity. *arXiv preprint arXiv:1710.09502*, 2017.

15 Michael A Forbes, Ramprasad Saptharishi, and Amir Shpilka. Hitting sets for multilinear read-once algebraic branching programs, in any order. In *Proceedings of the forty-sixth annual ACM symposium on Theory of computing*, pages 867–875. ACM, 2014.

16 Michael A Forbes, Amir Shpilka, and Ben Lee Volk. Succinct hitting sets and barriers to proving algebraic circuits lower bounds. *arXiv preprint arXiv:1701.05328*, 2017.

17 Hervé Fournier, Nutan Limaye, Guillaume Malod, and Srikanth Srinivasan. Lower bounds for depth-4 formulas computing iterated matrix multiplication. *SIAM Journal on Computing*, 44(5):1173–1201, 2015.

18 Rong Ge and Tengyu Ma. On the optimization landscape of tensor decompositions. *arXiv preprint arXiv:1706.05598*, 2017.

19 Fulvio Gesmundo and JM Landsberg. Explicit polynomial sequences with maximal spaces of partial derivatives and a question of k. mulmuley. *arXiv preprint arXiv:1705.03866*, 2017.

20 Joshua A Grochow. Unifying known lower bounds via geometric complexity theory. *computational complexity*, 24(2):393–475, 2015.

21 Joshua A Grochow, Mrinal Kumar, Michael Saks, and Shubhangi Saraf. Towards an algebraic natural proofs barrier via polynomial identity testing. *arXiv preprint arXiv:1701.01717*, 2017.

22 Ankit Gupta, Pritish Kamath, Neeraj Kayal, and Ramprasad Saptharishi. Approaching the chasm at depth four. *J. ACM*, 61(6):33:1–33:16, 2014. doi:10.1145/2629541.

23 Johan Håstad. Tensor rank is np-complete. *Journal of Algorithms*, 11(4):644–654, 1990.

24 Pavel Hrubes, 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.

25 Daniel Hsu and Sham M Kakade. Learning mixtures of spherical gaussians: moment methods and spectral decompositions. In *Proceedings of the 4th conference on Innovations in Theoretical Computer Science*, pages 11–20. ACM, 2013.

26 Laurent Hyafil. On the parallel evaluation of multivariate polynomials. *SIAM Journal on Computing*, 8(2):120–123, 1979.

27 N. Kayal, N. Limaye, C. Saha, and S. Srinivasan. Super-polynomial lower bounds for depth-4 homogeneous arithmetic formulas. In *Symposium on Theory of Computing, STOC 2014*, pages 119–127, 2014.

28 Neeraj Kayal. An exponential lower bound for the sum of powers of bounded degree polynomials. *Electronic Colloquium on Computational Complexity (ECCC)*, 19:81, 2012. URL: http://eccc.hpi-web.de/report/2012/081.

29 Mrinal Kumar and Ramprasad Saptharishi. An exponential lower bound for homogeneous depth-5 circuits over finite fields. *arXiv preprint arXiv:1507.00177*, 2015.

30 Mrinal Kumar and Shubhangi Saraf. Superpolynomial lower bounds for general homogeneous depth 4 arithmetic circuits. In *International Colloquium on Automata, Languages, and Programming*, pages 751–762. Springer, 2014.

31 JM Landsberg. Nontriviality of equations and explicit tensors in cm cm cm of border rank at least 2m- 2. *Journal of Pure and Applied Algebra*, 219(8):3677–3684, 2015.

32 Joseph M Landsberg. *Tensors: geometry and applications*, volume 128. American Mathematical Society Providence, RI, 2012.

33 Joseph M Landsberg. *Geometry and Complexity Theory*. Cambridge, 2017.

34 Joseph M Landsberg and Giorgio Ottaviani. New lower bounds for the border rank of matrix multiplication. *Theory of Computing*, 11(11):285–298, 2015.

35 Elchanan Mossel and Sébastien Roch. Learning nonsingular phylogenies and hidden markov models. In *Proceedings of the thirty-seventh annual ACM symposium on Theory of computing*, pages 366–375. ACM, 2005.

36 N. Nisan and A. Wigderson. Lower bound on arithmetic circuits via partial derivatives. *Computational Complexity*, 6:217–234, 1996.

37 Noam Nisan. Lower bounds for non-commutative computation. *STOC*, pages 410–418, 1991.

38 Aaron Potechin. A note on amortized space complexity. *arXiv preprint arXiv:1611.06632*, 2016.

39 Ran Raz. Multi-linear formulas for permanent and determinant are of super-polynomial size. *Journal of the ACM (JACM)*, 56(2):8, 2009.

40 Ran Raz. Tensor-rank and lower bounds for arithmetic formulas. 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 659–666. ACM, 2010. doi:10.1145/1806689.1806780.

41 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.

42 Alexander Razborov. On submodular complexity measures. URL: http://people.cs.uchicago.edu/~razborov/files/sub.pdf.

43 Alexander A Razborov. On the method of approximations. In *Proceedings of the twenty-first annual ACM symposium on Theory of computing*, pages 167–176. ACM, 1989.

44 Alexander A Razborov. Applications of matrix methods to the theory of lower bounds in computational complexity. *Combinatorica*, 10(1):81–93, 1990.

45 Alexander A Razborov and Steven Rudich. Natural proofs. In *Proceedings of the twenty-sixth annual ACM symposium on Theory of computing*, pages 204–213. ACM, 1994.

46 Robert Robere, Toniann Pitassi, Benjamin Rossman, and Stephen A Cook. Exponential lower bounds for monotone span programs. In *Foundations of Computer Science (FOCS), 2016 IEEE 57th Annual Symposium on*, pages 406–415. IEEE, 2016.

47 Jack Schwartz. Fast probabilistic algorithms for verification of polynomial identities. *Journal of the ACM*, 27:701–717, 1980.

48 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.

49 Roman Smolensky. On representations by low-degree polynomials. In *Foundations of Computer Science, 1993. Proceedings., 34th Annual Symposium on*, pages 130–138. IEEE, 1993.

50 James Joseph Sylvester. Lx. on a remarkable discovery in the theory of canonical forms and of hyperdeterminants. *Philosophical Magazine Series 4*, 2(12):391–410, 1851.

51 Leslie G Valiant. Completeness classes in algebra. In *Proceedings of the eleventh annual ACM symposium on Theory of computing*, pages 249–261. ACM, 1979.

52 Joachim Von Zur Gathen and Jürgen Gerhard. *Modern computer algebra*. Cambridge university press, 2013.

53 E. Waring. Meditationes algebraicæ. In *Archdeacon*, Cambridge, 1770.

54 Richard Zippel. Probabilistic algorithms for sparse polynomials. *EUROSAM*, pages 216–226, 1979.

A Preliminaries

In this section, we establish the notation which will be used throughout the paper and some important background which we shall need to prove our claims in the next sections.

A.1 General Facts and Notations

For simplicity of exposition, we will work over a field $\mathbb{F}$ which is algebraically closed and of characteristic zero, even though our results also hold over infinite fields which need not be algebraically closed.[12] From now on we will use boldface to denote a vector of variables or of field elements. For instance, $\mathbf{x} = (x_1, \ldots, x_n)$ is the vector of variables $x_1, \ldots, x_n$ and $\mathbf{a} = (a_1, \ldots, a_n) \in \mathbb{F}^n$ is a vector of elements $a_1, \ldots, a_n$ from the field $\mathbb{F}$.

For any vector of non-negative integers $\mathbf{a} \in \mathbb{N}^n$ and a vector of n variables $\mathbf{x}$, we define $\mathbf{a}! = \prod_{i=1}^{n} a_i!$ and $\mathbf{x}^{\mathbf{a}} = \frac{1}{\mathbf{a}!} \cdot \prod_{i=1}^{n} x_i^{a_i}$. Since the monomials $\mathbf{x}^{\mathbf{a}}$, $\mathbf{a} \in \mathbb{N}^n$, form a linear basis for the ring of polynomials $\mathbb{F}[\mathbf{x}]$, we can write any polynomial $f(\mathbf{x}) \in \mathbb{F}[\mathbf{x}]$ as

$$f(\mathbf{x}) = \sum_{\mathbf{a} \in \mathbb{N}^n} \alpha_{\mathbf{a}} \mathbf{x}^{\mathbf{a}}.$$

[12] In general, we only need a field with characteristic polynomial in the number of variables, the degree of the polynomials and the dimension of matrices being studied. We cannot work over field extensions, as we need to use Lemma 10 over the base field.

We will denote the coefficients of the polynomial $f(\mathbf{x})$ by $\text{coeff}_{\mathbf{a}}(f(\mathbf{x})) = \alpha_{\mathbf{a}}$.

The degree of a polynomial $f(\mathbf{x}) \in \mathbb{F}[\mathbf{x}]$ with respect to a variable x_i, denoted by $\deg_i(f(\mathbf{x}))$ is the maximum degree of x_i in a nonzero monomial of $f(\mathbf{x})$. If $\deg_i(f(\mathbf{x})) \leq 1$ for every variable x_i, we say that the polynomial $f(\mathbf{x})$ is a *multilinear* polynomial. Moreover, if $f(\mathbf{x})$ is multilinear and the variables in $\mathbf{x}$ can be partitioned into sets $\mathbf{x}_1, \ldots, \mathbf{x}_d$ such that each monomial from $f(\mathbf{x})$ has at most one variable from each of the sets $\mathbf{x}_i$, we say that $f(\mathbf{x})$ is a *set-multilinear* polynomial.

▶ **Definition 5** (Homogeneous Components)**.** For a polynomial $f(\mathbf{x})$, denote its homogeneous part of degree t by $H_t[f(\mathbf{x})]$. Additionally, define

$$H_{\leq t}[f(\mathbf{x})] = \sum_{i=0}^{t} H_i[f(\mathbf{x})],$$

that is, $H_{\leq t}[f]$ is the sum of the homogeneous components of $f(\mathbf{x})$ up to degree t. We can extend this definition to matrices of polynomials in the natural way. Namely, if $\mathbf{f}(\mathbf{x})$ is a matrix of polynomials of the form $(f_{ij}(\mathbf{x}))_{i,j}$, we define $H_t[\mathbf{f}(\mathbf{x})] = (H_t[f_{ij}(\mathbf{x})])_{i,j}$, that is, $H_t[\mathbf{f}(\mathbf{x})]$ is the matrix given by the homogeneous components of degree t of each entry of $\mathbf{f}(\mathbf{x})$.

▶ **Definition 6** (Homogeneous Set Multilinear Components)**.** Let $\mathbf{x} = (\mathbf{x}_1, \ldots, \mathbf{x}_d)$ be a set of variables, partitioned into d sets of variables $\mathbf{x}_1, \ldots, \mathbf{x}_d$. For a polynomial $f(\mathbf{x})$ of degree d, let $H_S^{SM}[f(\mathbf{x})]$ denote its homogeneous set-multilinear part corresponding to subpartition $S \subseteq [d]$. That is, $H_S^{SM}[f(\mathbf{x})]$ consists of the sum of all monomials (with the appropriate coefficients) of $f(\mathbf{x})$ of degree exactly $|S|$ which are set-multilinear with respect to the partition $(\mathbf{x}_i)_{i \in S}$.

The following lemma tells us that any nonzero polynomial cannot vanish on a large portion of any sufficiently large grid.

▶ **Lemma 7** (Schwartz-Zippel-DeMillo-Lipton [47, 54, 13])**.** *Let $\mathbb{F}$ be any field such that $|\mathbb{F}| > d$ and let $S \subseteq \mathbb{F}$ be such that $|S| > d$. If $p(\mathbf{x}) \in \mathbb{F}[\mathbf{x}]$ is a nonzero polynomial of degree d, then*

$$\Pr_{\mathbf{a} \in S^n}[p(\mathbf{a}) = 0] \leq \frac{d}{|S|}.$$

A.2 Matrix Spaces

In this section, we introduce the concept of matrix spaces and establish some of their important properties which we will use in the next sections. We begin by establishing some notations for matrices and tensors.

If V is a vector space of dimension n over a field $\mathbb{F}$, we can identify $V = \mathbb{F}^n$. In this case, we denote the d^{th} tensor power of V by $\text{Ten}_{n,d}(\mathbb{F}) = V^{\otimes d}$. We denote the space of $n \times n$ matrices $V^{\otimes 2}$ by $\text{Mat}_n(\mathbb{F}) = \text{Ten}_{n,2}(\mathbb{F})$. Sometimes we will abuse notation and write $\text{Mat}_n(R)$ for the ring of matrices whose entries take value over a ring R.

A tensor $T \in \text{Ten}_{n,d}(\mathbb{F})$ is a rank-1 tensor if it can be written in the form $T = \mathbf{v}_1 \otimes \cdots \otimes \mathbf{v}_d$, where each $\mathbf{v}_i \in \mathbb{F}^n$. Given any tensor $T \in \text{Ten}_{n,d}(\mathbb{F})$, its rank over $\mathbb{F}$ (denoted by $\text{rank}_{\mathbb{F}}(T)$) is the minimum number r of rank-1 tensors $T_1, \ldots, T_r$ such that $T = T_1 + \cdots + T_r$. Whenever the base field is clear from context, we will denote $\text{rank}_{\mathbb{F}}(T)$ simply by $\text{rank}(T)$.

If $M_1, \ldots, M_k$ are matrices in $\text{Mat}_m(\mathbb{F})$ and $x_1, \ldots, x_k$ are commuting variables, we denote $\text{rank}_{\mathbb{F}(x_1,\ldots,x_k)}(\sum_{i=1}^k x_i M_i)$ the *symbolic rank* of the matrix $\sum_{i=1}^k x_i M_i$.

▶ **Definition 8** (Rank of a Set of Matrices)**.** If $\mathcal{M} \subset \mathrm{Mat}_m(\mathbb{F})$ is a set of $m \times m$ matrices over the field $\mathbb{F}$, define

$$\mathrm{rank}(\mathcal{M}) = \max_{M \in \mathcal{M}} \mathrm{rank}(M).$$

That is, the rank of the set $\mathcal{M}$ is given by the maximum rank (over $\mathbb{F}$) among its elements.

The symbolic rank is important as it characterizes the rank of a linear space of matrices, as seen in the following proposition.

▶ **Proposition 9.** *Let $\mathcal{M} \subseteq \mathrm{Mat}_m(\mathbb{F})$ be a space of matrices. If $M_1, \ldots M_m$ is a basis for $\mathcal{M}$ and $x_1, x_2 \ldots x_m$ are variables then*

$$\mathrm{rank}(\mathcal{M}) = \mathrm{rank}_{\mathbb{F}(x_1,\ldots x_m)} \left(\sum_{i=1}^{m} x_i M_i \right).$$

The propostion above, together with Lemma 7, imply the following lemma:

▶ **Lemma 10** (Rank Upper Bound on Polynomial Matrices)**.** *Let $\mathbf{x} = (x_1, \ldots, x_n)$. If $M(\mathbf{x}) \in \mathrm{Mat}_m(\mathbb{F}[\mathbf{x}])$ is a matrix such that $\mathrm{rank}_{\mathbb{F}}(M(\mathbf{a})) \leq r$ for all $\mathbf{a} \in \mathbb{F}^n$, then $\mathrm{rank}_{\mathbb{F}(\mathbf{x})}(M(\mathbf{x})) \leq r$.*

The following proposition shows one way in which a linear space of matrices is of low rank. This decomposition and its variants will be very useful to us throughout the paper.

▶ **Proposition 11.** *Let $\mathcal{M} \subset \mathrm{Mat}_m(\mathbb{F})$ be a vector space of matrices such that $\mathcal{M} = \mathsf{span}(U \otimes V)$, where $U, V \subset \mathbb{F}^m$ are vector spaces of dimensions r and s, respectively. Then,*

$$\mathrm{rank}(\mathcal{M}) = \min(r, s).$$

A.3 Coefficient Spaces and Their Properties

As we saw in Section 1.3, linear spaces of matrices may possess special structure if they are generated by the coefficients of a matrix of polynomials. This observation, together with the definition below, are crucial in obtaining upper bounds for the rank techniques which we study.

▶ **Definition 12** (Coefficient Space)**.** Let $M(\mathbf{x}) \in \mathbb{F}[\mathbf{x}]^{m \times k}$ be a symbolic matrix of polynomials. Considering the monomial basis $\{\mathbf{x}^{\mathbf{e}}\}_{\mathbf{e} \in \mathbb{N}^n}$ for the space $\mathbb{F}[\mathbf{x}]$, we can write $M(\mathbf{x}) = \sum_{\mathbf{e} \in \mathbb{N}^n} M_{\mathbf{e}} \cdot \mathbf{x}^{\mathbf{e}}$, where each $M_{\mathbf{e}} \in \mathbb{F}^{m \times k}$ is a matrix of field elements. We define the *coefficient space* of $M(\mathbf{x})$, denoted by $\mathcal{C}(M(\mathbf{x}))$, as the vector space spanned by the vectors $M_{\mathbf{e}}$. That is,

$$\mathcal{C}(M(\mathbf{x})) = \mathsf{span}\{M_{\mathbf{e}} \mid \mathbf{e} \in \mathbb{N}^n\}.$$

Note that $\mathcal{C}(M(\mathbf{x})) \subseteq \mathbb{F}^{m \times k}$.

Having the definition above, we proceed to show some nice properties of the coefficient space of a matrix of polynomials.

▶ **Proposition 13.** *Let $\mathbf{x} = (\mathbf{x}_1, \ldots, \mathbf{x}_d)$ be a set of nd variables, partitioned into d sets of n variables each, denoted by $\mathbf{x}_i$. If $\mathbf{f}(\mathbf{x}) \in \mathbb{F}[\mathbf{x}]^m$ is a vector of homogeneous and set-multilinear polynomials of degree d, with respect to the partition $\mathbf{x} = (\mathbf{x}_1, \ldots, \mathbf{x}_d)$, then*

$$\dim(\mathcal{C}(\mathbf{f}(\mathbf{x}))) \leq n^d.$$

By using this new proposition and Propostion 11, we have the following corollary:

▶ **Corollary 14.** *Let $\mathbf{x} = (\mathbf{x}_1, \ldots, \mathbf{x}_d)$ be a set of nd variables, partitioned into d sets of n variables each, denoted by $\mathbf{x}_i$. Additionally, let $S_f \sqcup S_g = [d]$ be a partition of the set $[d]$ such that $|S_f| = d_f$ and $|S_g| = d_g$. If $\mathbf{f}(\mathbf{x}), \mathbf{g}(\mathbf{x}) \in \mathbb{F}[\mathbf{x}]^m$ are vectors of homogeneous set-multilinear polynomials, where $\mathbf{f}(\mathbf{x})$ is partitioned with respect to the variables $(\mathbf{x}_i)_{i \in S_f}$ and $\mathbf{g}(\mathbf{x})$ is partitioned with respect to the variables $(\mathbf{x}_i)_{i \in S_g}$, then we have:*

$$\operatorname{rank}(\mathcal{C}(\mathbf{f}(\mathbf{x}) \otimes \mathbf{g}(\mathbf{x}))) \leq \min\left\{n^{d_f}, n^{d_g}\right\}.$$

B Restricted Forms of Symbolic Matrix Rank Decompositions

If some matrix M over a field $\mathbb{F}$ has rank r, then we can write M as sum of r matrices $M = M_1 + \ldots + M_r$, where each M_i is a rank one matrix over $\mathbb{F}$, and thus can be written as $M_i = \mathbf{u}_i \otimes \mathbf{v}_i$, where $\mathbf{u}_i, \mathbf{v}_i$ are vectors over $\mathbb{F}$. In this section we would like to discus what happens when we impose additional conditions on the matrix M and on the rank one matrices M_i.

For instance, let $M(\mathbf{x}) \in \mathrm{Mat}_m(\mathbb{F}[\mathbf{x}])$ be a matrix of homogeneous polynomials of degree d such that $\operatorname{rank}_{\mathbb{F}(\mathbf{x})}(M) = r$. We want to know the minimal r' such that $M(\mathbf{x})$ can be written as sum of r' matrices $M_i(\mathbf{x})$ of rank one, where each $M_i(\mathbf{x})$ decomposes as $\mathbf{u}_i(\mathbf{x}) \otimes \mathbf{v}_i(\mathbf{x})$ for $\mathbf{u}_i(\mathbf{x}), \mathbf{v}_i(\mathbf{x}) \in \mathbb{F}[\mathbf{x}]^m$ being vectors of homogeneous polynomials. Notice that this decomposition imposes the condition that the vectors $\mathbf{u}_i(\mathbf{x}), \mathbf{v}_i(\mathbf{x})$ be vectors of polynomials, whereas in the general rank decomposition these vectors could be vectors of rational functions, that is, elements of $\mathbb{F}(\mathbf{x})^m$.

In this section, we define one non-standard notion of rank, along with some properties which will be useful to us in the main sections of the paper. We begin with the definition of set-multilinear rank.

▶ **Definition 15** (Set-Multilinear Rank)**.** Let $\mathbf{x} = (\mathbf{x}_1, \ldots, \mathbf{x}_d)$ be a set of variables, partitioned into sets of variables $\mathbf{x}_i$, and $M(\mathbf{x}) \in \mathrm{Mat}_m(\mathbb{F}[\mathbf{x}])$ be a matrix with polynomial entries such that each entry $M_{ij}(\mathbf{x})$ is a homogeneous set-multilinear polynomial of degree d, where the partition is given by $\mathbf{x}$.
The *set-multilinear rank* of $M(\mathbf{x})$, denoted by $\text{sm-rank}(M(\mathbf{x}))$, is the smallest integer r for which there exist r pairs of vectors $\mathbf{f}_i(\mathbf{x}), \mathbf{g}_i(\mathbf{x}) \in \mathbb{F}[\mathbf{x}]^m$ such that

$$M(\mathbf{x}) = \sum_{i=1}^{r} \mathbf{f}_i(\mathbf{x}) \otimes \mathbf{g}_i(\mathbf{x}), \tag{1}$$

where:

- $\mathbf{f}_i(\mathbf{x})$ and $\mathbf{g}_i(\mathbf{x})$ are homogeneous vectors of set-multilinear polynomials,
- for each $i \in [r]$, there exists a partition $S_f^i \sqcup S_g^i = [d]$ of the set $[d]$ such that $\mathbf{f}_i(\mathbf{x})$ is set-multilinear with respect to the variables $(\mathbf{x}_j)_{j \in S_f^i}$ and $\mathbf{g}_i(\mathbf{x})$ is set-multilinear with respect to the variables $(\mathbf{x}_j)_{j \in S_g^i}$.

In particular, $\deg(\mathbf{f}_i(\mathbf{x})) + \deg(\mathbf{g}_i(\mathbf{x})) = d$.

Now that we have a notion of rank, we will need the following decomposition lemma to prove that low rank matrices must also have low set-multilinear rank. Let $M(\mathbf{x}) \in \mathrm{Mat}_m(\mathbb{F}[\mathbf{x}])$ be a matrix whose entries are homogeneous polynomials of degree d. The following lemma shows that if $\operatorname{rank}(M(\mathbf{x})) = r$, then it can be written as the homogeneous component of degree d of a sum of r rank one matrices with polynomial entries.

▶ **Lemma 16** (Symbolic Matrix Decomposition Lemma). *Let $M(\mathbf{x}) \in \mathrm{Mat}_m(\mathbb{F}[\mathbf{x}])$ be a matrix of homogeneous polynomials of degree d. If $\mathrm{rank}_{\mathbb{F}(\mathbf{x})}(M(\mathbf{x})) = r$ then there are vectors $\mathbf{f}_1(\mathbf{x}), \ldots, \mathbf{f}_r(\mathbf{x}) \in \mathbb{F}[\mathbf{x}]^m$ and $\mathbf{g}_1(\mathbf{x}), \ldots, \mathbf{g}_r(\mathbf{x}) \in \mathbb{F}[\mathbf{x}]^m$ such that*

$$M(\mathbf{x}) = \sum_{i=1}^{r} H_d[\mathbf{f}_i(\mathbf{x}) \otimes \mathbf{g}_i(\mathbf{x})].$$

Proof. Since $\mathrm{rank}_{\mathbb{F}(\mathbf{x})}(M(\mathbf{x})) = r$, there exist r pairs of vectors of polynomials $\mathbf{p}_i(\mathbf{x}), \mathbf{q}_i(\mathbf{x}) \in \mathbb{F}[\mathbf{x}]^m$ and nonzero polynomials $t_i(\mathbf{x}) \in \mathbb{F}[\mathbf{x}]$ such that

$$M(\mathbf{x}) = \sum_{i=1}^{r} \frac{1}{t_i(\mathbf{x})} \mathbf{p}_i(\mathbf{x}) \otimes \mathbf{q}_i(\mathbf{x}).$$

Since $t_i(\mathbf{x})$ are nonzero polynomials for all $i \in [r]$, the polynomial given by $Q(\mathbf{x}) = \prod_{i=1}^{r} t_i(\mathbf{x})$ is a nonzero polynomial. By $\mathrm{char}(\mathbb{F}) = 0$ and Lemma 7, there exists $\mathbf{a} \in \mathbb{F}^n$ such that $Q(\mathbf{a}) \neq 0$. In particular, this implies that we can write $t_i(\mathbf{x} + \mathbf{a}) = b_i \cdot (1 - \hat{t}_i(\mathbf{x}))$, where $b_i \in \mathbb{F}$ are nonzero field elements and $\hat{t}_i(\mathbf{x})$ are polynomials such that $\hat{t}_i(\mathbf{0}) = 0$. Namely, the constant terms of $\hat{t}_i(\mathbf{x})$ are zero, for all $i \in [r]$.

Writing $\widehat{\mathbf{p}}_i(\mathbf{x}) = \mathbf{p}_i(\mathbf{x} + \mathbf{a})$, $\widehat{\mathbf{q}}_i(\mathbf{x}) = \mathbf{q}_i(\mathbf{x} + \mathbf{a})$, and from the power series expansion of $1/(1 - x)$, it follows that

$$\begin{aligned} M(\mathbf{x} + \mathbf{a}) &= \sum_{i=1}^{r} \frac{1}{t_i(\mathbf{x} + \mathbf{a})} \widehat{\mathbf{p}}_i(\mathbf{x}) \otimes \widehat{\mathbf{q}}_i(\mathbf{x}) \\ &= \sum_{i=1}^{r} \frac{1}{b_i \cdot (1 - \hat{t}_i(\mathbf{x}))} \widehat{\mathbf{p}}_i(\mathbf{x}) \otimes \widehat{\mathbf{q}}_i(\mathbf{x}) \\ &= \sum_{i=1}^{r} \frac{1}{b_i} [\widehat{\mathbf{p}}_i(\mathbf{x}) \otimes \widehat{\mathbf{q}}_i(\mathbf{x})] \cdot \left(\sum_{j=0}^{\infty} \hat{t}_i(\mathbf{x})^j \right). \end{aligned}$$

As $M(\mathbf{x} + \mathbf{a})$ is a matrix of polynomials of degree no larger than d, the equality above becomes:

$$\begin{aligned} M(\mathbf{x} + \mathbf{a}) &= H_{\leq d}[M(\mathbf{x} + \mathbf{a})] \\ &= H_{\leq d} \left\{ \sum_{i=1}^{r} \frac{1}{b_i} [\widehat{\mathbf{p}}_i(\mathbf{x}) \otimes \widehat{\mathbf{q}}_i(\mathbf{x})] \cdot \left(\sum_{j=0}^{\infty} \hat{t}_i(\mathbf{x})^j \right) \right\} \\ &= H_{\leq d} \left\{ \sum_{i=1}^{r} \frac{1}{b_i} [\widehat{\mathbf{p}}_i(\mathbf{x}) \otimes \widehat{\mathbf{q}}_i(\mathbf{x})] \cdot \left(\sum_{j=0}^{d} \hat{t}_i(\mathbf{x})^j \right) \right\} \\ &= \sum_{i=1}^{r} H_{\leq d}[\widetilde{\mathbf{p}}_i(\mathbf{x}) \otimes \widetilde{\mathbf{q}}_i(\mathbf{x})], \end{aligned}$$

where $\widetilde{\mathbf{p}}_i(\mathbf{x}) = \frac{1}{b_i} \widehat{\mathbf{p}}_i(\mathbf{x})$ and $\widetilde{\mathbf{q}}_i(\mathbf{x}) = \widehat{\mathbf{q}}_i(\mathbf{x}) \cdot \left(\sum_{j=0}^{d} \hat{t}_i(\mathbf{x})^j \right)$.

Moreover, from homogeneity of $M(\mathbf{x})$, we have $M(\mathbf{x}) = H_d[M(\mathbf{x} + \mathbf{a})]$, which implies

$$M(\mathbf{x}) = H_d[M(\mathbf{x} + \mathbf{a})] = \sum_{i=1}^{r} H_d[\widetilde{\mathbf{p}}_i(\mathbf{x}) \otimes \widetilde{\mathbf{q}}_i(\mathbf{x})].$$

Taking $\mathbf{f}_i(\mathbf{x}) = \widetilde{\mathbf{p}}_i(\mathbf{x})$ and $\mathbf{g}_i(\mathbf{x}) = \widetilde{\mathbf{q}}_i(\mathbf{x})$ completes the proof. ◀

With this concept of set-multilinear decomposition and the lemma above, we obtain the following relationship between the symbolic rank and the set-multilinear rank of a set-multilinear polynomial matrix.

▶ **Lemma 17** (Set-Multilinear Rank of Polynomial Matrices). *Let $M(\mathbf{x}) \in \mathrm{Mat}_m(\mathbb{F}[\mathbf{x}])$ be a set-multilinear matrix of degree d, with partition $\mathbf{x} = (\mathbf{x}_1, \ldots, \mathbf{x}_d)$.*

If $\mathrm{rank}_{\mathbb{F}(\mathbf{x})}(M(\mathbf{x})) \leq r$ then $\text{sm-rank}(M(\mathbf{x})) \leq r \cdot 2^d$.

Proof. W.l.o.g., we can assume that $\mathrm{rank}_{\mathbb{F}(\mathbf{x})}(M(\mathbf{x})) = r$. From Lemma 16, there exist vectors of polynomials $\mathbf{p}_1(\mathbf{x}), \mathbf{q}_1(\mathbf{x}), \ldots, \mathbf{p}_r(\mathbf{x}), \mathbf{q}_r(\mathbf{x}) \in \mathbb{F}[\mathbf{x}]^m$ such that

$$M(\mathbf{x}) = \sum_{i=1}^{r} H_d[\mathbf{p}_i(\mathbf{x}) \otimes \mathbf{q}_i(\mathbf{x})]. \tag{2}$$

Decomposing equality (2) into its homogeneous and set multilinear components, according to the partition $\mathbf{x} = (\mathbf{x}_1, \ldots, \mathbf{x}_d)$ we obtain:

$$M(\mathbf{x}) = \sum_{i=1}^{r} H^{SM}_{[d]}[\mathbf{p}_i(\mathbf{x}) \otimes \mathbf{q}_i(\mathbf{x})] = \sum_{i=1}^{r} \sum_{S \subseteq [d]} H^{SM}_{S}[\mathbf{p}_i(\mathbf{x})] \otimes H^{SM}_{[d]\setminus S}[\mathbf{q}_i(\mathbf{x})].$$

The last line of the equality above giving us the decomposition of $M(\mathbf{x})$ into $R \leq r \cdot 2^d$ rank-1 polynomial matrices. ◀

A Complexity Trichotomy for k-Regular Asymmetric Spin Systems Using Number Theory

Jin-Yi Cai[1], Zhiguo Fu[2], Kurt Girstmair[3], and Michael Kowalczyk[4]

1 **Computer Sciences Department, University of Wisconsin, Madison, WI, USA**
jyc@cs.wisc.edu
2 **School of Computer Science & Information Technology, Northeast Normal University, Changchun, China**
zfu8@wisc.edu
3 **Institute für Mathematik, Universität Innsbruck, Austria**
Kurt.Girstmair@uibk.ac.at
4 **Math & CS Department, Northern Michigan University, Marquette, MI, USA**
mkowalcz@nmu.edu

Abstract

Suppose φ and ψ are two angles satisfying $\tan(\varphi) = 2\tan(\psi) > 0$. We prove that under this condition φ and ψ *cannot* be both rational multiples of π. We use this number theoretic result to prove a classification of the computational complexity of spin systems on k-regular graphs with general (not necessarily symmetric) real valued edge weights. We establish explicit criteria, according to which the partition functions of all such systems are classified into three classes: (1) Polynomial time computable, (2) #P-hard in general but polynomial time computable on planar graphs, and (3) #P-hard on planar graphs. In particular problems in (2) are precisely those that can be transformed to a form solvable by the Fisher-Kasteleyn-Temperley algorithm by a holographic reduction.

1998 ACM Subject Classification F.1.3 Complexity Measures and Classes

Keywords and phrases Spin Systems, Holant Problems, Number Theory, Characters, Cyclotomic Fields

Digital Object Identifier 10.4230/LIPIcs.ITCS.2018.2

1 Introduction

We consider spin systems on finite k-regular graphs $G = (V, E)$. Here every vertex $v \in V$ has degree k, and every edge $(u, v) \in E$ is assigned a constraint function $f : \{0, 1\}^2 \to \mathbb{R}$. The function f is not assumed to be symmetric, and one of u or v is specified as the first input variable of f, and the other one the second. Equivalently one can think of G as a directed graph. Define the *partition function* on G as $Z_f(G) = \sum_{\sigma:V\to\{0,1\}} \prod_{(u,v)\in E(G)} f(\sigma(u), \sigma(v))$. Depending on the nature of the edge function f, we show that the problem $Z_f(\cdot)$ is either computable in polynomial time (denoted as P-time) or #P-hard. Furthermore, for those problems $Z_f(\cdot)$ that are #P-hard in general, if the input is restricted to planar graphs, then some of them become computable in P-time. We prove that for all such problems, it is computable in P-time by a universal algorithm that is a holographic reduction to Kasteleyn's algorithm (this is Valiant's Holographic Algorithm)—all other problems remain #P-hard on planar graphs.[1]

[1] Ladner's theorem [20] states that if P$\neq$ NP, then there are problems in NP that are neither in P nor NP-complete. The same is true for #P. Therefore the assertion that all $Z_f(\cdot)$ can be classified

To prove our classification theorem, we will make an unexpected detour into number theory. To state it in general terms, this came about as follows: In our attempt to prove #P-hardness for some particularly tricky cases, we found a pair of constructions. Each is controlled by a pair of eigenvalues of equal norm. If the ratio of the two eigenvalues is a *root of unity* then an iteration of the construction will end up repeating after a fixed number of steps (up to a scalar). This is undesirable because the Vandermonde matrix corresponding to the construction will have bounded rank, making it unable to perform polynomial interpolation for arbitrarily large instance graphs. On the other hand, if the ratio of eigenvalues is not a root of unity then the Vandermonde matrix corresponding to the construction will have full rank, and we can successfully interpolate, and thus prove #P-hardness for those tricky cases.

Unfortunately, it is indeed possible that the ratio of eigenvalues for either of the two constructions is a root of unity, depending on specific f. Having unit norm, being a root of unity is the same as the complex argument being a rational multiple of π. As it turns out, the pair of constructions we found has the following surprising property: If the complex arguments (of the ratio of eigenvalues) of the two constructions are φ and ψ respectively, then the tangent values of φ and ψ satisfy the equation $\tan(\varphi) = 2\tan(\psi) > 0$, in all settings of f. So if we can show, given that $\tan(\varphi) = 2\tan(\psi) > 0$, it is impossible that *both* φ and ψ are rational multiples of π, then we will have proved that in all cases at least one of the two constructions succeeds. This is indeed true and we prove it in Theorem 1.

Proving this rational incommensurability between two tangent values, and at the same time, their angle values divided by π, and then using it to prove the complexity classification is the most surprising aspect of this paper. For any fixed n, questions regarding $\mathbb{Q}$-linear independence among cotangent values of the form $\cot(k\pi/n)$ (for $1 \le k < n/2$ and $\gcd(k, n) = 1$) were first suggested by Chowla and proved by Siegel in 1949 (reported by Chowla [12] in 1964; see also [13]). For any fixed prime p, theorems of this type were found for tangent values $\tan(k\pi/p)$ by Hasse [17], and for cosecant values $\csc(2k\pi/p)$ by Jager and Lenstra [19] ($1 \le k \le (p-1)/2$), although linear dependence for the latter case is possible. For any n, Girstmair gave a representation theoretic treatment to the problem of determining $\mathbb{Q}$-linear relations among numbers of the form, respectively, $\cot(k\pi/n), \tan(k\pi/n), \csc(2k\pi/n)$ or $\sec(2k\pi/n)$, for $\gcd(k, n) = 1$ [15]. While these results do not directly imply what we need (Theorem 1), our proof uses a crucial formula in [15] (Theorem 2, p. 380) regarding Leopoldt's character coordinates of numbers in a number field. (Note that Siegel's theorem [12] does *not*, in view of the requirement $\gcd(k, n) = 1$, imply Theorem 1 because there may not be a common primitive order n for φ and ψ; furthermore, $\cot(\pi/6) = 3\cot(\pi/3)$ provides a counter example to the more general statement of $\mathbb{Q}$-linear independence.)

There have been a number of classification theorems for #CSP and related problems [4, 5, 6, 14, 9, 18, 16, 7, 22, 3]. Spin systems are special cases of #CSP (with a single edge function), and #CSP are special cases of Holant problems in which Equality functions of all arities are assumed to be present. The problem addressed in this paper can be viewed as only allowing Equality function of a fixed arity (regular graphs). Without all Equality functions reduction proofs become more challenging. The immediate predecessors to the present work are the classification for $Z_f(\cdot)$ for k-regular graphs where f is a *symmetric* edge function [10], and the classification for $Z_f(\cdot)$ for 3-regular graphs where f is not necessarily symmetric [11]. There are technical difficulties generalizing the proof in [10, 11] to 4-regular

into either P-time computable or #P-hard is not self-evident. To state our results strictly in Turing machine-based complexity theory, f takes values in algebraic numbers.

graphs with an asymmetric edge function. On the other hand, aside from its intrinsic interest, spin systems on k-regular graphs for *even* k have another pertinence. Although we do not intend to elaborate it here, the result in this paper fits in a bigger classification program for sum-of-product computations. In particular, to classify all Holant problems, a natural process is arity reduction by taking self loops and some similar operations. This reduces the arity by two, and thus there are two base cases in an inductive proof, arity 3 and arity 4. Often one can holographically transform such a signature to EQUALITY of arity 3 or 4 respectively, which gives rise to a spin system on 3- or 4-regular graphs.

This type of sum-of-product computations is studied in physics, where the term partition function originated. In physics, the 0-1 vertex assignments are called spins, and the edge function values $f(\sigma(u), \sigma(v))$ correspond to local interactions between particles. There is a long history in the statistical physics in the study of "Exactly Solved Models" [2, 26]. A rough correspondence exists between P-time computability and physicists' notion of an "Exactly Solvable" system. A central question is to identify which "systems" can be solved "exactly" and which "systems" are "difficult". While in physics there is no rigorous definition of being "difficult", complexity theory proposes that the right notion is #P-hardness.

This paper is organized as follows: In Section 2 we prove Theorem 1 to establish the incommensurability of (co)tangent values and angle values over π. In Section 3 we state some definitions and needed results. In Section 4 we prove the classification theorem for 4-regular graphs. In Section 5 we prove the classification theorem for k-regular graphs, for all k.

2 A Theorem in Number Theory

Let $0 < \varphi < \psi < \pi/2$ denote two angles. Then $0 < \cot(\psi) < \cot(\varphi) < \infty$. Is it possible that

$$\cot(\varphi) = 2\cot(\psi), \tag{1}$$

and yet φ and ψ are both rational multiples of π? We prove the following theorem. It says that, with exactly one obvious exception, it is *not* possible that *both* the ratio of the cotangent values of φ and ψ is rational, *and* the two angles are rational multiples of π. In particular (1) is not possible when both φ and ψ are rational multiples of π. This incommensurability will be used to prove a key complexity reduction to reach our complexity trichotomy classification.

▶ **Theorem 1.** *Suppose* $0 < \varphi < \psi < \pi/2$, *and* $\cot(\varphi) = r\cot(\psi)$, *for some* $r \in \mathbb{Q}$ *and* $r \neq 3$. *Then* φ *and* ψ *are not both rational multiples of* π.

Proof. We first note that the exception $r = 3$ is witnessed by $\cot(\pi/6) = 3\cot(\pi/3)$.

We write $r = \frac{a}{b}$ for relatively prime integers a and b. We are given $\cot(\varphi) = \frac{a}{b}\cot(\psi)$. For a contradiction, suppose φ and ψ are both rational multiples of π, and we write $\varphi = \frac{k\pi}{n}$ and $\psi = \frac{k'\pi}{n'}$, where $1 \leq k < \frac{n}{2}$, $1 \leq k' < \frac{n'}{2}$, and $\gcd(k, n) = \gcd(k', n') = 1$.

Let $\zeta_n = \exp(2\pi\mathrm{i}/n)$ be a primitive root of unity. Then it is easy to verify that $\mathrm{i}\cot(\varphi) = \frac{1+\zeta_n^k}{1-\zeta_n^k}$. If we write $t = \mathrm{i}\cot(\varphi)$, then $t \in \Phi_n = \mathbb{Q}(\zeta_n)$, the n-th cyclotomic field (the field extension by adjoining ζ_n to $\mathbb{Q}$). Also $\zeta_n^k = \frac{t-1}{t+1}$. As $\gcd(k, n) = 1$, we have $\Phi_n = \mathbb{Q}(\zeta_n^k) \subseteq \mathbb{Q}(t) \subseteq \Phi_n$, and so $\mathbb{Q}(t) = \Phi_n$.

By $\cot(\varphi) = \frac{a}{b}\cot(\psi)$, we have $\Phi_{n'} = \Phi_n$. It is well known that this implies that either $n = n'$, or n is odd and $n' = 2n$, or n' is odd and $n = 2n'$.

We first consider the case $n = n'$. This case actually follows from Siegel's theorem [12]. For a uniform treatment we give a direct proof here.

For $n = n'$, we have $1 \le k < k' < \frac{n}{2}$, and so $n \ge 5$. A Dirichlet character χ to the modulus n is a function from $\mathbb{Z}$ to $\mathbb{C}$ that is multiplicative, has period n, and $\chi(j) \neq 0$ iff $\gcd(j, n) = 1$. So χ is extended from a group character on $\mathbb{Z}_n^\times$ (i.e., a multiplicative function taking nonzero values in $\mathbb{C}$ on $\mathbb{Z}_n^\times = \{(j \bmod n) \in \mathbb{Z}_n \mid \gcd(j, n) = 1\}$), and all nonzero values of χ are roots of unity. A Dirichlet character χ is said to be *odd* if $\chi(-1) = -1$.

We need to take an odd Dirichlet character χ to the modulus n. An odd Dirichlet character χ (for $n > 2$) exists: The group of Dirichlet characters mod n is isomorphic to $\mathbb{Z}_n^\times$. Since $n > 2$, $\{1, -1\}$ is a subgroup of order two in $\mathbb{Z}_n^\times$. The subgroup of even characters is isomorphic to $\mathbb{Z}_n^\times/\{1, -1\}$. Hence, not every Dirichlet character mod n is even. A more constructive proof is as follows: The character group of $\mathbb{Z}_n^\times$, by Chinese remaindering, is a direct product of the character groups of $\mathbb{Z}_{p_i^{e_i}}^\times$ according to the prime factorization $n = \prod_i p_i^{e_i}$. For each odd prime p_i, the group $\mathbb{Z}_{p_i^{e_i}}^\times$ is cyclic of even order $m = \phi(p_i^{e_i}) = (p_i - 1)p_i^{e_i - 1}$. Let ρ be a generator, then $\rho^{m/2} = -1$. Thus we can define a character χ on $\mathbb{Z}_{p_i^{e_i}}^\times$ by $\chi(\rho^j) = \zeta_m^j$. Then $\chi(-1) = -1$. If $4 \mid n$, then $\mathbb{Z}_n^\times$ has a factor $\mathbb{Z}_{2^e}^\times$ for $e \ge 2$, which is isomorphic to $\mathbb{Z}_2 \oplus \mathbb{Z}_{2^{e-2}}$ as an additive group, with generators $\{-1, 5\}$. Thus every $j \in \mathbb{Z}_{2^e}^\times$ is uniquely expressed as $(-1)^u 5^v \bmod 2^e$, for $u = 0, 1$, and $0 \le v < 2^{e-2}$. Then an odd character χ on $\mathbb{Z}_{2^e}^\times$ can be defined by $\chi((-1)^u 5^v) = (-1)^u$. If $n \equiv 2 \bmod 4$, then $\mathbb{Z}_n^\times$ has a trivial factor $\mathbb{Z}_2^\times = 1$, and the character group of $\mathbb{Z}_n^\times$ is isomorphic to that of $\mathbb{Z}_{n/2}^\times$, where $n/2 > 1$ is odd. Hence by Chinese remaindering, we can define an odd Dirichlet character χ on $\mathbb{Z}_n^\times$.

An important notion we will use in this proof is that of Leopoldt's character coordinates [21, 15]. In our case, for any odd Dirichlet character χ to the modulus n, the following can be taken as the definition of Leopoldt's character coordinates $y(\chi \mid t) \in \mathbb{C}$, for $t = \mathrm{i}\cot(\frac{k\pi}{n}) \in \Phi_n$,

$$y(\chi \mid t)\ \tau(\overline{\chi_d} \mid 1) = \sum_{j=1}^{n} \overline{\chi(j)}\sigma_j(t), \tag{2}$$

where d is the *conductor* of χ, [2] χ_d is the induced primitive character of χ mod d, overline denotes complex conjugation, the value $\tau(\overline{\chi_d} \mid 1) = \sum_{j=1}^{d} \overline{\chi_d}(j)e^{-2\pi \mathrm{i} j/d}$ is the Gauss sum, and σ_j is the automorphism in the Galois group $G = \mathrm{Gal}(\Phi_n/\mathbb{Q})$ that maps ζ_n to ζ_n^j. The sum $\sum_{j=1}^{n} \overline{\chi(j)}\sigma_j(t)$ is actually over relatively prime integers $j \in \mathbb{Z}_n^\times$, since otherwise the Dirichlet character $\chi(j) = 0$. In the expression $t = \mathrm{i}\cot(\frac{k\pi}{n}) \in \Phi_n$, we have $\gcd(k, n) = 1$, and so $\sigma_k \in G$, and $t = \sigma_k(t_1)$ where $t_1 = \mathrm{i}\cot(\frac{\pi}{n})$. For any fixed $k \in \mathbb{Z}_n^\times$, $\sigma_j \circ \sigma_k = \sigma_{jk}$ runs through all G when j runs through $\mathbb{Z}_n^\times$. Then

$$\sum_{j=1}^{n} \overline{\chi(j)}\sigma_j(t) = \chi(k) \sum_{j \in \mathbb{Z}_n^\times} \chi(k)^{-1}\overline{\chi(j)}\sigma_j(\sigma_k(t_1)) = \chi(k) \sum_{j \in \mathbb{Z}_n^\times} \overline{\chi(kj)}\sigma_{jk}(t_1) = \chi(k) \sum_{j \in \mathbb{Z}_n^\times} \overline{\chi(j)}\sigma_j(t_1).$$

Hence $y(\chi \mid t) = \chi(k)\ y(\chi \mid t_1)$, as the Gauss sum $\tau(\overline{\chi_d} \mid 1) \neq 0$. Similarly for $t' = \mathrm{i}\cot(\frac{k'\pi}{n})$, (recall we have $n' = n$), $y(\chi \mid t') = \chi(k')\ y(\chi \mid t_1)$. Hence the norm of the two Leopoldt's character coordinates are equal, $|y(\chi \mid t)| = |y(\chi \mid t')|$. However if

[2] See [1], p. 167–p. 171 for the definitions of induced characters and the conductor of a character. In number theory it is traditional to denote the conductor of a character by f as is written in [15]; we use d here in order not to confuse it with the constraint function $f(u, v)$ in Section 1.

$t' = rt$, where $0 < r \in \mathbb{Q}$, then $y(\chi \mid t') = ry(\chi \mid t)$ by (2), and so $r = 1$ (we will see that $y(\chi \mid t) = \chi(k)\ y(\chi \mid t_1) \neq 0$). This contradicts $r > 1$, which is a consequence of $\cot(\varphi) > \cot(\psi)$ by $0 < \varphi < \psi < \pi/2$.

Now we assume n is odd and $n' = 2n$. We want to take an odd Dirichlet character χ to the modulus $2n$. Since n is odd, the character groups of $\mathbb{Z}_n^\times$ and $\mathbb{Z}_{2n}^\times$ are isomorphic, namely for every $j \in \mathbb{Z}_n^\times$ exactly one of j or $j+n$ is odd and so belongs to $\mathbb{Z}_{2n}^\times$. Since $n > 1$ is odd, in its prime factorization $n = \prod_i p_i^{e_i}$, every p_i is odd. Then $\mathbb{Z}_n^\times \cong \prod_i \mathbb{Z}_{p_i^{e_i}}^\times$, and every $\mathbb{Z}_{p_i^{e_i}}^\times$ is cyclic of even order $\phi(p_i^{e_i}) = (p_i - 1)p_i^{e_i - 1}$. So we can define an odd Dirichlet character χ on $\mathbb{Z}_n^\times$ by Chinese remaindering, by defining it to be odd on each $\mathbb{Z}_{p_i^{e_i}}^\times$, namely $\chi(-1) = -1$. In particular there is an odd character χ to the modulus $2n$. Since n is an induced modulus, and odd, the conductor d of χ is also odd.

Take any odd Dirichlet character χ mod $2n$. It is proved in [15] (Theorem 2, p. 380) that

$$y(\chi \mid \mathrm{i}\cot(\frac{\pi}{2n})) = \frac{4n}{d} \prod_{p \mid 2n} \left(1 - \frac{\overline{\chi_d(p)}}{p}\right) B_{\chi_d}, \tag{3}$$

and

$$y(\chi \mid \mathrm{i}\cot(\frac{\pi}{n})) = \frac{2n}{d} \prod_{p \mid n} \left(1 - \frac{\overline{\chi_d(p)}}{p}\right) B_{\chi_d}. \tag{4}$$

Here B_{χ_d} is the generalized Bernoulli number. (Eqn. (4) is proved in Theorem 2 of [15] for any non-principal χ mod n without requiring n being odd, and so the proof below that $y(\chi \mid \mathrm{i}\cot(\frac{\pi}{n})) \neq 0$ is also valid for the previous case $n = n'$.)

By definition the Bernoulli polynomial $B(Z)$ is the first $B^{(1)}(Z)$ defined by

$$\frac{te^{Zt}}{e^t - 1} = \sum_{m=0}^{\infty} B^{(m)}(Z)t^m/m!. \tag{5}$$

And the generalized Bernoulli number B_{χ_d} is defined by

$$\sum_{j=1}^{d} \chi_d(j) \frac{te^{jt}}{e^{dt} - 1} = \sum_{m=0}^{\infty} B_{\chi_d}^{(m)} t^m/m!, \tag{6}$$

with $B_{\chi_d} = B_{\chi_d}^{(1)}$. It follows immediately from (5) (and is also well known) that $B(Z) = Z - \frac{1}{2}$. Substituting t by dt and Z by j/d in (5), we get the following equality from (5) and (6)

$$B_{\chi_d} = \sum_{j=1}^{d} \chi_d(j) B(j/d). \tag{7}$$

It follows easily from the definition that $\sum_{j=1}^{d} \chi_d(j) = 0$. (This uses the fact that χ_d is not principal, namely not identically 1 on $\mathbb{Z}_d^\times = \{j \bmod d \mid \gcd(j,d) = 1\}$; indeed $\chi_d(-1) = -1$, and $\chi_d(j) = 0$ if $\gcd(j,d) > 1$, and so $\sum_j' \chi_d(j) = \sum_j' \chi_d(-j) = -\sum_j' \chi_d(j)$, where each sum $\sum_j'$ is over $\mathbb{Z}_d^\times$.) It follows that $B_{\chi_d} = \sum_{j=1}^{d} \chi_d(j) j/d$.

It is a nontrivial fact that $\sum_{j=1}^{d} \chi_d(j) j \neq 0$ for any odd character χ_d (see [28] Theorem 4.9, p. 37). Hence $B_{\chi_d} \neq 0$, and therefore also $y(\chi \mid \mathrm{i}\cot(\frac{\pi}{n})) \neq 0$ and $y(\chi \mid \mathrm{i}\cot(\frac{\pi}{2n})) \neq 0$.

For $t' = \mathrm{i}\cot(\frac{k'\pi}{2n})$ and $t = \mathrm{i}\cot(\frac{k\pi}{n})$, it follows from (3) and (4) that

$$\begin{aligned} y(\chi \mid t') &= \chi(k')\ y(\chi \mid \mathrm{i}\cot(\frac{\pi}{2n})) \\ &= \chi(k')2\left(1 - \frac{\overline{\chi_d(2)}}{2}\right)\ \frac{2n}{d}\prod_{p|n}\left(1 - \frac{\overline{\chi_d(p)}}{p}\right) B_{\chi_d} \\ &= \chi(k')\left(2 - \overline{\chi_d(2)}\right) y(\chi \mid \mathrm{i}\cot(\frac{\pi}{n})) \\ &= \chi(k')\overline{\chi(k)}\left(2 - \overline{\chi_d(2)}\right) y(\chi \mid t) \end{aligned}$$

On the other hand, since by assumption $t = \frac{a}{b}t'$ for integers a and b, we have

$$y(\chi \mid t) = \frac{a}{b}\ y(\chi \mid t').$$

Hence, by being nonzero, and taking the norm squared, we get

$$b^2 = a^2 \cdot |2 - \overline{\chi_d(2)}|^2. \tag{8}$$

Since χ_d is primitive mod d, and d is odd, we have $\rho = \chi_d(2) \neq 0$. Denote this root of unity by $\rho = \chi_d(2)$. We have

$$b^2 = a^2[5 - 2(\rho + \overline{\rho})].$$

If we started with n' odd and $n = 2n'$, we would have the same equation with a and b exchanged.

$$a^2 = b^2[5 - 2(\rho + \overline{\rho})].$$

If $\rho = 1$ then $a = b$, this is a contradiction to $\varphi \neq \psi$. If $\rho = -1$ then $b^2 = 9a^2$ or $a^2 = 9b^2$. This gives us the unique exceptional case $\varphi = \pi/6$ and $\psi = \pi/3$.

Back to $n' = 2n$ with n odd; the other case being symmetric. Suppose $\rho \neq \pm 1$, then it is a nonreal algebraic integer, and satisfies the equation $2a^2(\rho^2 + 1) = \rho(5a^2 - b^2)$. Its minimal polynomial is monic with integer coefficients. Hence $2a^2 \mid (5a^2 - b^2)$. Hence $a \mid b$. Since $\gcd(a, b) = 1$, we get $a = 1$. Back to (8) we get $b < 3$, since $\rho \neq \pm 1$. And so $b = 2$. But in this case the solution $(1 \pm \sqrt{15}\mathrm{i})/4$ to $2(\rho^2 + 1) = \rho$ is not a root of unity. ◀

We will use Theorem 1 to prove a key complexity reduction, stated in Lemma 18, after we formally define Holant problems and reductions in Section 3.

3 Definitions and Known Results

A constraint function f of arity k is a map $\{0,1\}^k \to \mathbb{C}$. Let $\mathcal{F}$ denote a set of constraint functions. A signature grid $\Omega = (G, \pi)$ is a tuple, where $G = (V, E)$ is a graph, π labels each $v \in V$ with a function $f_v \in \mathcal{F}$ of arity $\deg(v)$, and the incident edges $E(v)$ at v with input variables of f_v. We consider all 0-1 edge assignments σ, each gives an evaluation $\prod_{v\in V} f_v(\sigma|_{E(v)})$, where $\sigma|_{E(v)}$ denotes the restriction of σ to $E(v)$. The counting problem on the instance Ω is to compute

$$\mathrm{Holant}_\Omega(\mathcal{F}) = \sum_{\sigma: E\to\{0,1\}} \prod_{v\in V} f_v(\sigma|_{E(v)}).$$

The Holant problem parameterized by the set $\mathcal{F}$ is denoted by Holant($\mathcal{F}$). If the underlying graph is a planar graph, then we denote it by Pl-Holant($\mathcal{F}$). Replacing f by $c \cdot f$ for any $c \neq 0$ only changes the value $\text{Holant}_\Omega(\mathcal{F})$ by c^n where n is the number of times f appears in Ω. Thus it does not change its complexity, therefore we can ignore such constant factors. We also write Holant($\mathcal{F}, f$) for Holant($\mathcal{F} \cup \{f\}$). We use Holant($\mathcal{F}|\mathcal{G}$) to denote the Holant problem over signature grids with a bipartite graph $G = (U, V, E)$, where each vertex in U or V is assigned a signature in $\mathcal{F}$ or $\mathcal{G}$ respectively.

A constraint function is also called a signature. A signature f of arity k can be represented by listing its values in lexicographical order as in a truth table, which is a vector in $\mathbb{C}^{2^k}$, or as a tensor in $(\mathbb{C}^2)^{\otimes k}$. A binary signature $f(x_1, x_2) = (f_{00}, f_{01}, f_{10}, f_{11})$ can be represented as a matrix $M(f) = \begin{bmatrix} f_{00} & f_{01} \\ f_{10} & f_{11} \end{bmatrix}$. A function is symmetric if its value depends only on the Hamming weight of its input. A symmetric function f on k Boolean variables can be expressed as $[f_0, f_1, \ldots, f_k]$, where f_w is the value of f on inputs of Hamming weight w. For example, $(=_k)$ is the EQUALITY signature $[1, 0, \ldots, 0, 1]$ (with $k-1$ 0's) of arity k.

In this paper, we consider the complexity of spin systems on k-regular graphs with real-valued edge functions. This can be defined as Holant problems of the form Holant$(=_k | f)$, where $f(x_1, x_2) = (f_{00}, f_{01}, f_{10}, f_{11}) \in \mathbb{R}^4$ is a binary signature. If $k = 1$, the spin system is a union of disjoint edges (the bipartite vertex-edge incidence graph form for Holant$(=_k | f)$ is a union of disjoint 2-paths). If $k = 2$, the spin system is a union of disjoint cycles. Thus, for $k \leq 2$, the Holant is trivially computable in polynomial time. We assume $k \geq 3$.

For $T \in \text{GL}_2(\mathbb{C})$ and a signature f of arity n, written as a column vector $f \in \mathbb{C}^{2^n}$, we denote by $T^{-1}f = (T^{-1})^{\otimes n} f$ the transformed signature. For a signature set $\mathcal{F}$, define $T^{-1}\mathcal{F} = \{T^{-1}f \mid f \in \mathcal{F}\}$. For signatures written as row vectors we define $\mathcal{F}T$ similarly. The holographic transformation defined by T is the following operation: given a signature grid $\Omega = (H, \pi)$ of Holant $(\mathcal{F} \mid \mathcal{G})$, for the same bipartite graph H, we get a new signature grid $\Omega' = (H, \pi')$ of Holant $\left(\mathcal{F}T \mid T^{-1}\mathcal{G}\right)$ by replacing each signature in $\mathcal{F}$ or $\mathcal{G}$ with the corresponding signature in $\mathcal{F}T$ or $T^{-1}\mathcal{G}$.

▶ **Theorem 2** (Valiant's Holant Theorem [27]). *For any $T \in \text{GL}_2(\mathbb{C})$,*

$$\text{Holant}_\Omega(\mathcal{F} \mid \mathcal{G}) = \text{Holant}_{\Omega'}(\mathcal{F}T \mid T^{-1}\mathcal{G}).$$

Therefore, a holographic transformation does not change the value, and so it does not change the complexity of the Holant problem in the bipartite setting.

3.1 Gadget Construction

One basic notion used throughout the paper is realization. If f is realizable from a set $\mathcal{F}$, then we can freely add f into $\mathcal{F}$ while preserving the complexity. This notion is defined by an $\mathcal{F}$-gate. An $\mathcal{F}$-gate (G, π) is similar with a signature grid for Holant($\mathcal{F}$) except that $G = (V, E, D)$ is a graph with some dangling edges D. The dangling edges define external variables for the $\mathcal{F}$-gate. We name the regular edges in E by $1, 2, \ldots, m$ and the dangling edges in D by $m+1, \ldots, m+n$. Then we can define a function f for this $\mathcal{F}$-gate as

$$f(y_1, \ldots, y_n) = \sum_{x_1, \ldots, x_m \in \{0,1\}} H(x_1, \ldots, x_m, y_1, \ldots, y_n),$$

where $(y_1, \ldots, y_n) \in \{0, 1\}^n$ is an assignment on the dangling edges and $H(x_1, \ldots, y_n)$ is the value of the signature grid on an assignment of all edges in G, which is the product of evaluations at all vertices in V. We also call this function f the signature of the $\mathcal{F}$-gate.

If f is a binary signature, and g has arity $n > 2$, we may connect f to two consecutive variables of g. We call this operation "adding a loop to g using f". This produces a signature of arity $n-2$. Note that this $\{f, g\}$-gate (a gadget construction) is planar.

In an instance of Holant($\mathcal{F} \mid \mathcal{G}$), if we have $(=_2)$ on both sides, then we can move any signature f on one side to another side by connecting one copy of $(=_2)$ to each variable of f. So in this case, we can ignore the bipartite restriction when constructing gadgets.

3.2 Tractable Signature Sets

We define some sets of signatures that are known to define polynomial time computable problems (we call them tractable).

Affine Signatures $\mathcal{A}$

▶ **Definition 3.** Let f be a signature of arity n. We say f has affine support of dimension k if the support of f is an affine subspace of dimension k over $\mathbb{Z}_2$.

▶ **Definition 4.** A signature $f(x_1, \ldots, x_n)$ of arity n is *affine* if it has the form

$$\lambda \cdot \chi_{AX=0} \cdot \mathrm{i}^{Q(X)},$$

where $\lambda \in \mathbb{C}$, $X = (x_1, x_2, \ldots, x_n, 1)$, A is a matrix over $\mathbb{Z}_2$, $Q(x_1, x_2, \ldots, x_n) \in \mathbb{Z}_4[x_1, \ldots, x_n]$ is a quadratic (total degree at most 2) multilinear polynomial with the additional requirement that the coefficients of all cross terms are even, i.e., Q has the form

$$Q(x_1, x_2, \ldots, x_n) = a_0 + \sum_{k=1}^{n} a_k x_k + \sum_{1 \le i < j \le n} 2b_{ij} x_i x_j,$$

and χ is a 0-1 indicator function such that $\chi_{AX=0}$ is 1 iff $AX = 0$ over $\mathbb{Z}_2$. We use $\mathcal{A}$ to denote the set of all affine signatures.

The following lemma is an easy criterion for binary signatures in $\mathcal{A}$.

▶ **Lemma 5.** *Let $f = \lambda(\mathrm{i}^{r_1}, \mathrm{i}^{r_2}, \mathrm{i}^{r_3}, \mathrm{i}^{r_4})$ be a binary signature, where λ is a nonzero constant and $r_i \in \{0, 1, 2, 3\}$, then $f \in \mathcal{A}$ iff $r_1 + r_2 + r_3 + r_4 \equiv 0 \pmod 2$.*

Proof. If we normalize f, by dividing the constant i^{r_1}, whether $f \in \mathcal{A}$ is unchanged, nor is the stated criterion. So we may assume $r_1 = 0$. Then it is easy to check that $f(x_1, x_2) = \lambda \mathrm{i}^{Q(x_1, x_2)}$, where $Q(x_1, x_2) = r_3 x_1 + r_2 x_2 + (r_4 - r_2 - r_3) x_1 x_2$. The lemma follows. ◀

Product-Type Signatures $\mathcal{P}$

▶ **Definition 6.** A signature on a set of variables X is of *product type* if it can be expressed as a product of unary functions, binary equality functions ($[1, 0, 1]$), and binary disequality functions ($[0, 1, 0]$), each on not necessarily disjoint subsets of variables of X. We use $\mathcal{P}$ to denote the set of product-type functions.

For example, the binary signatures $(w, 0, 0, z)$ and $(0, x, y, 0)$ are in $\mathcal{P}$ for any $w, x, y, z \in \mathbb{C}$. If $\det \left[\begin{smallmatrix} w & x \\ y & z \end{smallmatrix}\right] = 0$, then $f = (w, x, y, z) \in \mathcal{P}$ and we say that f is degenerate.

Matchgate Signatures $\mathcal{M}$

Matchgates were introduced by Valiant [27] to give polynomial-time algorithms by the FKT algorithm for a collection of counting problems over planar graphs. We use $\mathcal{M}$ to denote the set of all matchgate signatures and Pl-Holant($\mathcal{M}$) is tractable. In this paper, we only need the following facts about $\mathcal{M}$ (see [8]):

1. A binary signature $f \in \mathcal{M}$ iff $f = (w, 0, 0, z)$ or $f = (0, x, y, 0)$ for any $w, x, y, z \in \mathbb{C}$;
2. The symmetric signature $[1, 1]^{\otimes n} + [1, -1]^{\otimes n} = (=_n)H^{\otimes n} \in \mathcal{M}$ for any positive integer n, where $H = \left[\begin{smallmatrix} 1 & 1 \\ 1 & -1 \end{smallmatrix}\right]$.
3. The symmetric signature $[1, 1]^{\otimes n} + \mathrm{i}^n[1, -1]^{\otimes n} = (=_n)Z^{\otimes n}$ is in $\mathcal{M}$ iff n is even, where $Z = \left[\begin{smallmatrix} 1 & 1 \\ \mathrm{i} & -\mathrm{i} \end{smallmatrix}\right]$.

Moreover, we have the following lemma.

▶ **Lemma 7.** *Let $f = (w, x, y, z)$ be a binary signature, where $w, x, y, z \in \mathbb{C}$.*

- *If $z = \epsilon w$ and $y = \epsilon x$, where $\epsilon = \pm 1$, then $(H^{-1})^{\otimes 2} f \in \mathcal{M}$.*
- *If $z = -\epsilon w$ and $y = \epsilon x$, where $\epsilon = \pm 1$, then $(Z^{-1})^{\otimes 2} f \in \mathcal{M}$.*

Proof. Note that $H^{-1} = \frac{1}{2}H$. Ignoring a constant factor,

$$H^{\otimes 2} f = (w + x + y + z, w - x + y - z, w + x - y - z, w - x - y + z).$$

Then $H^{\otimes 2} f$ satisfies the parity constraint (item **1.** above for the properties of $\mathcal{M}$) and is therefore in $\mathcal{M}$, if either $w = z, x = y$ or $w = -z, x = -y$.

For $(Z^{-1})^{\otimes 2} f$, note that up to a constant factor, $Z^{-1} = H \left[\begin{smallmatrix} 1 & 0 \\ 0 & -\mathrm{i} \end{smallmatrix}\right]$. Thus

$$(Z^{-1})^{\otimes 2} f = H^{\otimes 2} \left[\begin{smallmatrix} 1 & 0 \\ 0 & -\mathrm{i} \end{smallmatrix}\right]^{\otimes 2} f = H^{\otimes 2} f'$$

is in $\mathcal{M}$, where $f' = \left[\begin{smallmatrix} 1 & 0 \\ 0 & -\mathrm{i} \end{smallmatrix}\right]^{\otimes 2} f = (w, -x\mathrm{i}, -\epsilon x\mathrm{i}, \epsilon w)$ transforming it to the first case. ◀

Transformable

▶ **Definition 8.** We say a pair of signature sets $(\mathcal{G}|\mathcal{F})$ is $\mathcal{C}$-transformable for Holant $(\mathcal{G} \mid \mathcal{F})$ if there exists $T \in \mathbf{GL}_2(\mathbb{C})$ such that $\mathcal{G}T \subseteq \mathcal{C}$ and $T^{-1}\mathcal{F} \subseteq \mathcal{C}$.

If Holant($\mathcal{C}$) is tractable and $(\mathcal{G}|\mathcal{F})$ is $\mathcal{C}$-transformable, then Holant$(\mathcal{G}|\mathcal{F})$ is tractable by a holographic transformation.

For example, we consider Pl-Holant$(=_k \mid f)$, where $f = (w, x, y, z)$ with $w, x, y, z \in \mathbb{C}$. If $z = \epsilon w, y = \epsilon x$, where $\epsilon = \pm 1$, then $(=_k \mid f)$ is $\mathcal{M}$-transformable using the holographic transformation H by Lemma 7 and Pl-Holant$(=_k \mid f)$ can be computed in polynomial time by the FKT algorithm. Similarly, if k is even and $z = -\epsilon w, y = \epsilon x$, where $\epsilon = \pm 1$, then Pl-Holant$(=_k \mid f)$ can be computed in polynomial time by the FKT algorithm.

3.3 Some results

In [10], the following trichotomy theorem for k-regular *symmetric* spin systems is given.

▶ **Theorem 9.** *Let $k \geq 3$.* Holant$(=_k \mid f)$*, where $f = [w, x, z]$ is a symmetric binary signature ($w, x, z \in \mathbb{C}$), is* #P*-hard for all $w, x, z \in \mathbb{C}$, except in the following cases, for which the problem is in* P*:*

- *$f \in \mathcal{P}$: $wz = x^2$, or $w = z = 0$, or $x = 0$,*
- *f is $\mathcal{A}$-transformable: $wz = -x^2$ and $w^{4k} = x^{4k} = z^{4k}$.*

If the input is restricted to planar graphs, then another class becomes tractable but everything else remains #P-hard.

- $(=_k \mid f)$ *is* $\mathcal{M}$*-transformable:* $w^k = z^k$.

By Theorem 9, we have the following corollary.

▶ **Corollary 10.** *Let* $f = [w, x, z]$ *be a symmetric binary signature, where* $w, x, z \in \mathbb{C}$*, and* $f \notin \mathcal{P}$*, i.e.,* $wz \neq x^2$*,* $x \neq 0$ *and there is at most one zero in* $\{w, z\}$*. If* $|w| \neq |z|$*, then* Pl-Holant$(=_k \mid f)$*, where* $k \geq 3$*, is #P-hard.*

In [25], a trichotomy theorem for 3-regular asymmetric spin systems is given.

▶ **Theorem 11.** *Suppose* $w, x, y, z \in \mathbb{C}$*. Then* Holant$(=_3 \mid (w, x, y, z))$ *is #P-hard except in the following classes, for which the problem is in* P*.*

- $f \in \mathcal{P}$*:* $wz = xy$*, or* $w = z = 0$ *or* $x = y = 0$*;*
- f *is* $\mathcal{A}$*-transformable:* $wz = -xy, w^6 = \epsilon z^6, x^2 = \epsilon y^2$*, where* $\epsilon = \pm 1$*.*

If the input is restricted to planar graphs, then another class becomes tractable but everything else remains #P-hard.

- $(=_3 \mid f)$ *is* $\mathcal{M}$*-transformable:* $w^3 = \epsilon z^3, x = \epsilon y$*, where* $\epsilon = \pm 1$*.*

By Theorem 11, we have the following corollary.

▶ **Corollary 12.** *Let* $f = (w, x, y, z)$ *be a binary signature, where* $w, x, y, z \in \mathbb{C}$ *and* $f \notin \mathcal{P}$*, i.e.,* $wz \neq xy$ *and there is at most one zero in* $\{w, x, y, z\}$*. If* $|w| \neq |z|$ *or* $|x| \neq |y|$*, then* Pl-Holant$(=_3 \mid f)$ *is #P-hard.*

▶ **Lemma 13.** *Let* f *be a binary signature with the signature matrix* $N = P \left[\begin{smallmatrix} \lambda & 0 \\ 0 & \mu \end{smallmatrix}\right] P^{-1}$*, where* P *is an invertible* 2×2 *matrix. Suppose* $\lambda\mu \neq 0$ *and* $\frac{\lambda}{\mu}$ *is not a root of unity, then for any* $\mathcal{F}$ *and any* $a, b \in \mathbb{C}$*, if* g *has signature matrix* $P \left[\begin{smallmatrix} a & 0 \\ 0 & b \end{smallmatrix}\right] P^{-1}$*, then*

$$\text{Holant}(\mathcal{F}, =_2 \mid f, g) \leq_T^p \text{Holant}(\mathcal{F}, =_2 \mid f).$$

Proof. Let l be any positive integer. In Pl-Holant$(\mathcal{F}, =_2 \mid f)$, by connecting l copies of f on the RHS via $=_2$ on the LHS, we can implement f_l with the signature matrix $N^l = P \begin{bmatrix} \lambda^l & 0 \\ 0 & \mu^l \end{bmatrix} P^{-1}$. Since $\frac{\lambda}{\mu}$ is not a root of unity, for any positive integer l, $(\frac{\lambda}{\mu})^l \neq 1$.

Consider an instance Ω of Pl-Holant$(\mathcal{F}, =_2 \mid f, g)$. Suppose that g appears t times. We obtain Ω_l from Ω by replacing each occurrence of g with f_l. Since f_l has the signature matrix N^l, we can view our construction of Ω_l as replacing f_l by 3 signatures, with matrix P, $\begin{bmatrix} \lambda^l & 0 \\ 0 & \mu^l \end{bmatrix}$, and P^{-1}, respectively. We stratify the assignments in Ω_l with nonzero evaluations based on the assignments to the t occurrences of the signature with the signature matrix $\begin{bmatrix} \lambda^l & 0 \\ 0 & \mu^l \end{bmatrix}$. Suppose there are i times it was assigned 00 with function value λ^l, and j times 11 with function value μ^l. To have a nonzero evaluation clearly $i + j = t$. Let c_{ij} be the sum over all such assignments of the products of evaluations of all signatures (including the signatures corresponding to matrices P and P^{-1}) in Ω_l except for $\begin{bmatrix} \lambda^l & 0 \\ 0 & \mu^l \end{bmatrix}$. Then

$$\begin{aligned} \text{Holant}_{\Omega_l} &= \sum_{i+j=t} \left(\lambda^l\right)^i \left(\mu^l\right)^j c_{ij} \\ &= \mu^{lt} \sum_{0 \leq i \leq t} \left(\left(\frac{\lambda}{\mu}\right)^l\right)^i c_{i,t-i}. \end{aligned}$$

By oracle calls to Pl-Holant$(\mathcal{F}, =_2 \mid f)$, we can get Holant_{Ω_l} for any $1 \leq l \leq t+1$. Since $(\frac{\lambda}{\mu})^l \neq 1$ for $l \geq 1$, we have $(\frac{\lambda}{\mu})^u \neq (\frac{\lambda}{\mu})^v$, for any two distinct integers $u, v \geq 0$. Therefore we

get a non-singular Vandermonde system. We can solve all c_{ij} for $i+j=t$ given Holant_{Ω_l} for all $1 \le l \le t+1$. Then we can compute $\sum_{i+j=t} c_{ij}a^i b^j$, the desired Holant value. Hence,

$$\text{Pl-Holant}(\mathcal{F}, =_2 \mid f, g) \le_T^p \text{Pl-Holant}(\mathcal{F}, =_2 \mid f).$$ ◀

▶ **Lemma 14.** *Let f be a non-degenerate binary signature, then for any $\mathcal{F}$,*

$$\text{Pl-Holant}(\mathcal{F}, =_2 \mid f, =_2) \le_T^p \text{Pl-Holant}(\mathcal{F}, =_2 \mid f).$$

Proof. Since f is non-degenerate, by the Jordan normal form, there exists a non-singular matrix P such that the signature matrix of f takes the form $\left[\begin{smallmatrix} f_{00} & f_{01} \\ f_{10} & f_{11} \end{smallmatrix}\right] = P \left[\begin{smallmatrix} \lambda & 0 \\ 0 & \mu \end{smallmatrix}\right] P^{-1}$ with $\lambda\mu \neq 0$ or, up to a nonzero constant multiple, $\left[\begin{smallmatrix} f_{00} & f_{01} \\ f_{10} & f_{11} \end{smallmatrix}\right] = P \left[\begin{smallmatrix} 1 & \lambda \\ 0 & 1 \end{smallmatrix}\right] P^{-1}$ with $\lambda \neq 0$.

In the first case, if there is a positive integer j such that $\lambda^j = \mu^j$, then we may directly implement $=_2$ on the RHS by connecting j copies of f via $=_2$ on the LHS. Otherwise, $\frac{\lambda}{\mu}$ is not a root of unity and we get $=_2$ on the RHS by Lemma 13.

In the second case $P \left[\begin{smallmatrix} 1 & \lambda \\ 0 & 1 \end{smallmatrix}\right] P^{-1}$, by connecting l copies of f on the RHS via $=_2$ on the LHS, where l is a positive integer, we can implement f_l with the signature matrix $P \left[\begin{smallmatrix} 1 & l\lambda \\ 0 & 1 \end{smallmatrix}\right] P^{-1}$.

The following proof is similar to Lemma 13. Consider an instance Ω of Pl-Holant$(\mathcal{F}, =_2 \mid f, =_2)$. Suppose the signature $=_2$ on the RHS appears t times. We obtain a planar signature grid Ω_l, a problem in Pl-Holant$(\mathcal{F}, =_2 \mid f)$, by replacing each occurrence of $=_2$ on the RHS with f_l. We can view our construction of Ω_l as replacing f_l by 3 signatures, with matrix P, $\left[\begin{smallmatrix} 1 & l\lambda \\ 0 & 1 \end{smallmatrix}\right]$, and P^{-1}, respectively. We stratify the assignments in Ω_l with nonzero evaluations based on the assignments to the t occurrences of the signature with the signature matrix $\left[\begin{smallmatrix} 1 & l\lambda \\ 0 & 1 \end{smallmatrix}\right]$. Suppose there are i times it was assigned $00, 11$ with function value 1, and j times 01 with function value $l\lambda$. Then $i+j=t$. Let c_{ij} be the sum over all such assignments of the products of evaluations of all signatures (including the signatures corresponding to matrices P and P^{-1}) in Ω_l except for $\left[\begin{smallmatrix} 1 & l\lambda \\ 0 & 1 \end{smallmatrix}\right]$. Then

$$\text{Holant}_{\Omega_l} = \sum_{i+j=t} (l\lambda)^j c_{ij}.$$

By oracle calls to Pl-Holant$(\mathcal{F}, =_2 \mid f)$, we can get Holant_{Ω_l} for any $1 \le l \le t+1$. For any two distinct integers $l, l' \ge 0$, $l\lambda \neq l'\lambda$ since $\lambda \neq 0$. Therefore we get a non-singular Vandermonde system. We can solve for all c_{ij} $(i+j=t)$ given Holant_{Ω_l} for all $1 \le l \le t+1$. Then notice that c_{t0} is the desired Holant value. Therefore,

$$\text{Pl-Holant}(\mathcal{F}, =_2 \mid f, =_2) \le_T^p \text{Pl-Holant}(\mathcal{F}, =_2 \mid f).$$ ◀

Using a similar proof idea as in Lemma 13 we can prove

▶ **Corollary 15.** *Let $\mathcal{F}$ and $\mathcal{G}$ be any two signature sets, then we have*

$$\text{Pl-Holant}(=_4, \mathcal{G} \mid \mathcal{F}, =_2) \le_T^p \text{Pl-Holant}([1,0,0,0,x], \mathcal{G} \mid \mathcal{F}, =_2),$$

for any $x \neq 0$.

Lin and Wang proved the following lemma (Lemma 3.1 in [22]).

▶ **Lemma 16** (Lin-Wang). *Let $\mathcal{F}$ be a set of signatures, and f be a signature. Then*

$$\text{Holant}(\mathcal{F}, f) \le_T^p \text{Holant}(\mathcal{F}, f^{\otimes k}),$$

for any $k \ge 1$.

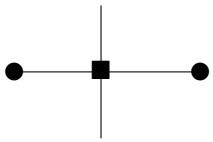


Figure 1 The operation replacing the edge between F and F', drawn vertically. The circle vertices are labeled by $[1,1]$ and the square is labeled by $=_4$. Effectively the new node has signature $=_2$, thus keeping the Holant value unchanged.

The proof of Lemma 16 is non-planar. Thus it cannot be applied directly to planar Holant problems. We give the following lemma for planar graphs.

▶ **Lemma 17.** *Let $\mathcal{F}$ be a set consisting of signatures of even arities and let f be a non-degenerate binary signature, then*

$$\text{Pl-Holant}(=_4 \mid \mathcal{F}, f, [1,1]) \leq_T^p \text{Pl-Holant}(=_4 \mid \mathcal{F}, f, [1,1]^{\otimes 2}).$$

Proof. In the setting of Pl-Holant$(=_4 \mid \mathcal{F}, f, [1,1]^{\otimes 2})$, by adding a loop to $=_4$ using $[1,1]^{\otimes 2}$, we have $=_2$ on the LHS. Then by Lemma 14, we have $=_2$ on the RHS. Now that we have $=_2$ on both sides, we can ignore the bipartite restriction. Thus we just need to prove that

$$\text{Pl-Holant}(=_4 \mid \mathcal{F}, f, [1,1]) \leq_T^p \text{Pl-Holant}(=_4, =_2, \mathcal{F}, f, [1,1]^{\otimes 2}).$$

Given any instance Ω of Pl-Holant$(=_4 \mid \mathcal{F}, f, [1,1])$, we may assume the plane graph of Ω is connected, since the Holant value on Ω is the product over its connected components. Moreover, since all signatures in $\mathcal{F}$ have even arities, the number of occurrences of $[1,1]$ must be even.

Let T be a spanning tree of the dual graph of Ω, and pick any node as the root of T. For definiteness we pick the node of T that corresponds to the external face of Ω as root. Let $\mathfrak{F}$ be a leaf node of T, corresponding to a face F of Ω. Suppose there are an even number of $[1,1]$ inside F, then we can connect them in pairs within the face by copies of $[1,1]^{\otimes 2}$, maintaining planarity. Suppose there are an odd number of $[1,1]$ in $\mathfrak{F}$ and suppose $\mathfrak{F}$ is not the root of T. Let the parent node of $\mathfrak{F}$ correspond to the face F' of Ω, and F and F' share the edge e in Ω. Then we replace e by a path of length 2, put $=_4$ on the new node, and connect two input variables of $=_4$ each to a copy of $[1,1]$, one inside F and one inside F'. This operation effectively changes the new $=_4$ to $=_2$, thus not changing the Holant value, while at the same time changing the parity of the numbers of $[1,1]$'s inside F *and* F'. This is illustrated in Figure 1.

Then we can replace those $[1,1]$'s inside F in pairs by $[1,1]^{\otimes 2}$. We delete the leaf node from T, and complete the proof by induction. Note that finally at the root of T, there must be an even number of $[1,1]$, because the parity of the total number of $[1,1]$ is unchanged during this process. Thus we can simulate Pl-Holant$(=_4 \mid \mathcal{F}, f, [1,1])$ by Pl-Holant$(=_4 \mid \mathcal{F}, f, [1,1]^{\otimes 2})$. ◀

4 Trichotomy for Spin Systems on 4-regular Graphs

In this section, we prove Theorem 24 for the special case $k = 4$.

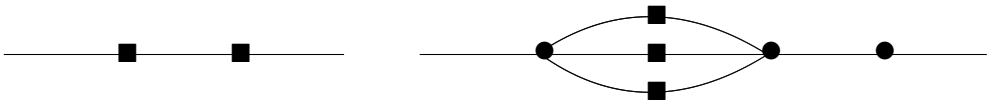


Figure 2 The gadgets realizing g_1 and g_2. The circle vertices are labeled by $=_4$ and squares are labeled by f. For the squares, the edge on the left side corresponds to the variable x_1 of f and the edge on the right side corresponds to x_2.

We say a non-singular M has infinite projective order if M^n is not a scalar multiple of I for any $n \geq 1$. Let $y \in \mathbb{R}$. The matrix $M = \left[\begin{smallmatrix} 1 & y \\ -y & 1 \end{smallmatrix}\right]$ is diagonalizable, $M = Z \left[\begin{smallmatrix} 1+y\mathrm{i} & 0 \\ 0 & 1-y\mathrm{i} \end{smallmatrix}\right] Z^{-1}$, where $Z = \frac{1}{\sqrt{2}} \left[\begin{smallmatrix} 1 & 1 \\ \mathrm{i} & -\mathrm{i} \end{smallmatrix}\right]$. The ratio of the two eigenvalues is $\frac{1+y\mathrm{i}}{1-y\mathrm{i}}$. Therefore this M has infinite projective order iff $\frac{1+y\mathrm{i}}{1-y\mathrm{i}}$ is not a root of unity.

The following lemma is a reduction that follows from Theorem 1.

▶ **Lemma 18.** *Let $\mathcal{F}$ be any signature set containing a binary signature $(1, x, -x, 1)$, where $x \in \mathbb{R}$ and $x \neq 0, \pm 1$. Then for some $y \in \mathbb{R}$,*

$$\text{Holant}(=_4 \mid \mathcal{F}, (1, y, -y, 1)) \leq_T^p \text{Holant}(=_4 \mid \mathcal{F}),$$

where the signature matrix $\left[\begin{smallmatrix} 1 & y \\ -y & 1 \end{smallmatrix}\right]$ has eigenvalues $1 \pm y\mathrm{i}$, with ratio $\frac{1+y\mathrm{i}}{1-y\mathrm{i}}$ not a root of unity, and thus the matrix has infinite projective order.

Proof. In Holant$(=_4 \mid \mathcal{F})$, by adding a loop to $=_4$ using $(1, x, -x, 1) \in \mathcal{F}$, we have $=_2$ on the LHS. Since $(1, x, -x, 1)$ is non-degenerate, by Lemma 14 we obtain $=_2$ on the RHS. Once we have $=_2$ on both sides we can freely move signatures from either side, and so we can ignore the bipartite restriction.

By the construction in Figure 2, using $f = \left[\begin{smallmatrix} 1 & x \\ -x & 1 \end{smallmatrix}\right]$, we can realize binary functions g_1 with $M(g_1) = \left[\begin{smallmatrix} 1-x^2 & 2x \\ -2x & 1-x^2 \end{smallmatrix}\right]$, and g_2 with $M(g_2) = \left[\begin{smallmatrix} 1-x^4 & x+x^3 \\ -x-x^3 & 1-x^4 \end{smallmatrix}\right] = (x^2+1) \left[\begin{smallmatrix} 1-x^2 & x \\ -x & 1-x^2 \end{smallmatrix}\right]$.

This means that, if we assign $(b_1, b_2) \in \{0,1\}^2$ to the two external edges, and form the sum of product over 0-1 assignments on internal edges we get the value in the matrix in row b_1 and column b_2.

The matrix $\left[\begin{smallmatrix} 1-x^2 & x \\ -x & 1-x^2 \end{smallmatrix}\right]$ has two nonzero eigenvalues $1 - x^2 \pm x\mathrm{i}$, with ratio $\frac{a+b\mathrm{i}}{a-b\mathrm{i}}$, where $a = 1 - x^2$ and $b = x$. This ratio is a root of unity iff the complex argument φ of $a + b\mathrm{i} = |a + b\mathrm{i}|e^{\mathrm{i}\varphi}$ is a rational multiple of π, where $\cot(\varphi) = \frac{a}{b}$.

Similarly, $\left[\begin{smallmatrix} 1-x^2 & 2x \\ -2x & 1-x^2 \end{smallmatrix}\right]$ has two nonzero eigenvalues $1 - x^2 \pm 2x\mathrm{i}$, with ratio $\frac{a+2b\mathrm{i}}{a-2b\mathrm{i}}$. This ratio is a root of unity iff the complex argument ψ of $a + 2b\mathrm{i} = |a + 2b\mathrm{i}|e^{\mathrm{i}\psi}$ is a rational multiple of π, where $\cot(\psi) = \frac{a}{2b}$.

By Theorem 1 these cannot both happen. Therefore at least one of the two constructions defines a matrix that has infinite projective order. ◀

Let $f = (w, x, y, z)$ be a binary signature where $w, x, y, z \in \mathbb{R}$. If $wz = xy$ or there are two or more zeros in $\{w, x, y, z\}$, then $f \in \mathcal{P}$ and Holant$(=_4 \mid f)$ can be computed in polynomial

time. Moreover, if $x = y$, then f is symmetric and Theorem 24 follows Theorem 9. Thus we now assume the following:

Assumption: The binary signature $f = (w, x, y, z)$ satisfies $wz \neq xy, x \neq y$ and there is at most one zero in $\{w, x, y, z\}$.

First, we consider the case that there is exactly one zero in $\{w, z\}$.

▶ **Lemma 19.** *Let $f = (w, x, y, z)$, where $w, x, y, z \in \mathbb{R}$. If there is exactly one zero in $\{w, z\}$ and $xy \neq 0$, then* Pl-Holant$(=_4 | f)$ *is* #P*-hard.*

Proof. By flipping 0 and 1, we may assume that $w \neq 0, z = 0$. By normalizing $w = 1$ we can assume that $f = (1, x, y, 0)$.

In the setting of Pl-Holant$(=_4 | f)$, by adding a loop using f to $=_4$, we have $[1, 0]^{\otimes 2}$ on the LHS. Then taking two copies of f and connecting $[1, 0]$ to the variable x_1 of each copy we get $[1, x]^{\otimes 2}$ on the RHS. This operation used $[1, 0]^{\otimes 2}$ on the LHS. By adding a loop using $[1, x]^{\otimes 2}$ to $=_4$, we have $[1, 0, x^2]$ on the LHS. So we have

$$\text{Pl-Holant}(=_4, [1, 0]^{\otimes 2}, [1, 0, x^2] | f) \leq_T^p \text{Pl-Holant}(=_4 | f). \tag{9}$$

By a holographic transformation using $\left[\begin{smallmatrix} 1 & 0 \\ 0 & x^{-1} \end{smallmatrix}\right]$, we have

$$\text{Pl-Holant}([1, 0, 0, 0, x^{-4}], [1, 0]^{\otimes 2}, =_2 | (1, x^2, xy, 0)) \equiv_T^p \text{Pl-Holant}(=_4, [1, 0]^{\otimes 2}, [1, 0, x^2] | f). \tag{10}$$

Since $(1, x^2, xy, 0)$ is non-degenerate, we can get $(=_2)$ on the RHS by Lemma 14,

$$\begin{aligned} & \text{Pl-Holant}([1, 0, 0, 0, x^{-4}], [1, 0]^{\otimes 2}, =_2 | (1, x^2, xy, 0), =_2) && (11) \\ \leq_T^p \quad & \text{Pl-Holant}([1, 0, 0, 0, x^{-4}], [1, 0]^{\otimes 2}, =_2 | (1, x^2, xy, 0)) && (12) \end{aligned}$$

Now that we have $=_2$ on both sides of (11), we will ignore the bipartiteness restriction. We construct $[1, 1]^{\otimes 2}$ in (11) as follows.

- If $|x| = 1$, as $x \in \mathbb{R}$, we have $(1, x^2, xy, 0) = (1, 1, xy, 0)$. Then by taking two copies of $(1, 1, xy, 0)$ and connecting $[1, 0]$ to the variable x_1 for each copy (using $[1, 0]^{\otimes 2}$), we get $[1, 1]^{\otimes 2}$.
- If $|x| \neq 1$, by adding a loop using $=_2$ to $[1, 0, 0, 0, x^{-4}]$ we have $[1, 0, x^{-4}]$. Since $|x^{-4}| \neq 1$, we can get $[1, 0, x^{-2}]$ by Lemma 13. Connecting one variable of $[1, 0, x^{-2}]$ to the variable x_2 of $(1, x^2, xy, 0)$, we get the signature $(1, 1, xy, 0)$ and proceed as above.

We can place the constructed $[1, 1]^{\otimes 2}$ on the RHS of (11). Moreover, note that we have $[1, 0, 0, 0, x^{-4}]$ on the LHS and $=_2$ on the RHS in (11). Thus we have $=_4$ on the LHS by Corollary 15. This implies that

$$\begin{aligned} & \text{Pl-Holant}(=_4 | [1, 1]^{\otimes 2}, (1, x^2, xy, 0)) \\ & \leq_T^p \text{Pl-Holant}([1, 0, 0, 0, x^{-4}], [1, 0]^{\otimes 2}, =_2 | (1, x^2, xy, 0), =_2). \end{aligned} \tag{13}$$

Then by Lemma 17 we have

$$\text{Pl-Holant}(=_4 | [1, 1], (1, x^2, xy, 0)) \leq_T^p \text{Pl-Holant}(=_4 | [1, 1]^{\otimes 2}, (1, x^2, xy, 0)). \tag{14}$$

In Pl-Holant($=_4$ $|[1,1],(1,x^2,xy,0)$), by connecting $[1,1]$ to $=_4$ we have $=_3$ on the LHS. This implies that

$$\text{Pl-Holant}(=_3 |(1,x^2,xy,0)) \leq_T^p \text{Pl-Holant}(=_4 |[1,1],(1,x^2,xy,0)). \tag{15}$$

By Theorem 11, Pl-Holant($=_3$ $|(1,x^2,xy,0)$) is #P-hard. Then by (9), (10), (11), (13), (14) and (15), Pl-Holant($=_4$ $|f$) is #P-hard. ◀

Now we can assume that $wz \neq 0$ and $f = (1,x,y,z)$ by normalizing $w = 1$.

▶ **Lemma 20.** *Let $f = (1,x,y,z)$, where $x,y,z \in \mathbb{R}$ and $z \neq 0$. If $|z| \neq 1$, then* Pl-Holant($=_4$ $|f$) *is* #P*-hard.*

Proof. In Pl-Holant($=_4$ $|f$), by adding a loop using f to $=_4$, we have $[1,0,z]$ on the LHS, i.e. we have

$$\text{Pl-Holant}(=_4,[1,0,z]|f) \leq_T^p \text{Pl-Holant}(=_4 |f). \tag{16}$$

For Pl-Holant($=_4,[1,0,z]|f$), by the holographic transformation using $\begin{bmatrix} 1 & 0 \\ 0 & z^{-\frac{1}{2}} \end{bmatrix}$, we have

$$\text{Pl-Holant}([1,0,0,0,z^{-2}],=_2 |(1,xz^{\frac{1}{2}},yz^{\frac{1}{2}},z^2)) \equiv_T^p \text{Pl-Holant}(=_4,[1,0,z]|f). \tag{17}$$

Note that $z^{\frac{1}{2}}$ can be a complex number.

Now we consider the LHS problem in (17). Firstly, since $(1,xz^{\frac{1}{2}},yz^{\frac{1}{2}},z^2)$ is non-degenerate, by Lemma 14, we have $=_2$ on the RHS. Then by Corollary 15, we have $=_4$ on the LHS. Moreover, by adding a loop using $=_2$ to $[1,0,0,0,z^{-2}]$, we have $[1,0,z^{-2}]$ on the LHS. This implies that

$$\begin{aligned} & \text{Pl-Holant}([1,0,z^{-2}],=_2,=_4 \mid =_2,(1,xz^{\frac{1}{2}},yz^{\frac{1}{2}},z^2)) && (18)\\ \leq_T^p \quad & \text{Pl-Holant}([1,0,0,0,z^{-2}],=_2 |(1,xz^{\frac{1}{2}},yz^{\frac{1}{2}},z^2)). && (19)\end{aligned}$$

In problem (18), we have $=_2$ on both sides, and so we can ignore the bipartiteness restriction. Since $|z| \neq 1$, by Lemma 13, we have $[1,0]^{\otimes 2}$ and $[1,0,x^{-1}z^{-\frac{1}{2}}]$. Connecting $[1,0,x^{-1}z^{-\frac{1}{2}}]$ to the variable x_2 of $(1,xz^{\frac{1}{2}},yz^{\frac{1}{2}},z^2)$, we have $(1,1,yz^{\frac{1}{2}},x^{-1}z^{\frac{3}{2}})$. By taking two copies of $(1,1,yz^{\frac{1}{2}},x^{-1}z^{\frac{3}{2}})$ and connecting $[1,0]$ to the variable x_1 for each copy, we have $[1,1]^{\otimes 2}$. This makes use of $[1,0]^{\otimes 2}$. This implies that

$$\begin{aligned} & \text{Pl-Holant}(=_4 |[1,1]^{\otimes 2},(1,xz^{\frac{1}{2}},yz^{\frac{1}{2}},z^2)) \\ & \leq_T^p \text{Pl-Holant}([1,0,z^{-2}],=_2,=_4 \mid =_2,(1,xz^{\frac{1}{2}},yz^{\frac{1}{2}},z^2)). && (20)\end{aligned}$$

Then by Lemma 17, we have

$$\text{Pl-Holant}(=_4 |[1,1],(1,xz^{\frac{1}{2}},yz^{\frac{1}{2}},z^2)) \leq_T^p \text{Pl-Holant}(=_4 |[1,1]^{\otimes 2},(1,xz^{\frac{1}{2}},yz^{\frac{1}{2}},z^2)). \tag{21}$$

By connecting $[1,1]$ to $=_4$, we have $=_3$ on the LHS. This implies that

$$\text{Pl-Holant}(=_3 |(1,xz^{\frac{1}{2}},yz^{\frac{1}{2}},z^2)) \leq_T^p \text{Pl-Holant}(=_4 |[1,1],(1,xz^{\frac{1}{2}},yz^{\frac{1}{2}},z^2)). \tag{22}$$

By Corollary 12, Pl-Holant($=_3$ $|(1,xz^{\frac{1}{2}},yz^{\frac{1}{2}},z^2)$) is #P-hard since $|z^2| \neq 1$ and there is at most one zero in $\{xz^{\frac{1}{2}},yz^{\frac{1}{2}}\}$. Then by (16), (17), (18), (20), (21) and (22), Pl-Holant($=_4$ $|f$) is #P-hard. ◀

In addition to $w = 1$, $wz \neq xy$, $x \neq y$, we may now assume $|z| = 1$. Being real, $z = \pm 1$. We next consider the case that $x \neq \pm y$.

▶ **Lemma 21.** *Let $f = (1, x, y, z)$, where $x, y, z \in \mathbb{R}$ and $|z| = 1, x \neq \pm y$, then* Pl-Holant$(=_4 |f)$ *is* #P*-hard.*

Proof. For Pl-Holant$(=_4 |f)$, by adding a loop using f to $=_4$ we have $[1, 0, z]$ on the LHS. Take two copies of f and one copy of $[1, 0, z]$, and connect them in a symmetric path, i.e., connect x_2 of both copies of f to $[1, 0, z]$, leaving x_1 of both copies of f as external edges, we get the following symmetric signature g on the RHS with the signature matrix (using $z^2 = 1$),

$$\left[\begin{smallmatrix} 1 & x \\ y & z \end{smallmatrix}\right] \left[\begin{smallmatrix} 1 & 0 \\ 0 & z \end{smallmatrix}\right] \left[\begin{smallmatrix} 1 & y \\ x & z \end{smallmatrix}\right] = \begin{bmatrix} 1+x^2z & x+y \\ x+y & z+y^2 \end{bmatrix}.$$

For $z = 1$, $g = [1 + x^2, x + y, 1 + y^2]$. We have $x + y \neq 0$ and $1 + x^2 \neq 1 + y^2$ by $x \neq \pm y$. Thus Pl-Holant$(=_4 |g)$ is #P-hard by Corollary 10. So Pl-Holant$(=_4 |f)$ is #P-hard.

For $z = -1$, $g = [1 - x^2, x + y, y^2 - 1]$. Still $x + y \neq 0$.

- If $|1-x^2| \neq |y^2-1|$, then Pl-Holant$(=_4 |g)$ is #P-hard by Corollary 10. So Pl-Holant$(=_4 |f)$ is #P-hard.
- If $1 - x^2 = 1 - y^2$, then $x^2 = y^2$. This is a contradiction.
- If $1 - x^2 = y^2 - 1$, we have $x^2 \neq 1$. Otherwise $y^2 = 1$ and again a contradiction. Thus we have $1 - x^2 = y^2 - 1 \neq 0$. So we can assume that $g = [1, \frac{x+y}{1-x^2}, 1]$ after the nonzero scalar $1 - x^2$. By adding a loop using g to $=_4$, we have $=_2$ on the LHS. By connecting two copies of f using $=_2$ on the LHS, we get the signature h with the signature matrix

  $$\left[\begin{smallmatrix} 1 & x \\ y & -1 \end{smallmatrix}\right] \left[\begin{smallmatrix} 1 & 0 \\ 0 & 1 \end{smallmatrix}\right] \left[\begin{smallmatrix} 1 & y \\ x & -1 \end{smallmatrix}\right] = \begin{bmatrix} 1+x^2 & y-x \\ y-x & 1+y^2 \end{bmatrix}.$$

  Note that h is symmetric and $y-x \neq 0$, $1+x^2 \neq 1+y^2$ by $x \neq \pm y$. Thus Pl-Holant$(=_4 |h)$ is #P-hard by Corollary 10. So Pl-Holant$(=_4 |f)$ is #P-hard. ◀

Now we can assume that $w = 1$, $z = \pm 1$, and $x = -y \neq 0$, i.e., $f = (1, x, -x, \pm 1)$. By Lemma 7 and the statements before Lemma 7, the pairs $(=_4 |(1, x, -x, -1))$ and $(=_4 |(1, x, -x, 1))$ are $\mathcal{M}$-transformable under the holographic transformation H and Z respectively. Thus Pl-Holant$(=_4 |(1, x, -x, \pm 1))$ can be computed in polynomial time. Moreover, if $x = \pm 1$ then $f \in \mathcal{A}$ by Lemma 5 and Holant$(=_4 |f)$ can be computed in polynomial time on general graphs. In the following, we consider Holant$(=_4 |f)$ on general graphs with $x \neq \pm 1$. These are cases where Holant$(=_4 |f)$ is #P-hard, but Pl-Holant$(=_4 |f)$ is in P. It is for the proof of these cases that we ultimately use Theorem 1 from number theory.

▶ **Lemma 22.** *Let $f = (1, x, -x, z)$, where $x \in \mathbb{R}$, $z = \pm 1$ and $x \neq 0, \pm 1$. Then* Holant$(=_4 |f)$ *is* #P*-hard.*

Proof. For $z = 1$, by Lemma 18, there exists $y \in \mathbb{R}$ such that

$$\text{Holant}(=_4 |f, (1, y, -y, 1)) \leq_T^p \text{Holant}(=_4 |f), \tag{23}$$

where $\left[\begin{smallmatrix} 1 & y \\ -y & 1 \end{smallmatrix}\right]$ has infinite projective order, i.e., the ratio of eigenvalues $\frac{1+y\mathrm{i}}{1-y\mathrm{i}}$ is not a root of unity. Recall that

$$\left[\begin{smallmatrix} 1 & y \\ -y & 1 \end{smallmatrix}\right] = Z \begin{bmatrix} 1+y\mathrm{i} & 0 \\ 0 & 1-y\mathrm{i} \end{bmatrix} Z^{-1},$$

where $Z = \frac{1}{\sqrt{2}} \left[\begin{smallmatrix} 1 & 1 \\ \mathrm{i} & -\mathrm{i} \end{smallmatrix}\right]$. Then by Lemma 13, we can choose g having the signature matrix

$$Z \left[\begin{smallmatrix} 2 & 0 \\ 0 & 0 \end{smallmatrix}\right] Z^{-1} = \left[\begin{smallmatrix} 1 & -\mathrm{i} \\ \mathrm{i} & 1 \end{smallmatrix}\right],$$

such that

$$\text{Holant}(=_4 \mid f, g) \leq_T^p \text{Holant}(=_4 \mid f, (1, y, -y, 1)). \tag{24}$$

Note that g is degenerate, being the tensor product of two unary signatures, $g = [1, \mathrm{i}] \otimes [1, -\mathrm{i}]$. If we take 4 copies of g and connect each of their first variable corresponding to $[1, -\mathrm{i}]$ to $=_4$, we obtain $2[1, \mathrm{i}]^{\otimes 4}$ on the RHS. The proof of Lemma 16 can be easily adapted to the bipartite case with $=_4$ on the LHS, and we get

$$\text{Holant}(=_4 \mid f, [1, \mathrm{i}]) \leq_T^p \text{Holant}(=_4 \mid f, [1, \mathrm{i}]^{\otimes 4}). \tag{25}$$

In $\text{Holant}(=_4 \mid f, [1, \mathrm{i}])$, by connecting $[1, \mathrm{i}]$ to $=_4$, we have $[1, 0, 0, \mathrm{i}]$ on the LHS, i.e.,

$$\text{Holant}([1, 0, 0, \mathrm{i}] \mid f) \leq_T^p \text{Holant}(=_4 \mid f, [1, \mathrm{i}]). \tag{26}$$

Then by the holographic transformation using $\begin{bmatrix} 1 & 0 \\ 0 & \mathrm{i}^{-\frac{1}{3}} \end{bmatrix}$, we have

$$\text{Holant}(=_3 \mid (1, x\mathrm{i}^{\frac{1}{3}}, -x\mathrm{i}^{\frac{1}{3}}, z\mathrm{i}^{\frac{2}{3}})) \equiv \text{Holant}([1, 0, 0, \mathrm{i}] \mid f). \tag{27}$$

Since $|x\mathrm{i}^{\frac{1}{3}}| \neq 0, 1$, $\text{Holant}(=_3 \mid (1, x\mathrm{i}^{\frac{1}{3}}, -x\mathrm{i}^{\frac{1}{3}}, z\mathrm{i}^{\frac{2}{3}}))$ is #P-hard by Theorem 11. Then by (23), (24), (25), (26) and (27), $\text{Holant}(=_4 \mid f)$ is #P-hard.

For $z = -1$, by adding a loop to $=_4$ using f, we have $[1, 0, -1]$ on the LHS. This implies that

$$\text{Holant}(=_4, [1, 0, -1] \mid f) \leq_T^p \text{Holant}(=_4 \mid f). \tag{28}$$

Then by the holographic transformation using $\begin{bmatrix} 1 & 0 \\ 0 & \mathrm{i} \end{bmatrix}$, we have

$$\text{Holant}(=_4, =_2 \mid (1, x\mathrm{i}, -x\mathrm{i}, 1)) \equiv \text{Holant}(=_4, [1, 0, -1] \mid f). \tag{29}$$

Note that $(1, x\mathrm{i}, -x\mathrm{i}, 1)$ has the signature matrix

$$\begin{bmatrix} 1 & x\mathrm{i} \\ -x\mathrm{i} & 1 \end{bmatrix} = Z' \begin{bmatrix} 1+x & 0 \\ 0 & 1-x \end{bmatrix} Z'^{-1},$$

where $Z' = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ -\mathrm{i} & \mathrm{i} \end{bmatrix}$. Note that the ratio $\frac{1+x}{1-x}$ is not a root of unity since $x \in \mathbb{R}$ and $x \neq 0, \pm 1$. By Lemma 13, we have the signature with the signature matrix

$$Z' \begin{bmatrix} 2 & 0 \\ 0 & 0 \end{bmatrix} Z'^{-1} = \begin{bmatrix} 1 & \mathrm{i} \\ -\mathrm{i} & 1 \end{bmatrix}$$

on the RHS, i.e., we have $[1, -\mathrm{i}] \otimes [1, \mathrm{i}]$ on the RHS. The remaining proof is the same as the previous case that $z = 1$ and we omit it here. ◀

Now we give the main theorem of this section.

▶ **Theorem 23.** *Let $f = (w, x, y, z)$ with $w, x, y, z \in \mathbb{R}$. Then $\text{Holant}(=_4 \mid f)$ is #P-hard except in the following cases, where the problem is computable in polynomial time.*

- *$f \in \mathcal{P}$: $wz = xy$, or $w = z = 0$, or $x = y = 0$;*
- *$f \in \mathcal{A}$: $w^2 = x^2 = y^2 = z^2$.*

If the input is restricted to planar graphs, then another case becomes polynomial time computable but everything else remains #P-hard.

- *The pair $(=_4 \mid f)$ is $\mathcal{M}$-transformable: $w^2 = z^2$ and $x^2 = y^2$.*

Proof. If $wz = xy$, or $w = z = 0$, or $x = y = 0$, then $f \in \mathcal{P}$ and $\text{Holant}(=_4 \mid f)$ can be computed in polynomial time. In the following, we assume that $wz \neq xy$ and there is at most one zero in $\{w, x, y, z\}$.

- If $wz = 0$, then Pl-Holant$(=_4 \mid f)$ is #P-hard by Lemma 19;
- if $wz \neq 0, |w| \neq |z|$, then Pl-Holant$(=_4 \mid f)$ is #P-hard by Lemma 20.

Now we can assume that $|w| = |z| \neq 0$, i.e., $f = (w, x, y, \pm w)$, with $w \neq 0$.

- If $x \neq \pm y$, then Pl-Holant$(=_4 \mid f)$ is #P-hard by Lemma 21;
- if $x = y$, then f is symmetric and the theorem has been proved as Theorem 9.

Now we can assume that $|w| = |z| \neq 0$ and $x = -y$, i.e., $f = (w, x, -x, \pm w)$, with $w \neq 0$. Since there is at most one zero among $\{w, x, y, z\}$, with the given form there is actually no zero entry. Since $(=_4 \mid (w, x, -x, -w))$ and $(=_4 \mid (w, x, -x, w))$ are $\mathcal{M}$-transformable under the holographic transformation H and Z respectively by Lemma 7, Pl-Holant$(=_4 \mid (w, x, -x, \pm w))$ can be computed in polynomial time. For general graphs,

- if $w^2 = x^2$, then $f \in \mathcal{A}$ by Lemma 5 and Holant$(=_4 \mid (w, x, -x, \pm w))$ can be computed in polynomial time;
- if $w^2 \neq x^2$, and since there are no zero entries, then Holant$(=_4 \mid f)$ is #P-hard by Lemma 22. ◀

5 Trichotomy for k-regular Graphs

In this section, we prove our main theorem, Theorem 24, a complexity trichotomy for spin systems with not necessarily symmetric real edge weights over k-regular graphs, for any $k \geq 3$.

▶ **Theorem 24.** *Let $f = (w, x, y, z)$ with $w, x, y, z \in \mathbb{R}$. Then* Holant$(=_k \mid f)$*, where $k \geq 3$, is* #P*-hard except in the following cases, where the problem is computable in polynomial time.*

- *$f \in \mathcal{P}$: $wz = xy$, or $w = z = 0$, or $x = y = 0$;*
- *$f \in \mathcal{A}$: $w^2 = x^2 = y^2 = z^2$.*

If the input is restricted to planar graphs, then another case becomes polynomial time computable but everything else remains #P*-hard.*

- *The pair $(=_k \mid f)$ is $\mathcal{M}$-transformable: $w = \epsilon z, x = \epsilon y$, or k is even and $w = \epsilon z, x = -\epsilon y$, where $\epsilon = \pm 1$.*

Proof. If $wz = xy$ or there are two or more zeros in $\{w, x, y, z\}$, then $f \in \mathcal{P}$ and Holant$(=_k \mid f)$ can be computed in polynomial time. If $x = y$, then f is symmetric and the theorem follows Theorem 9. In the following, we assume that $wz \neq xy$, $x \neq y$ and there is at most one zero in $\{w, x, y, z\}$.

For $k = 3$ or 4, the theorem has been proved in Theorem 11 and Theorem 23 respectively. So we can assume that $k \geq 5$.

Firstly, we consider the case that $wz = 0$. By assumption, we have $xy \neq 0$ and there is exact one zero in $\{w, z\}$. Without loss of generality, we assume that $w \neq 0, z = 0$. Then we may assume that $f = (1, x, y, 0)$ by normalizing $w = 1$.

- If k is odd, by adding $\frac{k-1}{2}$ loops using f to $=_k$, we have $[1, 0]$ on the LHS of Holant$(=_k \mid f)$. By connecting $[1, 0]$ to the variable x_1 of f, we get $[1, x]$ on the RHS. By connecting $k-3$ copies of $[1, x]$ to $=_k$, we have $[1, 0, 0, x^{k-3}]$ of arity 3 on the LHS, i.e., we have

$$\text{Pl-Holant}([1, 0, 0, x^{k-3}] \mid f) \leq_T^p \text{Pl-Holant}(=_k \mid f). \tag{30}$$

Then by the holographic transformation using $\begin{bmatrix} 1 & 0 \\ 0 & x^{-\frac{k-3}{3}} \end{bmatrix}$, we have

$$\text{Pl-Holant}(=_3 \mid (1, x^{\frac{k}{3}}, yx^{\frac{k-3}{3}}, 0)) \equiv_T^p \text{Pl-Holant}([1, 0, 0, x^{k-3}] \mid f). \tag{31}$$

By Theorem 11, Pl-Holant($=_3$ $|(1, x^{\frac{k}{3}}, yx^{\frac{k-3}{3}}, 0)$) is #P-hard. Thus Pl-Holant($=_k$ $|f$) is #P-hard by (30) and (31).

- If k is even, by adding $\frac{k-2}{2}$ loops using f to $=_k$, we have $[1, 0]^{\otimes 2}$ on the LHS. Then we take two copies of f and connect $[1, 0]$ to the variable x_1 for each copy to get $[1, x]^{\otimes 2}$ on the RHS. This can be realized by $[1, 0]^{\otimes 2}$ on the LHS. By adding $\frac{k-4}{2}$ loops using $[1, x]^{\otimes 2}$ to $=_k$, we have $[1, 0, 0, 0, x^{k-4}]$ of arity 4 on the LHS, i.e., we have

$$\text{Pl-Holant}([1, 0, 0, 0, x^{k-4}]|f) \leq_T^p \text{Pl-Holant}(=_k |f). \tag{32}$$

Since k is even and at least 5 by assumption, we have $k \geq 6$ and $k-4 \geq 2$ is even. Hence $x^{k-4} > 0$. Thus we may choose a 4th root $x^{\frac{k-4}{4}} \in \mathbb{R}$. (Any statement in a holographic transformation involving a quantity such as $z^{1/n}$ is valid for any choice as long as a consistent choice is made.)

Then by the holographic transformation using $\begin{bmatrix} 1 & 0 \\ 0 & x^{-\frac{k-4}{4}} \end{bmatrix}$, we have

$$\text{Pl-Holant}(=_4 |(1, x^{\frac{k}{4}}, yx^{\frac{k-4}{4}}, 0)) \equiv_T^p \text{Pl-Holant}([1, 0, 0, 0, x^{k-4}]|f). \tag{33}$$

Note that all the entries of $(1, x^{\frac{k}{4}}, yx^{\frac{k-4}{4}}, 0)$ are real numbers. Therefore we may apply Theorem 23, and conclude that Pl-Holant($=_4$ $|(1, x^{\frac{k}{4}}, yx^{\frac{k-4}{4}}, 0)$) is #P-hard. It follows that Pl-Holant($=_k$ $|f$) is #P-hard by (32) and (33).

Now we consider the case that $wz \neq 0$. So we may assume that $f = (1, x, y, z)$ by normalizing $w = 1$.

Firstly, we consider the case that k is odd. By adding $\frac{k-3}{2}$ loops using f to $=_k$, we have $[1, 0, 0, z^{\frac{k-3}{2}}]$ on the LHS, i.e., we have

$$\text{Pl-Holant}([1, 0, 0, z^{\frac{k-3}{2}}]|f) \leq_T^p \text{Pl-Holant}(=_k |f). \tag{34}$$

Then by the holographic transformation using $\begin{bmatrix} 1 & 0 \\ 0 & z^{-\frac{k-3}{6}} \end{bmatrix}$, we have

$$\text{Pl-Holant}(=_3 |(1, xz^{\frac{k-3}{6}}, yz^{\frac{k-3}{6}}, z^{\frac{k}{3}})) \equiv_T^p \text{Pl-Holant}([1, 0, 0, z^{\frac{k-3}{2}}]|f). \tag{35}$$

- If $z \neq \pm 1$, then since $z \in \mathbb{R}$, we have $|z^{\frac{k}{3}}| \neq 1$ and Pl-Holant($=_3$ $|(1, xz^{\frac{k-3}{6}}, yz^{\frac{k-3}{6}}, z^{\frac{k}{3}})$) is #P-hard by Corollary 12. Thus Pl-Holant($=_k$ $|f$) is #P-hard by (34) and (35).
- If $x \neq \pm y$, then since $x, y \in \mathbb{R}$, we have $|x| \neq |y|$, and thus $|xz^{\frac{k-3}{6}}| \neq |yz^{\frac{k-3}{6}}|$. Then Pl-Holant($=_3$ $|(1, xz^{\frac{k-3}{6}}, yz^{\frac{k-3}{6}}, z^{\frac{k}{3}})$) is #P-hard by Corollary 12. So Pl-Holant($=_k$ $|f$) is #P-hard by (34) and (35).
- The remaining case is that $z = \pm 1$ and $x = -y$ since $x \neq y$, i.e., $f = (1, x, -x, \pm 1)$. (Note that in this case there are no zero entries, since there could have been at most one zero entry; in particular $x \neq 0$.)
  - If $x = \pm 1$, then $f \in \mathcal{A}$ by Lemma 5 and Holant($=_k$ $|f$) can be computed in polynomial time.
  - Suppose $x \neq \pm 1$. For $z = -1$, since $(=_k |(1, x, -x, -1))$ is $\mathcal{M}$-transformable under the holographic transformation H by Lemma 7, Pl-Holant($=_k$ $|(1, x, -x, -1)$) is computable in polynomial time. But for $z = 1$, i.e., $f = (1, x, -x, 1)$, for the problem in the left-hand side of (35), the signature $(1, xz^{\frac{k-3}{6}}, yz^{\frac{k-3}{6}}, z^{\frac{k}{3}})$ is just $(1, x, -x, 1)$. By $|x| \neq 0, 1$, Pl-Holant($=_3$ $|(1, x, -x, 1)$) is #P-hard by Theorem 11. Thus Pl-Holant($=_k$ $|(1, x, -x, 1)$) is #P-hard by (34) and (35).
    Moreover, for general graphs, for either case of $z = +1$ and $= -1$, note that $|xz^{\frac{k-3}{6}}| = |yz^{\frac{k-3}{6}}| \neq 1$. Thus Holant($=_3$ $|(1, xz^{\frac{k-3}{6}}, yz^{\frac{k-3}{6}}, z^{\frac{k}{3}})$) is #P-hard by Theorem 11. So Holant($=_k$ $|f$) is #P-hard by (34) and (35).

Now we consider the case that k is even. To ensure that all the signatures we discuss are real-valued, we need to consider the cases $k \equiv 0 \pmod 4$ and $k \equiv 2 \pmod 4$ separately.

- If $k \equiv 0 \pmod 4$, by adding $\frac{k-4}{2}$ loops using f to $=_k$, we have $[1,0,0,0,z^{\frac{k-4}{2}}]$ on the LHS, i.e., we have

$$\text{Pl-Holant}([1,0,0,0,z^{\frac{k-4}{2}}]|f) \leq_T^p \text{Pl-Holant}(=_k |f). \tag{36}$$

  As $\frac{k-4}{2}$ is even, we have $z^{\frac{k-4}{2}} > 0$. Thus we can choose $z^{-\frac{k-4}{8}} \in \mathbb{R}$. It also follows that $z^{\frac{k}{4}} \in \mathbb{R}$. Then by the holographic transformation using $\begin{bmatrix} 1 & 0 \\ 0 & z^{-\frac{k-4}{8}} \end{bmatrix}$, we have

$$\text{Pl-Holant}(=_4 |(1, xz^{\frac{k-4}{8}}, yz^{\frac{k-4}{8}}, z^{\frac{k}{4}})) \equiv_T^p \text{Pl-Holant}([1,0,0,0,z^{\frac{k-4}{2}}]|f). \tag{37}$$

  - If $z \neq \pm 1$ or $x \neq \pm y$, then Pl-Holant$(=_4 |(1, xz^{\frac{k-4}{8}}, yz^{\frac{k-4}{8}}, z^{\frac{k}{4}}))$ is #P-hard by Theorem 23, and Pl-Holant$(=_k |f)$ is #P-hard by (36) and (37).
  - For $z = \pm 1$ and $x = -y$, i.e., $f = (1, x, -x, \pm 1)$, if $x = \pm 1$, then $f \in \mathcal{A}$ by Lemma 5 and Holant$(=_k |f)$ can be computed in polynomial time.
    Suppose $x \neq \pm 1$. since $(=_k |(1, x, -x, -1))$ and $(=_k |(1, x, -x, 1))$ are $\mathcal{M}$-transformable under the holographic transformations H and Z respectively by Lemma 7, the planar Holant problem Pl-Holant$(=_4 |(1, x, -x, \pm 1))$ is computable in polynomial time. But for general graphs, Holant$(=_4 |(1, xz^{\frac{k-4}{8}}, -xz^{\frac{k-4}{8}}, z^{\frac{k}{4}}))$ is #P-hard by Theorem 23 since $|xz^{\frac{k-4}{8}}| \neq 0, 1$. Thus Holant$(=_k |f)$ is #P-hard by (36) and (37) (These reductions also hold for non-planar Holant problems respectively).

- If $k \equiv 2 \pmod 4$, by adding $\frac{k-2}{2}$ loops using f to $=_k$, we have $[1,0,z^{\frac{k-2}{2}}]$, i.e.,

$$\text{Pl-Holant}(=_k, [1,0,z^{\frac{k-2}{2}}]|f) \leq_T^p \text{Pl-Holant}(=_k |f). \tag{38}$$

  Note that $\frac{k-2}{2}$ is even, and thus $z^{\frac{k-2}{2}} > 0$. Then we can choose $z^{-\frac{k-2}{4}} \in \mathbb{R}$. By the holographic transformation $\begin{bmatrix} 1 & 0 \\ 0 & z^{-\frac{k-2}{4}} \end{bmatrix}$, we have

$$\begin{aligned} &\text{Pl-Holant}([1,0,\cdots,0,z^{-\frac{k(k-2)}{4}}], =_2 |(1, xz^{\frac{k-2}{4}}, yz^{\frac{k-2}{4}}, z^{\frac{k}{2}})) \\ &\leq_T^p \text{Pl-Holant}(=_k, [1,0,z^{\frac{k-2}{2}}]|f). \end{aligned} \tag{39}$$

  Then by Lemma 14, we have

$$\begin{aligned} &\text{Pl-Holant}([1,0,\cdots,0,z^{-\frac{k(k-2)}{4}}], =_2 | =_2, (1, xz^{\frac{k-2}{4}}, yz^{\frac{k-2}{4}}, z^{\frac{k}{2}})) \\ &\leq_T^p \text{Pl-Holant}([1,0,\cdots,0,z^{-\frac{k(k-2)}{4}}], =_2 |(1, xz^{\frac{k-2}{4}}, yz^{\frac{k-2}{4}}, z^{\frac{k}{2}})). \end{aligned} \tag{40}$$

  In the left-hand side of (40) by adding $\frac{k-4}{2}$ loops to $[1,0,\cdots,0,z^{-\frac{k(k-2)}{4}}]$ using $=_2$, we have a signature of arity 4, $[1,0,0,0,z^{-\frac{k(k-2)}{4}}]$ on the LHS, i.e,

$$\begin{aligned} &\text{Pl-Holant}([1,0,0,0,z^{-\frac{k(k-2)}{4}}]|(1, xz^{\frac{k-2}{4}}, yz^{\frac{k-2}{4}}, z^{\frac{k}{2}})) \\ &\leq_T^p \text{Pl-Holant}([1,0,\cdots,0,z^{-\frac{k(k-2)}{4}}], =_2 | =_2, (1, xz^{\frac{k-2}{4}}, yz^{\frac{k-2}{4}}, z^{\frac{k}{2}})). \end{aligned} \tag{41}$$

  Note that $z^{\frac{k(k-2)}{4}} > 0$. Thus we can choose $z^{\frac{k(k-2)}{16}} \in \mathbb{R}$. By the holographic transformation $\begin{bmatrix} 1 & 0 \\ 0 & z^{\frac{k(k-2)}{16}} \end{bmatrix}$, we have

$$\begin{aligned} &\text{Pl-Holant}(=_4 |(1, xz^{-\frac{(k-2)(k-4)}{16}}, yz^{-\frac{(k-2)(k-4)}{16}}, z^{\frac{k(k+2)}{8}})) \\ &\leq_T^p \text{Pl-Holant}([1,0,0,0,z^{-\frac{k(k-2)}{4}}]|(1, xz^{\frac{k-2}{4}}, yz^{\frac{k-2}{4}}, z^{\frac{k}{2}})). \end{aligned} \tag{42}$$

  The remaining proof is similar with the previous case that $k \equiv 0 \pmod 4$ and we omit it here. ◀

Acknowledgement. We thank Kurt Kilpela for his contributions, especially for a conversation which triggered the discovery that incommensurability between tangent values and angles over π could be exploited. We thank Tonghai Yang for discussions on number theory. We also thank Yijia Chen, Pinyan Lu, Xiaoming Sun and Tyson Williams for insightful discussions.

References

1 Tom M. Apostol. Introduction to Analytic Number Theory. Undergraduate Texts in Mathematics, Springer, 1976.
2 Rodney J. Baxter. Exactly solved models in statistical mechanics. Academic Press London, 1982.
3 Miriam Backens: A New Holant Dichotomy Inspired by Quantum Computation. ICALP 2017: 16:1-16:14.
4 Andrei A. Bulatov: The complexity of the counting constraint satisfaction problem. J. ACM 60(5): 34:1-34:41 (2013).
5 Andrei A. Bulatov, Martin E. Dyer, Leslie Ann Goldberg, Mark Jerrum, Colin McQuillan: The expressibility of functions on the boolean domain, with applications to counting CSPs. J. ACM 60(5): 32:1-32:36 (2013).
6 Jin-Yi Cai, Xi Chen: Complexity of Counting CSP with Complex Weights. J. ACM 64(3): 19:1-19:39 (2017).
7 Jin-Yi Cai, Zhiguo Fu: Holographic algorithm with matchgates is universal for planar #CSP over boolean domain. STOC 2017: 842-855. The full version is available at https://arxiv.org/pdf/1603.07046.pdf.
8 Jin-Yi Cai, Aaron Gorenstein: Matchgates Revisited. Theory of Computing 10: 167-197 (2014).
9 Jin-Yi Cai, Pinyan Lu, Mingji Xia: The complexity of complex weighted Boolean #CSP. J. Comput. Syst. Sci. 80(1): 217-236 (2014).
10 Jin-Yi Cai, Michael Kowalczyk: Spin systems on k-regular graphs with complex edge functions. Theor. Comput. Sci. 461: 2-16 (2012).
11 Jin-Yi Cai, Michael Kowalczyk, Tyson Williams: Gadgets and anti-gadgets leading to a complexity dichotomy. ITCS 2012: 452-467. The full version is available at https://arxiv.org/pdf/1108.3383.pdf
12 S. Chowla. A special infinite series, Kong. Norsk. Vidensk. Selsk. Forhandl. 37, 85-87, (1964).
13 S. Chowla. The nonexistence of nontrivial relations between the roots of a certain irreducible equation, J. Number Th., 2, 120-123, (1970).
14 Martin E. Dyer, Leslie Ann Goldberg, Mark Jerrum: The Complexity of Weighted Boolean CSP. SIAM J. Comput. 38(5): 1970-1986 (2009).
15 Kurt Girstmair: Character coordinates and annihilators of cyclotomic numbers. Manuscripta Math. 59 (1987), no. 3, 375-389.
16 Heng Guo and Tyson Williams: The Complexity of Planar Boolean #CSP with Complex Weights. ICALP (1) 2013: 516-527. The full version is available at https://arxiv.org/pdf/1212.2284.pdf
17 Helmut Hasse: On a question of S. Chowla. Acta Arith. 18(1971), 275-280.
18 Sangxia Huang, Pinyan Lu: A Dichotomy for Real Weighted Holant Problems. Computational Complexity 25(1): 255-304 (2016).
19 H. Jager, H. W., Lenstra: Linear independence of cosecant values, Nieuw Archief Wisk. (3) 23, 131-144 (1975).

20 Richard E. Ladner: On the Structure of Polynomial Time Reducibility. J. ACM 22(1): 155-171 (1975).

21 H. W. Leopoldt, Über die Hauptordnung der ganzen Elemente eines abelschen Zahlkörpers, J. reine angew. Math. 201, 119-149 (1959).

22 Jiabao Lin, Hanpin Wang: The Complexity of Holant Problems over Boolean Domain with Non-Negative Weights. ICALP 2017: 29:1-29:14. The full version is available at https://arxiv.org/pdf/1611.00975.pdf.

23 P. W. Kasteleyn, The Statistics of Dimers on a Lattice, Physica 27 (1961) 1209-1225.

24 P. W. Kasteleyn, Graph Theory and Crystal Physics. In Graph Theory and Theoretical Physics, (F. Harary, ed.), Academic Press, London, 43-110 (1967).

25 Michael Kowalczyk, Jin-Yi Cai: Holant Problems for 3-Regular Graphs with Complex Edge Functions. Theory Comput. Syst. 59(1): 133-158 (2016).

26 H. N. V. Temperley and M. E. Fisher. Dimer Problem in Statistical Mechanics - an Exact Result. Philosophical Magazine 6: 1061- 1063 (1961).

27 Leslie G. Valiant: Holographic Algorithms. SIAM J. Comput. 37(5): 1565-1594 (2008).

28 L. C. Washington, Introduction to Cyclotomic Fields, Graduate Text in Mathematics, 83, Springer, 1980.

Quantum Query Algorithms are Completely Bounded Forms

Srinivasan Arunachalam[*1], **Jop Briët**[†2], **and Carlos Palazuelos**[‡3]

1 **CWI, QuSoft and University of Amsterdam, The Netherlands**
arunacha@cwi.nl
2 **CWI, QuSoft, The Netherlands**
j.briet@cwi.nl
3 **Facultad de C.C. Matematicas, UCM. Instituto de Ciencias Matematicas, Madrid, Spain**
carlospalazuelos@mat.ucm.es

Abstract

We prove a characterization of quantum query algorithms in terms of polynomials satisfying a certain (completely bounded) norm constraint. Based on this, we obtain a refined notion of approximate polynomial degree that equals the quantum query complexity, answering a question of Aaronson et al. (CCC'16). Using this characterization, we show that many polynomials of degree at least 4 are far from those coming from quantum query algorithms. Our proof is based on a fundamental result of Christensen and Sinclair (*J. Funct. Anal.*, 1987) that generalizes the well-known Stinespring representation for quantum channels to multilinear forms. We also give a simple and short proof of one of the results of Aaronson et al. showing an equivalence between one-query quantum algorithms and bounded quadratic polynomials.

1998 ACM Subject Classification F.1.1 Models of Computation, G.1.2 Approximation

Keywords and phrases Quantum query algorithms, operator space theory, polynomial method, approximate degree

Digital Object Identifier 10.4230/LIPIcs.ITCS.2018.3

1 Introduction

In the black-box model of quantum computation one is given access to a unitary operation, usually referred to as an oracle, that allows one to probe the bits of an unknown binary string $x \in \{-1,1\}^n$ in superposition. Promised that x lies in a subset $D \subseteq \{-1,1\}^n$, the goal in this model is to learn some property of x given by a Boolean function $f : D \to \{-1,1\}$, when only given access to x through the oracle. An application of the oracle is usually referred to as a *query*. The bounded-error quantum query complexity of f, denoted $Q_\varepsilon(f)$, is the minimal number of queries a quantum algorithm must make on the worst-case input $x \in D$ to compute $f(x)$ with probability at least $1-\varepsilon$, where $\varepsilon \in (0, 1/2)$ is usually some fixed but arbitrary positive constant.

* S. A. was supported by ERC Consolidator Grant QPROGRESS.
† J. B. was supported by a VENI grant and the Gravitation-grant NETWORKS-024.002.003 from the Netherlands Organisation for Scientific Research (NWO).
‡ C. P. was supported by the Ramon y Cajal program (RYC-2012-10449), the Spanish MINECO MTM2014-54240-P, Comunidad de Madrid (QUITEMAD+ Project S2013/ICE-2801) and ICMAT Severo Ochoa Grant No. SEV-2011-0087.

Many of the best-known quantum algorithms are naturally captured by this model. Famous partial functions whose quantum query complexity is exponentially smaller than their classical counterpart (the decision-tree complexity) are period finding [48], Simon's problem [49] and Forrelation [1]. Famous problems related to total functions that admit polynomial quantum speed-ups include unstructured search [26], element distinctness [8] and NAND-tree evaluation [23]. It is well-known that for all total functions, the quantum and classical query complexities are polynomially related [11]; see Ambainis et al. [9] and Aaronson et al. [3] for recent progress on the largest possible separations.

Despite the simplicity of the query model, determining the quantum query complexity of a given function f appears to be highly non-trivial. Several methods were introduced to tackle this problem. For constructing quantum query algorithms, there are general methods based on quantum walks [8, 33], span programs [46] and learning graphs [12]. For proving lower bounds there are two main methods, known as the *polynomial method* [11] and the *adversary method* [6]. The latter was eventually generalized to the "negative weight" adversary method [27] and was shown to *characterize* quantum query complexity [27, 46, 47, 32], but proving lower bounds using this method appears to be hard in general. This paper will focus on the polynomial method.

1.1 The polynomial method

The polynomial method is based on a connection between quantum query algorithms and polynomials discovered by Beals et al. [11]. They observed that for every t-query quantum algorithm $\mathcal{A}$ that on input $x \in \{-1,1\}^n$ returns a random sign $\mathcal{A}(x)$, there exists a degree-$(2t)$ polynomial p such that $p(x) = \mathbf{E}[\mathcal{A}(x)]$ for every x (where the expectation is taken over the randomness of the output). It follows that if $\mathcal{A}$ computes $f : \{-1,1\}^n \to \{-1,1\}$ with probability at least $1-\varepsilon$, then p satisfies $|p(x) - f(x)| \leq 2\varepsilon$ for every x. The polynomial method thus converts the problem of lower bounding quantum query complexity to the problem of proving lower bounds on the minimum degree of a polynomial p such that $|p(x) - f(x)| \leq 2\varepsilon$ holds for all inputs x. The minimal degree of such a polynomial is called the *approximate (polynomial) degree* and is denoted by $\deg_\varepsilon(f)$. Notable applications of this approach showed optimality for Grover's search algorithm [11][1] and the above-mentioned algorithms for collision-finding and element distinctness [4]. In a recent work, Bun et al. [18] use the polynomial method to resolve the quantum query complexity of several other well-studied Boolean functions.

Converses to the polynomial method. A natural question is whether the polynomial method admits a converse. If so, this would imply a succinct characterization of quantum algorithms in terms of basic mathematical objects. However, Ambainis [7] answered this question in the negative, showing that for infinitely many n, there is a function f with $\deg_{1/3}(f) \leq n^\alpha$ and $Q_{1/3}(f) \geq n^\beta$ for some positive constants $\beta > \alpha$ (recently larger separations were obtained in [3]). The approximate degree thus turns out to be an imprecise measure for quantum query complexity in general. These negative results leave open the following two possibilities:

1. There is a (simple) refinement of approximate polynomial degree that approximates $Q_\varepsilon(f)$ up to a constant factor.

[1] The first quantum lower bound for the search problem was proven by Bennett et al. [13] using the so-called hybrid method. Beals et al. [11] reproved their result using the polynomial method.

2. Constant-degree polynomials characterize constant-query quantum algorithms.

These avenues were recently explored by Aaronson et al. [1, 2]. The first work strengthened the polynomial method by observing that quantum algorithms give rise to polynomials with a so-called *block-multilinear* structure. Based on this observation, they introduced a refined degree measure, bm-deg$_\varepsilon(f)$ which lies between deg$_\varepsilon(f)$ and $2Q_\varepsilon(f)$, prompting the immediate question of how well that approximates $Q_\varepsilon(f)$. The subsequent work showed, among other things, that for infinitely many n, there is a function f with bm-deg$_{1/3}(f) = O(\sqrt{n})$ and $Q_{1/3}(f) = \Omega(n)$, thereby also ruling out the possibility that this degree measure validates possibility 1. The natural next question then asks if there is another refined notion of polynomial degree that approximates quantum query complexity [2, Open problem 3].

In the direction of the second avenue, [2] showed a surprising converse to the polynomial method for quadratic polynomials. Say that a polynomial $p \in \mathbb{R}[x_1, \ldots, x_n]$ is *bounded* if it satisfies $p(x) \in [-1, 1]$ for all $x \in \{-1, 1\}^n$.

▶ **Theorem 1** (Aaronson et al.). *There exists an absolute constant $C \in (0, 1]$ such that the following holds. For every bounded quadratic polynomial p, there exists a one-query quantum algorithm that, on input $x \in \{-1, 1\}^n$, returns a random sign with expectation $Cp(x)$.*

This implies that item 2 holds true for quadratic polynomials. It also leads to the problem of finding a similar converse for higher-degree polynomials, asking for instance whether two-query quantum algorithms are equivalent to quartic polynomials [2, Open problem 1].

1.2 Our results

This paper addresses the above-mentioned two problems. Our first result is a new notion of polynomial degree that gives a tight characterization of quantum query complexity (Definition 4 and Corollary 5 below), giving an answer to [2, Open problem 3]. Using this characterization, we show that there is no generalization of Theorem 1 to higher-degree polynomials, in the sense that there is no absolute constant $C \in (0, 1]$ for which the analogous statement holds true. This gives a partial answer to [2, Open problem 1], ruling out a strong kind of equivalence. Finally, we give a simplified shorter proof of Theorem 1. Below we explain our results in more detail.

Quantum algorithms are completely bounded forms. For the rest of the discussion, all polynomials will be assumed to be bounded, real and $(2n)$-variate if not stated otherwise. We refer to a homogeneous polynomial as a *form*. For $\alpha \in \{0, 1, 2, \ldots\}^{2n}$ and $x \in \mathbb{R}^{2n}$, write $|\alpha| = \alpha_1 + \cdots + \alpha_{2n}$ and $x^\alpha = x_1^{\alpha_1} \cdots x_{2n}^{\alpha_{2n}}$. Then, a form p of degree t can be written as

$$p(x) = \sum_{\alpha \in \{0,1,\ldots,t\}^{2n} : |\alpha| = t} c_\alpha x^\alpha, \tag{1}$$

where c_α are some real coefficients. Our new notion of polynomial degree is based on a characterization of quantum query algorithms in terms of forms satisfying a certain norm constraint. The norm we assign to a form as in (1) is given by a norm of the symmetric t-tensor $T_p \in \mathbb{R}^{2n \times \cdots \times 2n}$ with $(i_1, \ldots, i_t)$-coordinate

$$(T_p)_{i_1,\ldots,i_t} = \frac{c_{e_{i_1} + \cdots + e_{i_t}}}{|\{i_1, \ldots, i_t\}|!}, \tag{2}$$

where e_i is the ith standard basis vector for $\mathbb{R}^{2n}$ and $|\{i_1, \ldots, i_t\}|$ denotes the number of distinct elements in the set $\{i_1, \ldots, i_t\}$. Note that p can then also be written as

$$p(x) = \sum_{i_1,\ldots,i_t=1}^{2n} (T_p)_{i_1,\ldots,i_t} x_{i_1} \cdots x_{i_t}. \tag{3}$$

The relevant norm of T_p is in turn given in terms of an infimum over decompositions of the form $T_p = \sum_{\sigma \in S_t} T^\sigma \circ \sigma$, where the sum is over permutations of $\{1, \ldots, t\}$, each T^σ is a t-tensor, and $T^\sigma \circ \sigma$ is the permuted version of T^σ given by

$$(T^\sigma \circ \sigma)_{i_1,\ldots,i_t} = T^\sigma_{i_{\sigma(1)},\ldots,i_{\sigma(t)}}.$$

Finally, the actual norm is based on the *completely bounded norm* of each of the T^σ. Given a t-tensor $T \in \mathbb{R}^{2n \times \cdots \times 2n}$, its completely bounded norm $\|T\|_{\mathrm{cb}}$ is given by the supremum over positive integers k and collections of $k \times k$ unitary matrices $U_1(i), \ldots, U_t(i)$, for $i \in [2n]$, of the operator norm

$$\Big\| \sum_{i_1,\ldots,i_t=1}^{2n} T_{i_1,\ldots,i_t} U_1(i_1) \cdots U_t(i_t) \Big\|. \tag{4}$$

▶ **Definition 2** (Completely bounded norm of a form)**.** Let p be a form of degree t and let T_p be the symmetric t-tensor as in (2). Then, the *completely bounded norm* of p is defined by

$$\|p\|_{\mathrm{cb}} = \inf \Big\{ \sum_{\sigma \in S_t} \|T^\sigma\|_{\mathrm{cb}} \ : \ T_p = \sum_{\sigma \in S_t} T^\sigma \circ \sigma \Big\}. \tag{5}$$

This norm was originally introduced in the general context of tensor products of operator spaces in [37]. In that framework, the definition considered here corresponds to a particular operator space based on ℓ_1^n, but we shall not use this fact here. Our characterization of quantum query algorithms is as follows.

▶ **Theorem 3** (Characterization of quantum algorithms)**.** *Let $\beta : \{-1,1\}^n \to [-1,1]$ and let t be a positive integer. Then, the following are equivalent.*

1. *There exists a form p of degree $2t$ such that $\|p\|_{\mathrm{cb}} \leq 1$ and $p((x, \mathbf{1})) = \beta(x)$ for every $x \in \{-1,1\}^n$, where $\mathbf{1} \in \mathbb{R}^n$ is the all-ones vector.*
2. *There exists a t-query quantum algorithm that, on input $x \in \{-1,1\}^n$, returns a random sign with expected value $\beta(x)$.*

It may be observed that the content of the polynomial method is contained in the above statement, since any $(2n)$-variate form p defines an n-variate polynomial given by $q(x) = p((x, \mathbf{1}))$. The above theorem refines the polynomial method in the sense that quantum algorithms can only yield polynomials of the form $q(x) = p((x, \mathbf{1}))$ where p has completely bounded norm at most one. Our proof is based on a fundamental result of Christensen and Sinclair [19] concerning multilinear forms on C^*-algebras that generalizes the well-known Stinespring representation theorem for quantum channels (see also [40] and [42, Chapter 5]).

Completely bounded approximate degree. Theorem 3 motivates the following new notion of approximate degree for partial Boolean functions.

▶ **Definition 4** (Completely bounded approximate degree)**.** For $D \subseteq \{-1,1\}^n$, let $f : D \to \{-1,1\}$ be a (possibly partial) Boolean function and let $\varepsilon > 0$. Then, the *ε-completely bounded approximate degree* of f, denoted $\text{cb-deg}_\varepsilon(f)$, is the smallest positive integer t for which there exists a form p of degree $2t$ such that $\|p\|_{\mathrm{cb}} \leq 1$ as in Eq. (5) and we have $|p((x, \mathbf{1})) - f(x)| \leq 2\varepsilon$ for every $x \in D$.

As a corollary of Theorem 3, we get the following characterization of quantum query complexity.

▶ **Corollary 5.** *For $D \subseteq \{-1,1\}^n$, $f : D \to \{-1,1\}$ and $\varepsilon > 0$, we have $cb\text{-}deg_\varepsilon(f) = Q_\varepsilon(f)$.*

Separations for higher-degree forms. Theorem 1 follows from our Theorem 3 and the fact that for every bounded quadratic form $p(x) = x^T Ax$, the matrix A has completely bounded norm bounded from above by an absolute constant (independent on n); this is discussed in more detail below. If the same were true for the tensors T_p corresponding to higher-degree forms p then Theorem 3 would clearly give higher-degree extensions of Theorem 1. Unfortunately, this is false. Bounded forms whose associated tensors have unbounded completely bounded norm appeared before in the work of Smith [50], who gave an explicit example with completely bounded norm $\sqrt{\log n}$. Since $\|p\|_{\mathrm{cb}}$ involves an infimum over decompositions of T_p, this does not yet imply a counterexample to higher-degree versions of Theorem 1. However, such counterexamples are implied by recent work on Bell inequalities, multiplayer XOR games in particular. It is not difficult to see that $\|p\|_{\mathrm{cb}}$ is bounded from below by the so-called *jointly completely bounded norm* of the tensor T_p, a quantity that in quantum information theory is better known as the entangled bias of the XOR game whose (unnormalized) game tensor is given by T_p. One obtains this quantity by inserting tensor products between the unitaries appearing in (4). Pérez-García et al. [41] and Vidick and the second author [17] gave examples of bounded cubic forms with unbounded jointly completely bounded norm. Both constructions are non-explicit, the first giving a completely bounded norm of order $\Omega((\log n)^{1/4})$ and the latter of order $\widetilde{\Omega}(n^{1/4})$. Here, we explain how to get a larger separation by means of a much simpler (although still non-explicit) construction and show that a bounded cubic form p given by a suitably normalized random sign tensor has completely bounded norm $\|p\|_{\mathrm{cb}} = \Omega(\sqrt{n})$ with high probability (Theorem 11). The result presented here is not new, but it follows from the existence of commutative operator algebras which are not Q-algebras. Here, we present a self-contained proof which follows the same lines as in [22, Theorem 18.16] and, in addition, we prove the result with high probability (rather than just the existence of such trilinear forms). We also explain how to obtain from this result quartic examples by embedding into 3-dimensional "tensor slices", which in turn imply counterexamples to a quartic versus two-query version of Theorem 1.

Short proof of Theorem 1. As shown in [2], Theorem 1 is yet another surprising consequence of the ubiquitous Grothendieck inequality [25] (Theorem 17 below), well known for its relevance to Bell inequalities [53, 20] and combinatorial optimization [5, 30], not to mention its fundamental importance to Banach spaces [43]. An equivalent formulation of Grothendieck's inequality again recovers Theorem 1 for quadratic forms $p(x) = x^{\mathsf{T}} Ax$ given by a matrix $A \in \mathbb{R}^{n\times n}$ satisfying a certain norm constraint $\|A\|_{\ell_\infty \to \ell_1} \leq 1$, which in particular implies that p is bounded (see Section 2 for more on this norm). Indeed, in that case Grothendieck's inequality implies that $\|A\|_{\mathrm{cb}} \leq K_G$ for some absolute constant $K_G \in (0,\infty)$ (independent of n and A). Normalizing by K_G^{-1}, one obtains Theorem 1 with $C = K_G^{-1}$ for such quadratic forms from Theorem 3. The general version of Theorem 1 for quadratic polynomials follows from this via a so-called decoupling argument (see Section 5). This arguably does not simplify the original proof of Theorem 1, as Theorem 3 relies on deep results itself. However, in Section 5 we give a short simplified proof, showing that Theorem 1 follows almost directly from a "factorization version" of Grothendieck's inequality (Theorem 18) that follows from the more standard version (Theorem 17). The factorization version was used in the original proof as well, but only as a lemma in a more intricate argument. In computer science, this

factorization version already found applications in an algorithmic version of the Bourgain–Tzafriri Column Subset Theorem [52] and algorithms for community detection in the stochastic block model [31]. This appears to be its first occurrence in quantum computing.

1.3 Related work

Although there is no converse to the polynomial method for arbitrary polynomials, equivalences between quantum algorithms and polynomials have been studied before in certain models of computation. For example, we do know of such characterization in the model of non-deterministic query complexity [54], the unbounded-error query complexity [35] and quantum query complexity in expectation [29]. In all these settings, the quantum algorithms constructed from polynomials were *non-adaptive* algorithms, i.e., the quantum algorithm begins with a quantum state, repeatedly applies the oracle some fixed number of times and then performs a projective measurement. Crucially, these algorithms do not contain interlacing unitaries that are present in the standard model of query complexity, hence are known to be a much weaker class of algorithms (see Montanaro [34] for more details).

Our main result is yet another demonstration of the expressive power of C^*-algebras and operator space theory in quantum information theory; for a survey on applications of these areas to two-prover one-round games, see [38]. The appearance of Q-algebras (mentioned in the above paragraph on separations) is also not a first in quantum information theory, see for instance [41, 15, 16].

1.4 Organization

In Section 2, we give a brief introduction to normed vector spaces, C^*-algebras and define the model of quantum query complexity. In Section 3, we prove our main theorem characterizing quantum query algorithms. In Section 4, we explain the separation obtained for higher-degree forms. In Section 5, we give a short proof of the main theorem in Aaronson et al. [2].

2 Preliminaries

Notation. For a positive integer t denote $[t] = \{1, \ldots, t\}$. For $x \in \mathbb{C}^n$, let $\mathrm{Diag}(x)$ be the $n \times n$ diagonal matrix whose diagonal forms x. Given a matrix $X \in \mathbb{C}^{n\times n}$, let $\mathrm{diag}(X) \in \mathbb{C}^n$ denote its diagonal vector. For $x \in \{0,1\}^n$, denote $(-1)^x = ((-1)^{x_1}, \ldots, (-1)^{x_n})$. Let $e_1, e_2, \ldots, e_n \in \mathbb{C}^n$ be the standard basis vectors and let $E_{ij} = e_i e_j^*$. For $i, j \in [n]$, let $\delta_{i,j}$ be the indicator for the event $[i = j]$. Let $\mathbf{1} = (1, \ldots, 1)$ and $\mathbf{0} = (0, \ldots, 0)$ denote the n-dimensional all-ones (resp. all-zeros) vector.

Normed vector spaces. For parameter $p \in [1, \infty)$, the p-norm of a vector $x \in \mathbb{R}^n$ is defined by $\|x\|_{\ell_p} = (|x_1|^p + \cdots + |x_n|^p)^{1/p}$ and for $p = \infty$ by $\|x\|_{\ell_\infty} = \max\{|x_i| : i \in [n]\}$. Denote the n-dimensional Euclidean unit ball by $B_2^n = \{x \in \mathbb{R}^n : \|x\|_{\ell_2} \leq 1\}$. For a matrix $A \in \mathbb{R}^{n\times n}$, denote the standard operator norm by $\|A\|$ and define

$$\|A\|_{\ell_\infty \to \ell_1} = \sup\left\{\|Ax\|_{\ell_1} : \|x\|_{\ell_\infty} \leq 1\right\} = \max_{x,y\in\{-1,1\}^n} x^{\mathsf{T}} A y.$$

We denote the norm of a general normed vector space X by $\|\cdot\|_X$, if there is a danger of ambiguity. Denote by Id_X the identity map on X and by Id_d the identity map on $\mathbb{C}^d$. For normed vector spaces X, Y, let $L(X, Y)$ be the collection of all linear maps $T : X \to Y$. We will use the notation $L(X)$ as a shorthand for $L(X, X)$. The (operator) norm of a linear

map $T \in L(X,Y)$ is given by $\|T\| = \sup\{\|T(x)\|_Y : \|x\|_X \leq 1\}$. Such a map is an *isometry* if $\|T(x)\|_Y = \|x\|_X$ for every $x \in X$ and a *contraction* if $\|T(x)\|_Y \leq \|x\|$ for every $x \in X$. Throughout we endow $\mathbb{C}^d$ with the standard Euclidean norm. Note that the space $L(\mathbb{C}^d)$ is naturally identified with the set of $d \times d$ matrices, sometimes denoted $M_d(\mathbb{C})$, and we use the two notations interchangeably. For a Hilbert space $\mathcal{H}$, we endow $\mathcal{H} \otimes \mathbb{C}^d$ with the norm given by the inner product $\langle f \otimes a, g \otimes b\rangle = \langle f, g\rangle_{\mathcal{H}} \langle a, b\rangle$, making this space isometric to $\mathcal{H} \oplus \cdots \oplus \mathcal{H}$ (d times). Similarly, we endow $L(\mathcal{H}) \otimes L(\mathbb{C}^d)$ with the operator norm of the space $L(\mathcal{H} \otimes \mathbb{C}^d)$ of linear operators on the Hilbert space $\mathcal{H} \otimes \mathbb{C}^d$; with some abuse of notation, we shall identify the two spaces of operators.

C^*-algebras. We collect a few basic facts of C^*-algebras that we use later and refer to [10] for an extensive introduction. A C^*-algebra $\mathcal{X} = (X, \cdot, *)$ is a normed complex vector space X, complete with respect to its norm (i.e., a Banach space), that is endowed with two operations in addition to the standard vector-space addition and scalar multiplication operations:

1. an associative multiplication $\cdot : X \times X \to X$, denoted $x \cdot y$ for $x, y \in X$, that is distributive with respect to the vector space addition and continuous with respect to the norm of X, which is to say that $\|x \cdot y\|_X \leq \|x\|_X \|y\|_X$ for all $x, y \in X$;
2. an involution $* : X \to X$, that is, a conjugate linear map that sends $x \in X$ to (a unique) $x^* \in X$ satisfying $(x^*)^* = x$ and $(xy)^* = y^* x^*$ for any $x, y \in X$, and such that $\|x \cdot x^*\|_X = \|x\|_X^2$.

Any finite-dimensional normed vector space is a Banach space. A C^*-algebra $\mathcal{X}$ is *unital* if it has a multiplicative identity, denoted $\mathrm{Id}_{\mathcal{X}}$. The most important example of a unital C^*-algebra is $M_n(\mathbb{C})$, where the involution operator is the conjugate-transpose and the norm is the operator norm. A linear map $\pi : \mathcal{X} \to \mathcal{Y}$ from one C^*-algebra $\mathcal{X}$ to another $\mathcal{Y}$ is a *$*$-homomorphism* if it preserves the multiplication operation, $\pi(xy) = \pi(x)\pi(y)$, and satisfies $\pi(x)^* = \pi(x^*)$ for all $x, y \in \mathcal{X}$. For a complex Hilbert space $\mathcal{H}$, a mapping $\pi : \mathcal{X} \to L(\mathcal{H})$ is a *$*$-representation* if it is a $*$-homomorphism. An important fact is the Gelfand–Naimark Theorem [36, Theorem 3.4.1] asserting that any C^*-algebra admits an isometric (that is, norm-preserving) $*$-representation for some complex Hilbert space.

Completely bounded norms. We also collect a few basic facts about completely bounded norms that we use later and refer to [39] for an extensive introduction. For a C^*-algebra $\mathcal{X}$ and positive integer d, we denote by $M_d(\mathcal{X})$ the set of d-by-d matrices with entries in $\mathcal{X}$. Note that this set can naturally be identified with the algebraic tensor product $\mathcal{X} \otimes L(\mathbb{C}^d)$, that is, the linear span of all elements of the form $x \otimes M$, where $x \in X$ and $M \in L(\mathbb{C}^d)$. We shall endow $M_d(\mathcal{X})$ with a norm induced by an isometric $*$-representation π of $\mathcal{X}$ into $L(\mathcal{H})$ for a Hilbert space $\mathcal{H}$. The linear map $\pi \otimes \mathrm{Id}_{L(\mathbb{C}^d)}$ sends elements in $M_d(\mathcal{X})$ (or $\mathcal{X} \otimes L(\mathbb{C}^d)$) to elements (operators) in $L(\mathcal{H} \otimes \mathbb{C}^d)$. The norm of an element $A \in M_d(\mathcal{X})$ is then defined to be $\|A\| = \|(\pi \otimes \mathrm{Id}_{L(\mathbb{C}^d)})(A)\|$. The notation $\|A\|$ reflects the fact that this norm is in fact independent of the particular $*$-representation π. Based on this, we can define a norm on linear maps $\sigma : \mathcal{X} \to L(\mathcal{H})$ as follows:

$$\|\sigma\|_{\mathrm{cb}} = \sup\left\{ \frac{\|(\sigma \otimes \mathrm{Id}_{L(\mathbb{C}^d)})(A)\|}{\|A\|} : d \in \mathbb{N},\ A \in \mathcal{X} \otimes L(\mathbb{C}^d),\ A \neq 0 \right\}$$

Tensors and multilinear forms. For vector spaces X, Y over the same field and positive integer t, recall that a mapping

$$T : \underbrace{X \times \cdots \times X}_{t \text{ times}} \to Y$$

is t-*linear* if for every $x_1, \ldots, x_t \in X$ and $i \in [t]$, the map $y \mapsto T(x_1, \ldots, x_{i-1}, y, x_{i+1}, \ldots, x_t)$ is linear. A t-*tensor* of dimension n is a map $T : [n] \times \cdots \times [n] \to \mathbb{C}$, which can alternatively be identified by $T = (T_{i_1,\ldots,i_t})_{i_1,\ldots,i_t=1}^{n} \in \mathbb{C}^{n \times \cdots \times n}$. With abuse of notation we identify a t-tensor $T \in \mathbb{C}^{n \times \cdots \times n}$ with the t-linear form $T : \mathbb{C}^n \times \cdots \times \mathbb{C}^n \to \mathbb{C}$ given by

$$T(x_1, \ldots, x_t) = \sum_{i_1,\ldots,i_t=1}^{n} T_{i_1,\ldots,i_t} x_1(i_1) \cdots x_t(i_t).$$

Next, we introduce the completely bounded norm of a t-linear form $T : \mathcal{X} \times \cdots \times \mathcal{X} \to \mathbb{C}$ on a C^*-algebra $\mathcal{X}$. First, we use the standard identification of such forms with the linear form on the tensor product $\mathcal{X} \otimes \cdots \otimes \mathcal{X}$ given by $T(x_1 \otimes \cdots \otimes x_t) = T(x_1, \ldots, x_t)$. We consider a bilinear map $\odot : \big(\mathcal{X} \otimes L(\mathbb{C}^d), \mathcal{X} \otimes L(\mathbb{C}^d)\big) \to \mathcal{X} \otimes \mathcal{X} \otimes L(\mathbb{C}^d)$ for any positive integer d defined as follows. For $x, y \in \mathcal{X}$ and $M_x, M_y \in L(\mathbb{C}^d)$, let

$$(x \otimes M_x) \odot (y \otimes M_y) = (x \otimes y) \otimes (M_x M_y).$$

Observe that this operation changes the order of the tensor factors and *multiplies* M_x with M_y. This operation is associative but *not* commutative. Extend the definition of the $\odot$ operation bi-linearly to its entire domain. Define the t-linear map $T_d : M_d(\mathcal{X}) \times \cdots \times M_d(\mathcal{X}) \to L(\mathbb{C}^d)$ by

$$T_d(A_1 \ldots, A_t) = \big(T \otimes \mathrm{Id}_{L(\mathbb{C}^d)}\big)(A_1 \odot \cdots \odot A_t).$$

The completely bounded norm of T is now defined by

$$\|T\|_{\mathrm{cb}} = \sup\Big\{ \big\|T_d(A_1, \ldots, A_t)\big\| : d \in \mathbb{N},\ \ A_j \in M_d(\mathcal{X}),\ \ \|A_j\| \leq 1 \Big\}.$$

Note that the definition given in (4) corresponds to the particular case where the C^*-algebra $\mathcal{X}$ is formed by the $n \times n$ diagonal matrices. Since any square matrix with operator norm at most 1 is a convex combination of unitary matrices (by the Russo-Dye Theorem)[2], the completely bounded norm can also be defined by taking the supremum over unitaries $A_j \in M_d(\mathcal{X})$. The completely bounded norm can be defined more generally for multilinear maps into $L(\mathcal{H})$, for an Hilbert space $\mathcal{H}$, to yield the definition of this norm for linear maps given above, but we will not use this here.

Quantum query complexity. The quantum query model was formally defined by Beals et al. in [11]. In this model, we are given black-box access to a unitary operator, often called an oracle O_x, whose description depends in a simple way on some binary input string $x \in \{0,1\}^n$. An application of the oracle on a quantum register is referred to as a quantum *query* to x. In the standard form of the model, a query acts on a pair of registers on (Q, A), where Q is an n-dimensional query register and A is a one-qubit auxiliary register. A query to the oracle effects the unitary transformation given by

$$O_x : |i, b\rangle \to |i, b \oplus x_i\rangle$$

where $i \in [n]$, $b \in \{0,1\}$. (These oracles are also commonly called *bit oracles*.)

[2] A precise statement and short proof of the Russo-Dye theorem can be found in [24].

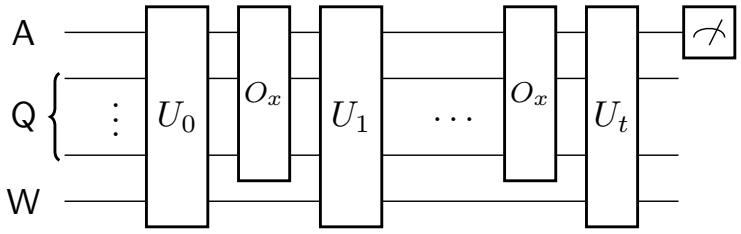


Figure 1 A t-query quantum algorithm that starts with the all-zero state and concludes by measuring the register A.

A quantum query algorithm consists of a fixed sequence of unitary operations acting on (Q, A) in addition to a *workspace* register W. A t-query quantum algorithm begins by initializing the joint register $(\mathsf{Q}, \mathsf{A}, \mathsf{W})$ in the all-zero state and continues by interleaving a sequence of unitaries $U_0, \ldots, U_t$ on $(\mathsf{Q}, \mathsf{A}, \mathsf{W})$ with oracles O_x on (Q, A). Finally, the algorithm performs a 2-outcome measurement on A and returns the measurement outcome.

For a Boolean function $f : \{0,1\}^n \to \{0,1\}$, the algorithm is said to compute f with error $\varepsilon > 0$ if for every x, the measurement outcome of register A equals $f(x)$ with probability at least $1 - \varepsilon$. The *bounded-error query complexity* of f, denoted $Q_\varepsilon(f)$, is the smallest t for which such an algorithm exists. Note that in this model, we are not concerned with the amount of time (i.e., the number of gates) it takes to implement the interlacing unitaries, which could be much bigger than the query complexity itself.

Here we will work with a slightly less standard oracle sometime referred to as a *phase oracle*, in which the standard oracle is preceded and followed by a Hadamard on A. Since the Hadamards can be undone by the unitaries surrounding the queries in a quantum query algorithm, using the phase oracle does not reduce generality. A query to this oracle, sometimes denoted $O_{x,\pm}$, applies the (controlled) unitary $\mathrm{Diag}((\mathbf{1}, (-1)^x))$ to joint register (A, Q). To avoid having to write $(-1)^x$ later on, we shall work in the equivalent setting where Boolean functions send $\{-1,1\}^n$ to $\{-1,1\}$.

3 Characterization of quantum query algorithms

In this section we prove Theorem 3. The main ingredient of the proof is the following celebrated representation theorem by Christensen and Sinclair [19] showing that completely-boundedness of a multilinear form is equivalent to the existence of an exceedingly nice factorization.

▶ **Theorem 6** (Christensen–Sinclair). *Let t be a positive integer and let $\mathcal{X}$ be a C^*-algebra. Then, for any t-linear form $T : \mathcal{X} \times \cdots \times \mathcal{X} \to \mathbb{C}$, we have $\|T\|_{\mathrm{cb}} \leq 1$ if and only if there exist Hilbert spaces $\mathcal{H}_0, \ldots, \mathcal{H}_{t+1}$ where $\mathcal{H}_0 = \mathcal{H}_{t+1} = \mathbb{C}$, $*$-representations $\pi_i : \mathcal{X} \to L(\mathcal{H}_i)$ for each $i \in [t]$ and contractions $V_i \in L(\mathcal{H}_i, \mathcal{H}_{i-1})$, for each $i \in [t+1]$ such that for any $x_1, \ldots, x_t \in \mathcal{X}$, we have*

$$T(x_1, \ldots, x_t) = V_1 \pi_1(x_1) V_2 \pi_2(x_2) V_3 \cdots V_t \pi_t(x_t) V_{t+1}. \tag{6}$$

We first show how the above result simplifies when restricting to the special case in which the C^*-algebra $\mathcal{X}$ is formed by the set of diagonal n-by-n matrices.

▶ **Corollary 7.** *Let m, n, t be positive integers such that $t \geq 2$ and $m = n^t$. Let $T \in \mathbb{C}^{n\times\cdots\times n}$ be a t-tensor. Then, $\|T\|_{\mathrm{cb}} \leq 1$ if and only if there exist a positive integer d, unit vectors $u, v \in \mathbb{C}^m$ and contractions $U_i, V_i \in L(\mathbb{C}^m, \mathbb{C}^{dn})$ such that for any $x_1, \ldots, x_t \in \mathbb{C}^n$, we have*

$$T(x_1, \ldots, x_t) = u^* U_1^* \big(\operatorname{Diag}(x_1) \otimes \mathrm{Id}_d\big) V_1 \cdots U_t^* \big(\operatorname{Diag}(x_t) \otimes \mathrm{Id}_d\big) V_t v. \tag{7}$$

The proof of the above corollary uses the following fact about the completely bounded norm of $*$-representations of C^*-algebras [42, Theorem 1.6].

▶ **Lemma 8.** *Let $\mathcal{X}$ be a finite-dimensional C^*-algebra, $\mathcal{H}, \mathcal{H}'$ be Hilbert spaces, $\pi : \mathcal{X} \to L(\mathcal{H})$ be a $*$-representation and $U \in L(\mathcal{H}, \mathcal{H}')$ and $V \in L(\mathcal{H}', \mathcal{H})$ be linear maps. Then, the map $\sigma : \mathcal{X} \to L(\mathcal{H}')$, defined as $\sigma(x) = U\pi(x)V$, satisfies that $\|\sigma\|_{\mathrm{cb}} \leq \|U\|\|V\|$.*

In addition, we use the famous Fundamental Factorization Theorem [39, Theorem 8.4]. Below we state the theorem when restricted to finite-dimensional spaces (see also remark after [28, Theorem 16]).

▶ **Theorem 9** (Fundamental factorization theorem)**.** *Let $\sigma : L(\mathbb{C}^n) \to L(\mathbb{C}^m)$ be a linear map and let $d = nm$. Then, there exist $U, V \in L(\mathbb{C}^m, \mathbb{C}^{dn})$ such that $\|U\|\|V\| \leq \|\sigma\|_{\mathrm{cb}}$ and for any $M \in L(\mathbb{C}^n)$, we have $\sigma(M) = U^*(M \otimes \mathrm{Id}_d)V$.*

Proof of Corollary 7. The set $\mathcal{X} = \operatorname{Diag}(\mathbb{C}^n)$ of diagonal matrices is a (finite-dimensional) C^*-algebra (endowed with the standard matrix product and conjugate-transpose involution). Define the t-linear form $R : \mathcal{X} \times \cdots \times \mathcal{X} \to \mathbb{C}$ by $R(X_1, \ldots, X_t) = T(\operatorname{diag}(X_1), \ldots, \operatorname{diag}(X_t))$. We claim that $\|R\|_{\mathrm{cb}} = \|T\|_{\mathrm{cb}}$. Observe that for every positive integer d, the set $\{B \in M_d(\mathcal{X}) : \|B\| \leq 1\}$ can be identified with the set of block-diagonal matrices $B = \sum_{i=1}^n E_{i,i} \otimes B(i)$ of size $nd \times nd$ and blocks $B(1), \ldots, B(n)$ of size $d \times d$ satisfying $\|B(i)\| \leq 1$ for all $i \in [n]$. It follows that

$$\begin{aligned} R_d(B_1, \ldots, B_t) &= \sum_{i_1,\ldots,i_t=1}^{n} R(E_{i_1,i_1}, \ldots, E_{i_t,i_t}) B_1(i_1) \cdots B_t(i_t) \\ &= \sum_{i_1,\ldots,i_t=1}^{n} T_{i_1,\ldots,i_t} B_1(i_1) \cdots B_t(i_t), \end{aligned}$$

which shows the claim.

Next, we show that (6) is equivalent to (7). The fact that (7) implies (6) follows immediately from the fact that the map $\operatorname{Diag}(x) \mapsto \operatorname{Diag}(x) \otimes \mathrm{Id}_d$ is a $*$-representation. Now assume (6). Without loss of generality, we may assume that each of the Hilbert spaces $\mathcal{H}_1, \ldots, \mathcal{H}_t$ has dimension at least m. If not, we can expand the dimensions of the ranges and domains of the representations π_i and contractions V_i by dilating with appropriate isometries into larger Hilbert spaces ("padding with zeros"). For each $i \in [t]$, let $S_i \subseteq \mathcal{H}_i$ be the subspace

$$S_i = \operatorname{Span}\big\{\pi_i(x_i) V_{i+1} \cdots V_t \pi_t(x_t) V_{t+1} : x_i, \ldots, x_t \in \mathcal{X}\big\}.$$

Since $\dim(\mathcal{X}) = n$, we have that $\dim(S_i) \leq m$. For each $i \in [t]$, let $Q_i \in L(\mathbb{C}^m, \mathcal{H}_i)$ be an isometry such that $S_i \subseteq \operatorname{Im}(Q_i)$. Note that V_{i+1} is a vector in the unit ball of $\mathcal{H}_t$. Let $Q_{t+1} \in L(\mathbb{C}^m, \mathcal{H}_t)$ be an isometry such that $V_{t+1} \in \operatorname{Im}(Q_{t+1})$. Note that for each $i \in [t+1]$, the map $Q_i Q_i^*$ acts as the identity on $\operatorname{Im}(Q_i)$. For each $i \in \{2, \ldots, t\}$ define the map $\sigma_i : \mathcal{X} \to L(\mathbb{C}^m)$ by $\sigma_i(x) = Q_i^* V_i \pi_i(x) Q_{i+1}$ and $\sigma_1(x) = Q_1^* \pi_1(x) Q_2$. Finally define $u = Q_1^* V_1^*$ and $v = Q_{t+1}^* V_{t+1}$. Then, the right-hand side of (6) can be written as

$$u^* \sigma_1(x_1) \cdots \sigma_t(x_t) v.$$

It follows from Lemma 8 that $\|\sigma_i\|_{\mathrm{cb}} \leq 1$. Let $\sigma_i' : L(\mathbb{C}^n) :\to L(\mathbb{C}^m)$ be the linear map given by $\sigma_i'(M) = \sigma_i(\mathrm{Diag}(M_{11}, \ldots, M_{nn}))$ for any $M \in L(\mathbb{C}^m)$. Then, for any diagonal matrix $x \in \mathcal{X}$, we have $\sigma_i(x) = \sigma_i'(x)$ and $\|\sigma_i'\|_{\mathrm{cb}} = \|\sigma_i\|_{\mathrm{cb}}$. It follows from Theorem 9 that there exists a positive integer d_i and contractions $U_i, V_i : L(\mathbb{C}^m, \mathbb{C}^{dn})$ such that $\sigma_i(x) = U_i^*(x \otimes \mathrm{Id}_{d_i})V_i$ for any $x \in \mathcal{X}$. We can take all d_i equal to $d = \max_i\{d_i\}$ by suitably dilating the contractions U_i, V_i. Setting $u' = u/\|u\|_{\ell_2}$ and $U_1' = \|u\|_{\ell_2}U_1$, and similarly defining v', V_{i+1}' gives the remaining implication. ◀

Corollary 7 implies the following lemma, from which Theorem 3 easily follows.

▶ **Lemma 10.** *Let $\beta : \{-1,1\}^n \to [-1,1]$ be some map and let t be a positive integer. Then, the following are equivalent.*

1. *There exists a $(2t)$-tensor $T \in \mathbb{R}^{2n\times\cdots\times 2n}$ such that $\|T\|_{\mathrm{cb}} \leq 1$ and for every $x \in \{-1,1\}^n$ and $y = (x, \mathbf{1})$, we have*

$$\sum_{i_1,\ldots,i_{2t}=1}^{2n} T_{i_1,\ldots,i_{2t}} y_{i_1} \cdots y_{i_{2t}} = \beta(x).$$

2. *There exists a t-query quantum algorithm that, on input $x \in \{-1,1\}^n$, returns a random sign with expected value $\beta(x)$.*

Proof. We first prove that (2) implies (1). As discussed in Section 2, a t-query quantum algorithm with phase oracles initializes the joint register $(\mathsf{A}, \mathsf{Q}, \mathsf{W})$ in the all-zero state on which it then performs a sequence of unitaries $U_1, \ldots, U_t$ interlaced with queries $D(x) = \mathrm{Diag}((\mathbf{1}, x)) \otimes \mathrm{Id}_{\mathsf{W}}$. Let $\{P_0, P_1\}$ be the the two-outcome measurement done at the end of the algorithm and assume that it returns $+1$ on measurement outcome zero and -1 otherwise. Let $Q = P_0 - P_1$ and note that Q is a contraction since P_0, P_1 are positive semi-definite and satisfy $P_0 + P_1 = \mathrm{Id}$. The expected value of the measurement outcome is then given by

$$e_0^* U_1^* D(x) U_2^* \cdots D(x) U_t^* Q U_t D(x) \cdots U_2 D(x) U_1 e_0. \tag{8}$$

By assumption, this expected value equals $\beta(x)$ for every $x \in \{-1,1\}^n$. For $z \in \mathbb{C}^{2n}$, denote $D'(z) = \mathrm{Diag}((z_{n+1}, \ldots, z_{2n}, z_1, \ldots, z_n)) \otimes \mathrm{Id}_{\mathsf{W}}$ and $\widetilde{U}_t = U_t^* Q U_t$. Define the $(2t)$-linear form T by

$$T(y_1, \ldots, y_{2t}) = u^* U_1^* D'(y_1) U_2^* \cdots D'(y_t) \widetilde{U}_t D'(y_{t+1}) \cdots U_2 D'(y_{2t}) U_1 u.$$

Clearly $T((x,\mathbf{1}) \ldots, (x,\mathbf{1})) = \beta(x)$ for every $x \in \{-1,1\}^n$. Moreover, by definition T admits a factorization as in (7). It thus follows from Corollary 7 that $\|T\|_{\mathrm{cb}} \leq 1$. We turn T into a real tensor by taking its real part $T' = (T + \overline{T})/2$, where $\overline{T}$ is the coordinate-wise complex conjugate of T. Since for any $x \in \{-1,1\}^n$ and $y = (x, \mathbf{1})$, the value $T(y, \ldots, y)$ is real, we have $T'(y, \ldots, y) = \beta(x)$. We need to show that $\|T'\|_{\mathrm{cb}} \leq 1$. To this end, consider an arbitrary positive integer d, unit vectors $v, w \in \mathbb{C}^d$ and sequences of unitary matrices $V_1(i), \ldots, V_{2t}(i)$ for $i \in [n]$ such that

$$\Big\| \sum_{i_1,\ldots,i_{2t}=1}^{2n} \overline{T_{i_1,\ldots,i_{2t}}} V_1(i_1) \cdots V_{2t}(i_{2t}) \Big\| = \Big| \sum_{i_1,\ldots,i_{2t}=1}^{2n} \overline{T_{i_1,\ldots,i_{2t}}} v^* V_1(i_1) \cdots V_{2t}(i_{2t})\, w \Big|.$$

Note that $\|\overline{T}\|_{\mathrm{cb}}$ is given by the supremum over d and $V_j(i)$. Taking the complex conjugate of the above summands on the right-hand side allows us to express the above absolute value as

$$\Big| \sum_{i_1,\ldots,i_{2t}=1}^{2n} T_{i_1,\ldots,i_{2t}} \bar{v}^* \overline{V_1(i_1)} \cdots \overline{V_{2t}(i_{2t})}\, \bar{w} \Big|, \tag{9}$$

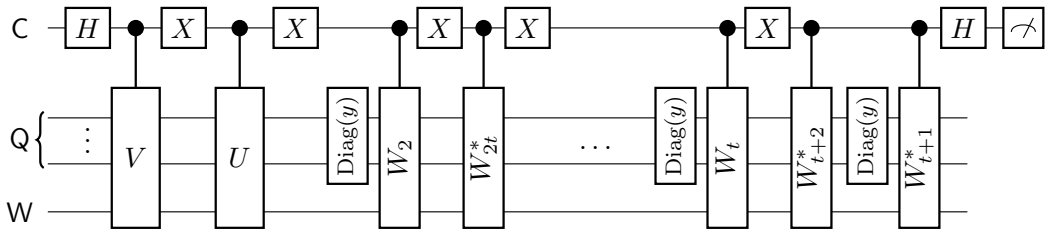


Figure 2 The registers $\mathsf{C}, \mathsf{Q}, \mathsf{W}$ denote the control, query and workspace registers. Let U, V be unitaries with $W_1\widetilde{u}$ and $W_{2t+1}\widetilde{v}$ as their first rows, respectively and for $x \in \{-1,1\}^n$ and $y = (x, \mathbf{1})$, let $\mathrm{Diag}(y)$ be the query operator. The algorithm begins by initializing the joint register $(\mathsf{C}, \mathsf{Q}, \mathsf{W})$ in the all-zero state and proceeds by performing the displayed operations. The algorithm returns $+1$ if the outcome of the measurement on C equals zero and -1 otherwise.

where $\bar{v}, \bar{w}, \overline{V_j(i)}$ denote the coordinate-wise complex conjugates. Since each $\overline{V_j(i)}$ is still unitary, it follows that (9) is at most $\|T\|_{\mathrm{cb}}$ and so $\|\overline{T}\|_{\mathrm{cb}} \leq \|T\|_{\mathrm{cb}} \leq 1$. Hence, by the triangle inequality, $\|T'\|_{\mathrm{cb}} \leq (\|T\|_{\mathrm{cb}} + \|\overline{T}\|_{\mathrm{cb}})/2 \leq 1$ as desired.

Next, we show that (1) implies (2). Let T be a $(2t)$-tensor as in item 1. Since any matrix with operator norm at most 1 is a convex combination of unitary matrices (by the Russo-Dye Theorem), it follows from Corollary 7 that T admits a factorization as in (7). Let $V_0, U_{2t+1} \in L(\mathbb{C}^m, \mathbb{C}^{2dn})$ be isometries. For each $i \in [2t+1]$, define the map $W_i \in L(\mathbb{C}^{2dn})$ by $W_i = V_{i-1}U_i^*$. Observe that each W_i is a contraction and recall that unitaries are contractions. For the moment, assume for simplicity that each W_i is in fact unitary. Define two vectors $\widetilde{u} = V_0 u$ and $\widetilde{v} = U_{2t+1}v$ and observe that these are unit vectors in $\mathbb{C}^{2dn}$. The right-hand side of (7) then gives us

$$T(y_1, \ldots, y_{2t}) = \widetilde{u}^* W_1 \widetilde{D}(y_1) W_2 \widetilde{D}(y_2) W_3 \cdots W_{2t} \widetilde{D}(y_{2t}) W_{2t+1} \widetilde{v}, \tag{10}$$

where $\widetilde{D}(y_i) = \mathrm{Diag}(y_i) \otimes \mathrm{Id}_d$ for $i \in [2t]$. Based on this, we obtain the quantum query algorithm described in Figure 2.

To see why this algorithm satisfies the requirements, first note that the algorithm makes t queries to the input x. For the correctness of the algorithm, we begin by observing that before the application of the first query, the state of the joint register $(\mathsf{C}, \mathsf{Q}, \mathsf{W})$ is

$$\tfrac{1}{\sqrt{2}}(e_0 \otimes W_1\widetilde{u} + e_1 \otimes W_{t+1}\widetilde{v}).$$

Before the final Hadamard gate, the state of the joint register is given by

$$\begin{aligned} &\frac{1}{\sqrt{2}} e_0 \otimes \big((\mathrm{Diag}(y) \otimes \mathrm{Id}_d) W_t \cdots W_2 (\mathrm{Diag}(y) \otimes \mathrm{Id}_d) W_1 \widetilde{u}\big) \\ +&\frac{1}{\sqrt{2}} e_1 \otimes \big(W_{t+1}^* (\mathrm{Diag}(y) \otimes \mathrm{Id}_d) W_{t+2}^* \cdots W_{2t}^* (\mathrm{Diag}(y) \otimes \mathrm{Id}_d) W_{2t+1}^* \widetilde{v}\big). \end{aligned}$$

A standard calculation and (10) then show that after the final Hadamard gate, the expected output of the algorithm is precisely $T((x, \mathbf{1}), \ldots, (x, \mathbf{1})) = \beta(x)$. In the general case where the W_is are not necessarily unitary, we can use the fact that, by the Russo–Dye Theorem and Carathéodory's Theorem, each W_i is a convex combination of at most $(dn)^2 + 1$ unitaries. The algorithm can thus use randomness to effect each W_i on expectation. Alternatively, by linear algebra there exists a unitary matrix $W_i' \in \mathbb{C}^{2dn \times 2dn}$ that has W_i as its upper-left corner (see [2, Lemma 7]), through which the algorithm could implement W_i by working on a larger quantum register. ◀

Using Lemma 10, we now prove our main Theorem 3.

Proof of Theorem 3. We first show that (2) implies (1). Using the equivalence in Lemma 10, it follows that there exists a $(2t)$-tensor $T \in \mathbb{R}^{2n\times\cdots\times 2n}$ such that $\|T\|_{\mathrm{cb}} \leq 1$ and for every $x \in \{-1,1\}^n$ and $y = (x, \mathbf{1})$, we have

$$\sum_{i_1,\ldots,i_{2t}=1}^{2n} T_{i_1,\ldots,i_{2t}} y_{i_1} \cdots y_{i_{2t}} = \beta(x).$$

Define the symmetric $2t$-tensor $T_p = \frac{1}{(2t)!}\sum_{\sigma\in S_{2t}} T\circ\sigma$. Let $p \in \mathbb{R}[x_1,\ldots,x_{2n}]$ be the form of degree $2t$ associated with T_p by (2) (note that there is a unique polynomial associated with the symmetric tensor T_p). Then, $p((x,\mathbf{1})) = \beta(x)$ for every $x \in \{-1,1\}^n$. Moreover, if we set $T^\sigma = T$ for each $\sigma \in S_t$, it follows from the above decomposition of T_p and Definition 2 that $\|p\|_{\mathrm{cb}} \leq \|T\|_{\mathrm{cb}} \leq 1$.

Next, we show that (1) implies (2). Let p be a degree-$(2t)$ form satisfying $\|p\|_{\mathrm{cb}} \leq 1$. Suppose T_p as defined in Eq. (2) can be written as $T_p = \sum_{\sigma\in S_{2t}} T^\sigma \circ \sigma$ and $\sum_{\sigma\in S_{2t}} \|T^\sigma\|_{\mathrm{cb}} = \|p\|_{\mathrm{cb}} \leq 1$. Define $T = \sum_{\sigma\in S_{2t}} T^\sigma$. Then, using the triangle inequality, it follows that $\|T\|_{\mathrm{cb}} \leq \sum_{\sigma\in S_{2t}} \|T^\sigma\|_{\mathrm{cb}} \leq 1$. Also note that for any $y \in \mathbb{R}^{2n}$,

$$T(y,\ldots,y) = \sum_{\sigma\in S_{2t}} T^\sigma(y,\ldots,y) = \sum_{\sigma\in S_{2t}} (T^\sigma\circ\sigma)(y,\ldots,y) = T_p(y,\ldots,y) = p(y).$$

Using Lemma 10 (in particular (1) $\implies$ (2)) for the tensor T, the theorem follows. ◀

We now prove Corollary 5, which is an immediate consequence of our main theorem.

Proof of Corollary 5. We first show $\text{cb-deg}_\varepsilon(f) \geq Q_\varepsilon(f)$: Suppose $\text{cb-deg}_\varepsilon(f) = d$, then there exists a degree-$(2d)$ form p satisfying: $|p(x) - f(x)| \leq 2\varepsilon$ for every $x \in D$ and $\|p\|_{\mathrm{cb}} \leq 1$. Using our characterization in Theorem 3, it follows that there exists a d-query quantum algorithm $\mathcal{A}$, that on input $x \in D$, returns a random sign with expected value $p(x)$. So, our ε-error quantum algorithm for f simply runs $\mathcal{A}$ and outputs the random sign.

We next show $\text{cb-deg}_\varepsilon(f) \leq Q_\varepsilon(f)$. Suppose $Q_\varepsilon(f) = t$. There exists a t-query quantum algorithm that, on input $x \in D$, outputs a random sign with expected value $\beta(x)$ satisfying $|\beta(x) - f(x)| \leq 2\varepsilon$. Note that we could also run the quantum algorithm for $x \notin D$ and let $\beta(x)$ be the expected value of the quantum algorithm for such xs. Using Theorem 3, we know that there exists a degree-$(2t)$ form p satisfying $\beta(x) = p(x)$ for every $x \in \{-1,1\}^n$ and $\|p\|_{\mathrm{cb}} \leq 1$. Clearly p satisfies satisfies the conditions of Definition 4, hence $\text{cb-deg}_\varepsilon(f) \leq t$. ◀

4 Separations for quartic polynomials

In this section we show the existence of a quartic polynomial p that is bounded but for which any two-query quantum algorithm $\mathcal{A}$ satisfying $\mathbf{E}[\mathcal{A}(x)] = Cp(x)$ for every $x \in \{-1,1\}^n$ must have $C = O(n^{-1/2})$. We show this using a (random) *cubic* form that is bounded, but whose completely bounded norm is poly(n), following a construction of [22, Theorem 18.16].

Given a form $p : \mathbb{R}^n \to \mathbb{R}$, we define its norm as

$$\|p\| = \sup\{|p(x)| : x \in \{-1,1\}^n\}.$$

Note that the condition $\|p\| \leq 1$ is equivalent to p being bounded.

▶ **Theorem 11.** *There exist absolute constants $C, c \in (0,\infty)$ such that the following holds. Let*

$$p(x) = \sum_{\alpha\in\{0,1,2,3\}^n:\,|\alpha|=3} c_\alpha x^\alpha$$

be the random cubic form such the coefficients c_α are independent uniformly distributed $\{-1,1\}$-valued random variables. Then, with probability at least $1 - Cne^{-cn}$, we have $\|p\|_{\mathrm{cb}} \geq c\sqrt{n}\|p\|$.

We shall use the following standard concentration-of-measure results. The first is the Hoeffding bound [44, Corollary 3 (Appendix B)].

▶ **Lemma 12** (Hoeffding bound)**.** *Let $X_1,\ldots,X_m$ be independent uniformly distributed $\{-1,1\}$-random variables and let $a \in \mathbb{R}^m$. Then, for any $\tau > 0$, we have*

$$\Pr\Big[\Big|\sum_{i=1}^m a_i X_i\Big| > \tau\Big] \leq 2e^{-\frac{\tau^2}{2(a_1^2+\cdots+a_m^2)}}$$

The second result is one from random matrix theory concerning upper tail estimates for Wigner ensembles (see [51, Corollary 2.3.6]).

▶ **Lemma 13.** *There exist absolute constants $C, c \in (0,\infty)$ such that the following holds. Let n be a positive integer and let M be a random $n \times n$ symmetric random matrix such that for $j \geq i$, the entries M_{ij} are independent random variables with mean zero and absolute value at most 1. Then, for any $\tau \geq C$, we have*

$$\Pr\big[\|M\| > \tau\sqrt{n}\big] \leq Ce^{-c\tau n}.$$

We also use the following proposition.

▶ **Proposition 14.** *Let m, n, t be positive integers, let $p \in \mathbb{R}[x_1,\ldots,x_n]$ be a t-linear form, let $T_p \in \mathbb{R}^{n\times\cdots\times n}$ be as in (2) and $A_1,\ldots,A_n \in L(\mathbb{R}^m)$ be pairwise commuting contractions. Then,*

$$\|p\|_{\mathrm{cb}} \geq \Big\|\sum_{i_1,\ldots,i_t=1}^n (T_p)_{i_1,\ldots,i_t} A_{i_1}\cdots A_{i_t}\Big\|.$$

Proof. Consider an arbitrary decomposition $T_p = \sum_{\sigma\in S_t} T^\sigma \circ \sigma$. Then, the definition of the completely bounded norm and triangle inequality show that

$$\sum_{\sigma\in S_t}\|T^\sigma\|_{\mathrm{cb}} \;\geq\; \sum_{\sigma\in S_t}\Big\|\sum_{i_1,\ldots,i_t=1}^n T^\sigma_{i_1,\ldots,i_t} A_{i_1}\cdots A_{i_t}\Big\| \;\geq\; \Big\|\sum_{\sigma\in S_t}\sum_{i_1,\ldots,i_t=1}^n T^\sigma_{i_1,\ldots,i_t} A_{i_1}\cdots A_{i_t}\Big\|.$$

Since the A_i commute, the above reduces to

$$\begin{aligned}\Big\|\sum_{\sigma\in S_t}\sum_{i_1,\ldots,i_t=1}^n T^\sigma_{i_1,\ldots,i_t} A_{\sigma^{-1}(i_1)}\cdots A_{\sigma^{-1}(i_t)}\Big\| &= \Big\|\sum_{\sigma\in S_t}\sum_{i_1,\ldots,i_t=1}^n (T^\sigma\circ\sigma)_{i_1,\ldots,i_t} A_{i_1}\cdots A_{i_t}\Big\| \\ &= \Big\|\sum_{i_1,\ldots,i_t=1}^n (T_p)_{i_1,\ldots,i_t} A_{i_1}\cdots A_{i_t}\Big\|.\end{aligned}$$

The claim now follows from the definition of $\|p\|_{\mathrm{cb}}$ and since the decomposition of T_p was arbitrary. ◀

Proof of Theorem 11. We begin by showing that with high probability, $\|p\| \leq O(n^2)$. To this end, let us fix an arbitrary $x \in \{-1,1\}^n$. Then, $p(x)$ is a sum of at most n^3 independent uniformly distributed random $\{-1,1\}$-random variables. It follows from Lemma 12 that

$$\Pr\big[|p(x)| > 2n^2\big] \leq 2e^{-2n},$$

By the union bound over $x \in \{-1,1\}^n$, it follows that $\|p\| > 2n^2$ with probability at most $2e^{-n}$, which gives the claim.

We now lower bound $\|p\|_{\mathrm{cb}}$. Let $\tau > 0$ be a parameter to be set later. Let $T \in \mathbb{R}^{n\times n\times n}$ be the random symmetric 3-tensor associated with p as in (2). For every $i \in [n]$, we define the linear map $A_i : \mathbb{R}^{2n+2} \to \mathbb{R}^{2n+2}$ by

$$\begin{cases} A_i e_0 = e_i \\ A_i e_j = \frac{1}{\tau\sqrt{n}} \sum_{k=1}^n T_{i,j,k} e_{k+n} \\ A_i e_{j+n} = \delta_{i,j} e_{2n+1} \\ A_i e_{2n+1} = 0. \end{cases}$$

Observe that for every $i,j,k \in [n]$, we have

$$e_{2n+1}^* A_i A_j A_k e_0 = \frac{1}{\tau\sqrt{n}} T_{i,j,k}. \tag{11}$$

Since T is symmetric, it follows easily that these maps commute, which is to say that $A_iA_j = A_jA_i$ for every $i,j \in [n]$. In addition, we claim that with high probability, these maps are contractions (i.e., the associated matrices have operator norm at most 1). To see this, for each $i \in [n]$, let M_i be the random matrix given by $M_i = (T_{i,j,k})_{j,k=1}^n$. Observe that M_i is symmetric and its entries have mean zero and absolute value at most 1. By Lemma 13 and a union bound, we get that

$$\Pr\Big[\max_{i\in[n]} \big\|M_i\big\| > \tau\sqrt{n}\Big] \leq Cne^{-c\tau n}. \tag{12}$$

for absolute constants c, C and provided $\tau \geq C$. Now, for any Euclidean unit vector $u \in \mathbb{R}^{2n+2}$, we have

$$\|A_i u\|^2 = |u_0|^2 + \frac{1}{\tau^2 n} \sum_{k=1}^n \Big|\sum_{j=1}^n u_j T_{i,j,k}\Big|^2 + |u_{i+n}|^2 \leq |u_0|^2 + \frac{\|M_i\|^2}{\tau^2 n} \sum_{j=1}^n |u_j|^2 + |u_{i+n}|^2.$$

It follows from (12) that $\max_i \|M_i\| \leq \tau\sqrt{n}$ with probability at least $1 - Cne^{-c\tau n}$, which in turn implies the above is at most $\|u\|^2 \leq 1$ and therefore that all A_i have operator norm ≤ 1.

By Proposition 14,

$$\|p\|_{\mathrm{cb}} \geq \Big\| \sum_{i,j,k=1}^n T_{i,j,k} A_i A_j A_k \Big\|,$$

provided that the A_is are contractions. By (11), and since $|T_{i,j,k}| \geq 1/6$ for every $i,j,k \in [n]$, the above is at least $n^{5/2}/(36\tau)$. with probability at least $1 - Cne^{-c\tau n}$. Letting τ be a sufficiently large constant then gives the result. ◀

To demonstrate the failure of Theorem 1 for quartic polynomials, we embed As mentioned in the introduction, one can easily extend this result to the case of 4-linear forms.

▶ **Corollary 15.** *There exists a bounded quartic form*

$$q(x_1,\ldots,x_n) = \sum_{\alpha\in\{0,1\}^n:\,|\alpha|=4} d_\alpha x^\alpha, \tag{13}$$

and pairwise commuting contractions $A_1,\ldots,A_n \in L(\mathbb{R}^{2n+2})$ such that

$$\Big\| \sum_{i,j,k,\ell=1}^{n} (T_q)_{i,j,k,\ell} A_i A_j A_k A_\ell \Big\| \geq c\sqrt{n}$$

where $c \in (0,1]$ is some absolute constant.

Proof. Let p be a bounded multi-linear cubic form such that $\|p\|_{\mathrm{cb}} \geq C\sqrt{n}$, the existence of which is guaranteed by Theorem 11. Let $T_p \in \mathbb{R}^{n\times n\times n}$ be the random symmetric 3-tensor associated to p. Consider the symmetric 4-tensor $S \in \mathbb{R}^{(n+1)\times(n+1)\times(n+1)\times(n+1)}$ defined by $S_{0,j,k,\ell} = T_{j,k,\ell}$, $S_{i,0,k,\ell} = T_{i,k,\ell}$, $S_{i,j,0,\ell} = T_{i,j,\ell}$, $S_{i,j,k,0} = T_{i,j,k}$ for every $i,j,k,\ell \in [n]$ and $S_{i,j,k,\ell} = 0$ otherwise. Since S is symmetric, there exists a unique multi-linear quartic form q associated to S. It follows easily that $\|q\| = 4\|p\|$. Moreover, by considering the contractions A_i used in the proof of Theorem 11 and defining $A_0 = \mathrm{Id}_{n+2}$, it follows that $\|q\|_{\mathrm{cb}} \geq 4\|p\|_{\mathrm{cb}}$. The form $q/4$ is thus as desired. ◀

We claim that a form q as in Corollary (15) gives a counterexample to possible quartic extensions of Theorem 1. To see this, suppose there exists a two-query quantum algorithm $\mathcal{A}$ and a $C \in (0,\infty)$ such that $\mathbf{E}[\mathcal{A}(x)] = Cq(x)$ for each $x \in \{-1,1\}^n$. By Theorem 3 that there exists a $(2n)$-variate quartic form h such that $h(x,\mathbf{1}) = Cq(x)$ for each $x \in \{-1,1\}^n$ and $\|h\|_{\mathrm{cb}} \leq 1$. We now show that the degree-4 coefficients in $h(x,y)$ are completely determined by $q(x)$. Indeed, if we expand

$$h(x,y) = \sum_{\alpha,\beta\in\{0,1,2,3,4\}^n:\,|\alpha|+|\beta|=4} d'_{\alpha,\beta} x^\alpha y^\beta,$$

then

$$h(x,\mathbf{1}) = \sum_{\alpha,\beta\in\{0,1,2,3,4\}^n:\,|\alpha|+|\beta|=4} d'_{\alpha,\beta} x^\alpha = C \sum_{\alpha\in\{0,1\}^n:\,|\alpha|=4} d_\alpha x^\alpha = Cq(x). \tag{14}$$

It follows from the above that $d'_{\alpha,\mathbf{0}} = Cd_\alpha$ for all $\alpha \in \{0,1\}^n$ such that $|\alpha| = 4$.

In order to lower bound $\|h\|_{\mathrm{cb}}$, let $T_h \in \mathbb{R}^{(2n)\times(2n)\times(2n)\times(2n)}$ be the symmetric 4-tensor associated to h. By Proposition (14), we have

$$\|h\|_{\mathrm{cb}} \geq \Big\| \sum_{i,j,k,\ell=1}^{2n} (T_h)_{i,j,k,\ell} B_i B_j B_k B_\ell \Big\|,$$

for every set of pairwise commuting contractions $B_1,\ldots,B_{2n}$. In particular, set $B_i = A_i$ as in Corollary (15) for $i \in [n]$ and let B_i be the all-zero matrix for $i \in \{n+1,\ldots,2n\}$. Since the A_is were pairwise commuting in Corollary (15) (which ofcourse commute with the all-zero matrix), the B_is are pairwise commuting. Finally, observe that for all $i,j,k,\ell \in [n]$, we have $(T_h)_{i,j,k,\ell} = d'_{\alpha,\mathbf{0}}/(|\{i,j,k,\ell\}|!)$, which is equal to $Cd_\alpha/(|\{i,j,k,\ell\}|!)$ (by Eq. (14)). In particular, using Corollary (15), we have

$$\|h\|_{\mathrm{cb}} \geq \Big\| \sum_{i,j,k,\ell=1}^{2n} (T_h)_{i,j,k,\ell} B_i B_j B_k B_\ell \Big\| = C \Big\| \sum_{i,j,k,\ell=1}^{n} (T_q)_{i,j,k,\ell} A_i A_j A_k A_\ell \Big\| \geq Cc\sqrt{n}.$$

This implies that $1 \geq \|h\|_{\mathrm{cb}} = C\|q\|_{\mathrm{cb}} \geq Cc\sqrt{n}$, and so $C \leq 1/(c\sqrt{n})$.

5 Short proof of Theorem 1.

In this section, we give a short proof of Theorem 1, restated below for convenience.

▶ **Theorem 1** (Aaronson et al.). *There exists an absolute constant $C \in (0,1]$ such that the following holds. For any bounded quadratic polynomial p, there exists a one-query quantum algorithm that, on input $x \in \{-1,1\}^n$, returns a random sign with expectation $Cp(x)$.*

Proof sketch of Theorem 1. The first step is to show that without loss of generality, we may assume that the polynomial p is a quadratic form. This is the content of the decoupling argument mentioned in the introduction, proved for polynomials of arbitrary degree in [2], but stated here only for the quadratic case.

▶ **Lemma 16.** *There exists an absolute constant $C \in (0,1]$ such that the following holds. For any bounded quadratic polynomial p, there exists a matrix $A \in \mathbb{R}^{(n+1)\times(n+1)}$ with $\|A\|_{\ell_\infty \to \ell_1} \leq 1$, such that the quadratic form $q(y) = y^{\mathsf{T}} A y$ satisfies $q((x,1)) = Cp(x)$ for all $x \in \{-1,1\}^n$.*

To prove the theorem, we may thus restrict to a quadratic form $p(x) = x^{\mathsf{T}} A x$ given by some matrix $A \in \mathbb{R}^{n\times n}$ such that $\|A\|_{\ell_\infty \to \ell_1} \leq 1$. The next step is to massage the matrix A into a unitary matrix (that can be applied by a quantum algorithm). To do so, the authors use an argument based on two versions of Grothendieck's inequality and a technique known as *variable splitting*, developed in earlier work of Aaronson and Ambainis [1]. The first version of Grothendieck's inequality is the one most commonly used in applications [25].

▶ **Theorem 17** (Grothendieck). *There exists a universal constant $K_G \in (0,\infty)$ such that the following holds. For every positive integer n and matrix $A \in \mathbb{R}^{n\times n}$, we have*

$$\sup\left\{ \sum_{i,j=1}^{n} A_{ij}\langle u_i, v_j\rangle : d \in \mathbb{N},\ u_i, v_j \in B_2^n \right\} \leq K_G \|A\|_{\ell_\infty \to \ell_1}.$$

Elementary proofs of this theorem can be found for instance in [5]. The *Grothendieck constant* K_G is the smallest real number for which Theorem 17 holds true. The problem of determining its exact value, posed in [25], remains open. The best lower and upper bounds $1.6769\cdots \leq K_G < 1.7822\cdots$ were proved by Davie and Reeds [21, 45], and Braverman et al. [14], resp. The second version of Grothendieck's inequality is as follows.

▶ **Theorem 18** (Grothendieck). *For every positive integer n and matrix $A \in \mathbb{R}^{n\times n}$, there exist $u, v \in (0,1]^n$ such that $\|u\|_{\ell_2} = \|v\|_{\ell_2} = 1$ and such that the matrix*

$$B = \frac{1}{K_G} \operatorname{Diag}(u)^{-1} A \operatorname{Diag}(v)^{-1} \tag{15}$$

satisfies $\|B\| \leq \|A\|_{\ell_\infty \to \ell_1}$.

Our contribution. The first (standard) version of Grothendieck's inequality (Theorem 17) easily implies that any matrix A with $\|A\|_{\ell_\infty \to \ell_1} \leq 1$ has completely bounded norm at most K_G. Combing this with our Theorem 3 and Lemma 16, one quickly retrieves Theorem 1. However, Theorem 3 is based on the rather deep Theorem 6. We observe that Theorem 1 also follows readily from the much simpler Theorem 18 alone (proved below for completeness), after one assumes that p is a quadratic form as above. Indeed, Theorem 18 gives unit vectors u, v such that the matrix B as in (15) has (operator) norm at most 1. Unitary

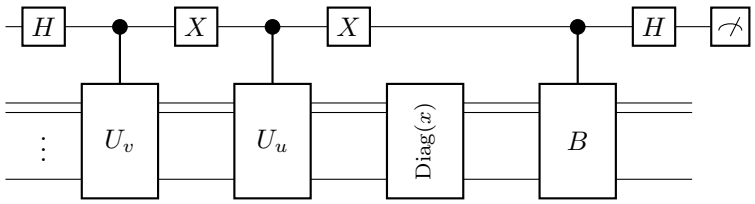


Figure 3 Let U_u, U_v be unitaries that have u, v as their first rows, respectively. The algorithm initializes a $(1+\log n)$-qubit register in the all-zero state, transforms this state into the superposition $\frac{1}{\sqrt{2}}(e_0 \otimes u + e_1 \otimes v)$, queries the input x via the unitary $\mathrm{Diag}(x)$ applied to the $(\log n)$-qubit register, applies a controlled-B, and finishes by measuring the first qubit in the Hadamard basis.

matrices have norm exactly 1 and represent the type of operation a quantum algorithm can implement. Moreover, since u, v are unit vectors, they represent $(\log n)$-qubit quantum states. Using the fact that for $w, z \in \mathbb{R}^n$, we have $\mathrm{Diag}(w)z = \mathrm{Diag}(z)w$, we get the following *factorization* formula (not unlike the one of Corollary 7, which is of course no coincidence):

$$\frac{x^{\mathsf{T}}Ax}{K_G} = x^{\mathsf{T}}\,\mathrm{Diag}(u)B\,\mathrm{Diag}(v)x = u^T\,\mathrm{Diag}(x)B\,\mathrm{Diag}(x)v. \tag{16}$$

If we assume for the moment that the matrix B actually is unitary, then the right-hand side of (16) suggests the simple one-query quantum algorithm described in Figure 3.

Using (16), we observe that the algorithm returns zero with probability

$$\frac{1}{2} + \frac{1}{2}\big\langle \mathrm{Diag}(u)x, B\,\mathrm{Diag}(v)x\big\rangle = \frac{1}{2} + \frac{x^{\mathsf{T}}Ax}{2K_G},$$

Now, it is clear that the the expected value of the measurement result is precisely $p(x)/K_G$, giving Theorem 1 with $C = 1/K_G$. In case B is not unitary, one can use the same argument used in the final step of the proof of Theorem 3.

References

1 S. Aaronson and A. Ambainis. Forrelation: A problem that optimally separates quantum from classical computing. In *Proceedings of 47th ACM STOC*, pages 307–316, 2015. arXiv:1411.5729v1.

2 S. Aaronson, A. Ambainis, J. Iraids, M. Kokainis, and J. Smotrovs. Polynomials, quantum query complexity, and Grothendieck's inequality. In *31st Conference on Computational Complexity, CCC 2016*, pages 25:1–25:19, 2016. arXiv:1511.08682.

3 S. Aaronson, S. Ben-David, and R. Kothari. Separations in query complexity using cheat sheets. In *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing, STOC*, pages 863–876, 2016. arXiv:1511.01937.

4 S. Aaronson and Y. Shi. Quantum lower bounds for the collision and the element distinctness problems. *J. ACM*, 51(4):595–605, 2004.

5 N. Alon and A. Naor. Approximating the cut-norm via Grothendieck's inequality. *SIAM Journal of Computing*, 35(4):787–803, 2006. Earlier version in STOC'04.

6 A. Ambainis. Quantum lower bounds by quantum arguments. *J. Comput. Syst. Sci.*, 64(4):750–767, 2002. Earlier version in STOC'00. arXiv:quant-ph/0002066.

7 A. Ambainis. Polynomial degree vs. quantum query complexity. *J. Comput. System Sci.*, 72(2):220–238, 2006. Earlier version in FOCS'03. quant-ph/0305028.

8 A. Ambainis. Quantum walk algorithm for element distinctness. *SIAM J. Comput.*, 37(1):210–239, 2007. Earlier version in FOCS'04. arXiv:quant-ph/0311001.

9 A. Ambainis, K. Balodis, A. Belovs, T. Lee, M. Santha, and J. Smotrovs. Separations in query complexity based on pointer functions. In *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2016*, pages 800–813, 2016. arXiv:1506.04719.

10 W. Arveson. *An invitation to C^*-algebras*, volume 39 of *Graduate Texts in Mathematics*. Springer, 2012.

11 Robert Beals, Harry Buhrman, Richard Cleve, Michele Mosca, and Ronald de Wolf. Quantum lower bounds by polynomials. *J. ACM*, 48(4):778–797, 2001. Earlier version in FOCS'98. quant-ph/9802049. doi:10.1145/502090.502097.

12 A. Belovs. Span programs for functions with constant-sized 1-certificates. In *Proceedings of the 44th Symposium on Theory of Computing Conference, STOC 2012*, pages 77–84, 2012. arXiv:1105.4024.

13 C. H. Bennett, E. Bernstein, G. Brassard, and U. Vazirani. Strengths and weaknesses of quantum computing. *SIAM Journal of Computing*, 26(5):1510–1523, 1997. quant-ph/9701001.

14 M. Braverman, K. Makarychev, Y. Makarychev, and A. Naor. The Grothendieck constant is strictly smaller than Krivine's bound. *Forum Math. Pi*, 1:453–462, 2013. Preliminary version in FOCS'11. arXiv:1103.6161.

15 J. Briët, H. Buhrman, T. Lee, and T. Vidick. All Schatten spaces endowed with the Schur product are Q-algebras. *Journal of Functional Analysis*, 262(1):1–9, 2012.

16 J. Briët, H. Buhrman, T. Lee, and T. Vidick. Multipartite entanglement in XOR games. *Quantum Information & Computation*, 13(3-4):334–360, 2013. arXiv:0911.4007.

17 J. Briët and T. Vidick. Explicit lower and upper bounds on the entangled value of multiplayer XOR games. *Communications in Mathematical Physics*, 321(1):181–207, 2013. arXiv:1108.5647.

18 M. Bun, R. Kothari, and J. Thaler. The polynomial method strikes back: Tight quantum query bounds via dual polynomials. arXiv:1710.09079, 2017.

19 E. Christensen and A. M. Sinclair. Representations of completely bounded multilinear operators. *Journal of Functional analysis*, 72(1):151–181, 1987.

20 R. Cleve, P. Høyer, B. Toner, and J. Watrous. Consequences and limits of nonlocal strategies. In *Computational Complexity, 2004. Proceedings. 19th IEEE Annual Conference on*, pages 236–249. IEEE, 2004. arXiv:quant-ph/0404076.

21 A. Davie. Lower bound for K_G. Unpublished, 1984.

22 J. Diestel, H. Jarchow, and A. Tonge. *Absolutely summing operators*, volume 43 of *Cambridge Studies in Advanced Mathematics*. Cambridge University Press, Cambridge, 1995.

23 E. Farhi, J. Goldstone, and S. Gutmann. A quantum algorithm for the Hamiltonian NAND tree. *Theory of Computation*, 4(8):169–190, 2008. arXiv:quant-ph/0702144.

24 L. T. Gardner. An elementary proof of the Russo-Dye theorem. *Proceedings of the American Mathematical Society*, 90(1):171, 1984.

25 A. Grothendieck. Résumé de la théorie métrique des produits tensoriels topologiques (French). *Bol. Soc. Mat. São Paulo*, 8:1–79, 1953.

26 L. 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.

27 P. Høyer, T. Lee, and R. Špalek. Negative weights make adversaries stronger. In *Proceedings of the 39th Annual ACM Symposium on Theory of Computing, 2007*, pages 526–535, 2007. arXiv:quant-ph/0611054.

28 N. Johnston, D. W. Kribs, and V. I. Paulsen. Computing stabilized norms for quantum operations via the theory of completely bounded maps. *Quantum Information & Computation*, 9(1):16–35, 2009. arXiv:0711.3636.

29 J. Kaniewski, T. Lee, and R. de Wolf. Query complexity in expectation. In *Automata, Languages, and Programming - 42nd International Colloquium, ICALP*, pages 761–772, 2015. arXiv:1411.7280.

30 S. Khot and A. Naor. Grothendieck-type inequalities in combinatorial optimization. *Communications on Pure and Applied Mathematics*, 65(7):992–1035, 2012. arXiv:1108.2464.

31 C. M. Le, E. Levina, and R. Vershynin. Sparse random graphs: regularization and concentration of the Laplacian. arXiv:1502.03049, 2015.

32 T. Lee, R. Mittal, B. W. Reichardt, R. Špalek, and M. Szegedy. Quantum query complexity of state conversion. In *IEEE 52nd Annual Symposium on Foundations of Computer Science, FOCS 2011*, pages 344–353, 2011. arXiv:1011.3020.

33 F. Magniez, A. Nayak, J. Roland, and M. Santha. Search via quantum walk. *SIAM J. Comput.*, 40(1):142–164, 2011. Earlier version in STOC'07. arXiv:quant-ph/0608026.

34 A. Montanaro. Nonadaptive quantum query complexity. *Inf. Process. Lett.*, 110(24), 2010.

35 A. Montanaro, H. Nishimura, and R. Raymond. Unbounded-error quantum query complexity. *Theor. Comput. Sci.*, 412(35):4619–4628, 2011.

36 Gerard J. Murphy. *C^*-algebras and operator theory*. Academic Press, Inc., Boston, MA, 1990.

37 T. Oikhberg and G. Pisier. The "maximal" tensor product of operator spaces. *Proceedings of the Edinburgh Mathematical Society*, 42(2):267–284, 1999.

38 C. Palazuelos and T. Vidick. Survey on nonlocal games and operator space theory. *Journal of Mathematical Physics*, 57(1):015220, 2016.

39 V. Paulsen. *Completely bounded maps and operator algebras*, volume 78. Cambridge University Press, Cambridge, 2002.

40 V. I. Paulsen and R. R. Smith. Multilinear maps and tensor norms on operator systems. *Journal of functional analysis*, 73(2):258–276, 1987.

41 D. Pérez-García, M. Wolf, C. Palazuelos, I. Villanueva, and M. Junge. Unbounded violation of tripartite Bell inequalities. *Communications in Mathematical Physics*, 279:455, 2008. arXiv:quant-ph/0702189.

42 G. Pisier. *Introduction to operator space theory*, volume 294 of *London Mathematical Society Lecture Note Series*. Cambridge University Press, Cambridge, 2003.

43 G. Pisier. Grothendieck's theorem, past and present. *Bull. Amer. Math. Soc.*, 49(2):237–323, 2012. also available at arXiv:1101.4195.

44 D. Pollard. *Convergence of stochastic processes*. Science & Business Media. Springer, 2012.

45 J. Reeds. A new lower bound on the real Grothendieck constant. Manuscript (`http://www.dtc.umn.edu/~reedsj/bound2.dvi`), 1991.

46 B. Reichardt. Span programs and quantum query complexity: The general adversary bound is nearly tight for every boolean function. In *50th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2009*, pages 544–551, 2009. arXiv:0904.2759.

47 B. Reichardt. Reflections for quantum query algorithms. In *Proceedings of the Twenty-Second Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2011*, pages 560–569, 2011. arXiv:1005.1601.

48 P. W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Journal of Computing*, 26(5):1484–1509, 1997. Earlier version in FOCS'94.

49 D. Simon. On the power of quantum computation. *Siam journal of computing*, 26(5):1474–1483, 1997. Earlier version in FOCS'94.

50 R.R. Smith. Completely bounded multilinear maps and Grothendieck's inequality. *Bulletin of the London Mathematical Society*, 20(6):606–612, 1988.

51 T. Tao. *Topics in random matrix theory*, volume 132. American Mathematical Society, 2012.

52 J. A. Tropp. Column subset selection, matrix factorization, and eigenvalue optimization. In *Proceedings of the Twentieth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 978–986, 2009. arXiv:0806.4404.

53 B. S. Tsirelson. Quantum analogues of the Bell inequalities. The case of two spatially separated domains. *J. Soviet Math.*, 36:557–570, 1987.

54 R. de Wolf. Nondeterministic quantum query and communication complexities. *SIAM J. Comput.*, 32(3):681–699, 2003. cs.CC/0001014.

A Complete Characterization of Unitary Quantum Space*†‡

Bill Fefferman[1] and Cedric Yen-Yu Lin[2]

1 **Electrical Engineering and Computer Science, University of California, Berkeley and Joint Center for Quantum Information and Computer Science (QuICS), University of Maryland, and NIST, Gaithersburg, MD, USA**
wjf@berkeley.edu

2 **Joint Center for Quantum Information and Computer Science (QuICS), University of Maryland, College Park, MD, USA**
cedricl@umiacs.umd.edu

Abstract

Motivated by understanding the power of quantum computation with restricted number of qubits, we give two complete characterizations of unitary quantum space bounded computation. First we show that approximating an element of the inverse of a well-conditioned efficiently encoded $2^{k(n)} \times 2^{k(n)}$ matrix is complete for the class of problems solvable by quantum circuits acting on $\mathcal{O}(k(n))$ qubits with all measurements at the end of the computation. Similarly, estimating the minimum eigenvalue of an efficiently encoded Hermitian $2^{k(n)} \times 2^{k(n)}$ matrix is also complete for this class. In the logspace case, our results improve on previous results of Ta-Shma [30] by giving new space-efficient quantum algorithms that avoid intermediate measurements, as well as showing matching hardness results.

Additionally, as a consequence we show that PreciseQMA, the version of QMA with exponentially small completeness-soundess gap, is equal to PSPACE. Thus, the problem of estimating the minimum eigenvalue of a *local* Hamiltonian to inverse exponential precision is PSPACE-complete, which we show holds even in the frustration-free case. Finally, we can use this characterization to give a provable setting in which the ability to prepare the ground state of a local Hamiltonian is more powerful than the ability to prepare PEPS states.

Interestingly, by suitably changing the parameterization of either of these problems we can completely characterize the power of quantum computation with *simultaneously* bounded time and space.

1998 ACM Subject Classification F.1.3 Complexity Measures and Classes, F.2 Analysis of Algorithms and Problem Complexity

Keywords and phrases Quantum complexity, space complexity, complete problems, QMA

Digital Object Identifier 10.4230/LIPIcs.ITCS.2018.4

1 Introduction

How powerful is quantum computation with a restricted number of qubits? In this work we will study unitary quantum space-bounded classes - those problems solvable using a given amount of (quantum and classical) space, with all quantum measurements performed at the

* This work was supported by the Department of Defense.
† A full version of the paper is available at https://arxiv.org/abs/1604.01384
‡ This work is a contribution of the National Institute of Standards and Technology and is not subject to U.S. copyright.

end of the computation. We give two sets of complete problems for these classes; to the best of our knowledge, these are the first natural complete problems proposed for quantum space-bounded classes.

The first problem we consider, the $k(n)$-*Well-conditioned Matrix Inversion* problem, is a well-conditioned version of the ubiquitous matrix inversion problem. The second problem we consider, the $k(n)$-*Minimum Eigenvalue* problem, asks us to compute the minimum eigenvalue of a Hermitian matrix to high precision – in the context of quantum complexity, this is a natural generalization of the familiar local Hamiltonian problem [23]. Interestingly enough, the first (resp. second) problem is the space-bounded variant of a BQP-complete [18] (resp. QMA-complete [23]) problem; their complexities coincide in the space-bounded regime. For the sake of readability, we defer precise definitions of these problems and statements of our results until Sections 3 and 4.

We now proceed to give some justification for the importance of our results. In the following discussion, $\mathsf{BQ_USPACE}[k(n)]$ refers to the class of problems solvable with bounded error by a quantum algorithm running in $\mathcal{O}(k(n))$; the subscript $\mathbf{U}$ indicates that the algorithm is unitary, i.e. employs no intermediate measurements.

1.1 Background and Motivation

The Matrix Inversion problem is of central importance in computational complexity theory. Matrix inversion is known to be complete for DET, the class of functions as hard as computing the determinant of an integer matrix, which can be solved in classical $\mathcal{O}(\log^2(n))$ space [5, 12]. It is a major open problem to determine if Matrix Inversion can be solved in classical logarithmic space, which would imply $\mathsf{L} = \mathsf{NL} = \mathsf{DET}$.

Recently, Ta-Shma [30], building on work of Harrow, Hassidim, and Lloyd [18], showed that a well-conditioned $n \times n$ matrix can be inverted (up to $1/\operatorname{poly}(n)$ error) by a quantum $O(\log n)$ space algorithm using intermediate measurements. Similarly, Ta-Shma also gives an algorithm for computing eigenvalues of a Hermitian matrix with similar space. These algorithms achieve a quadratic advantage in space over the best known classical algorithms, which require $\Omega(\log^2 n)$ space. This is the maximum quantum advantage possible, since Watrous has shown $\mathsf{BQSPACE}[k(n)] \subseteq \mathsf{SPACE}[\mathcal{O}(k(n)^2)]$ [35, 36] even for quantum algorithms with intermediate measurements.

Our completeness result for matrix inversion, along with observing our algorithm for matrix inversion (Theorem 14) actually gives a high-precision approximation, gives the following corollary in the logspace case (see Remark 2.3).

▶ **Corollary 1.** *The problem of approximating, to constant precision, an entry of the inverse of an $n \times n$ positive semidefinite matrix with condition number at most $\operatorname{poly}(n)$ is $\mathsf{BQ_UL}$-complete under L-reductions, where $\mathsf{BQ_UL}$ is the set of problems solvable in unitary quantum logspace. This problem remains in $\mathsf{BQ_UL}$ even if $1/\operatorname{poly}(n)$ precision is required.*

Similarly, restricting Thereom 4 to the logspace case gives the following corollary.

▶ **Corollary 2.** *The problem of approximating, to $1/\operatorname{poly}(n)$ precision, the minimum eigenvalue of an $n \times n$ positive semidefinite matrix is $\mathsf{BQ_UL}$-complete under L-reductions.*

These corollaries improve upon Ta-Shma's results [30] in two ways. First, our algorithms solve these problems without needing intermediate measurements. Unlike in time complexity, where the "Principle of safe storage" gives a time-efficient procedure to defer intermediate measurements, these methods may incur an exponential blow-up in space.

One might wonder why we care so much about avoiding intermediate measurements. The main reason is that removing intermediate measurements from the computation allows us to give matching hardness results, showing the optimality of our algorithms. This is the second way our results improve on those of Ta-Shma. In particular, our proofs crucially use space-efficient methods for the amplification of unitary quantum computations, which are not known in the non-unitary model. This is because the techniques require applying the inverse of the circuit, which of course is impossible if the circuit contains intermediate measurements. We will also rely on ideas from Kitaev's clock construction, which constructs a local Hamiltonian from a unitary quantum circuit.

Specifically, we will show that the problems of inverting well-conditioned matrices and computing minimum eigenvalues of Hermitian matrices are hard for unitary quantum logspace under L-reductions. In the case of our algorithm for Matrix Inversion, this means that the upper bound on the condition number bound is unlikely to be improved upon. Likewise, this gives some of the strongest evidence that even well-conditioned matrices cannot be inverted in deterministic logspace, since otherwise our results would immediately imply $\mathsf{L} = \mathsf{BQ_UL}$, which seems quite unlikely.

Interestingly, although our algorithms for both problems use different techniques from those of Ta-Shma, our algorithm for computing the minimum eigenvalue is completely different. In particular, our algorithm crucially relies on new methods for space efficient QMA amplification, together with some powerful recent results in Hamiltonian simulation [6, 8].

Concurrently with our work, Doron, Sarid, and Ta-Shma have shown that analogous problems for stochastic matrices (e.g. computing the eigenvalue gap) are complete for classical randomized logspace, or BPL [14, 13]. In addition, Le Gall has shown that analogous problems for Laplacian matrices can be solved in BPL [16]. Since it is straightforward to see that Well-conditioned Matrix Inversion reduces to Integer Matrix Inversion, we obtain a direct proof that $\mathsf{BQ_UL} \subseteq \mathsf{DET}$, which was previously known indirectly via the containments $\mathsf{BQ_UL} \subseteq \mathsf{PL} \subseteq \mathsf{DET}$ [36, 9].

Therefore the power of classical and quantum space-bounded classes are characterized by the ability to approximate solutions of different problems in DET (stochastic matrices for the former, and Hermitian matrices for the latter). This could shed light on the differences between deterministic, randomized, and quantum space complexity. An open question is to find a class of interesting matrices whose inverse (or eigenvalues) can be computed in *deterministic* logspace.

Interestingly, if we change the scaling of the parameters in our Matrix Inversion and Minimum Eigenvalue problems suitably, then we obtain problems that are known to be complete for BQP [18] and QMA [23, 2]. Thus by appropriately bounding the dimension of the matrix and either the condition number or the promise gap, we can give problems complete for quantum time or quantum space. In fact we can strengthen these results to settings with a simultaneously bounded amount of space and time; see Section 5.

1.2 Relationship with Matchgates

Matchgates are a subclass of quantum gates introduced by Valiant [32], who also showed that nearest neighbor matchgate circuits (which we will just call matchgate computations) are classically simulable. Matchgate computations were further shown to be equivalent to a one-dimensional model of noninteracting fermions by Terhal and DiVincenzo [31]; and equivalent to unitary quantum logspace by Jozsa, Kraus, Miyake, and Watrous [20]. Our complete problems therefore elucidate the computational power of noninteracting fermions.

We know that sampling from output distributions of matchgate computations gives us the power of BPL; but what is the computational power of computing exactly the output probabilities of matchgate computations? We conjecture that this computational power corresponds to DET, since amplitudes of noninteracting fermion circuits are related to determinants (and see also the discussion in the previous subsection). It is known that output probabilities of matchgate computations can be exactly calculated by an efficient classical algorithm [21], which is consistent with our conjecture because $\mathsf{DET} \in \mathsf{P}$.

1.3 Quantum Merlin-Arthur with Small Gap

A consequence of our proof of completeness for the $k(n)$-*Minimum Eigenvalue* problem is an equivalence between space-bounded quantum computations and quantum Merlin-Arthur proof systems. Here we give this equivalence for the polynomial space case: let PreciseQMA be the variant of QMA with exponentially small completeness-soundness gap. Then we show the following:

▶ **Corollary 3.** $\mathsf{PreciseQMA} = \mathsf{BQ_UPSPACE} = \mathsf{PSPACE}$.

The second equality is due to Watrous [35, 36]. We give similar equivalences for space-bounded quantum computations with and without a witness for other space bounds as well (Theorem 18).

We note that PreciseQMA is likely far more powerful than its classical counterpart. The analogous classical complexity class is contained in $\mathsf{NP}^{\mathsf{PP}}$: given a classical witness, the verifier runs a classical computation that in the YES case accepts with probability at least c, or in the NO case accepts with probability at most s, where $c > s$. Note that in the classical case $c - s > \exp(-\operatorname{poly})$ is automatically satisfied. Since $\mathsf{NP}^{\mathsf{PP}}$ is in the counting hierarchy, the entirety of which is contained in PSPACE (see e.g., [3]), we see that the quantum proof protocol is strictly stronger than the classical one, unless the counting hierarchy collapses to the second level.

We also show that the *local* Hamiltonian problem is PSPACE-complete when the promise gap is exponentially small (for details see Appendix D). This is in contrast to the usual case when the gap is polynomially small, where the problem is QMA-complete. Perhaps more surprisingly, $\mathsf{PreciseQMA} = \mathsf{PSPACE}$ is more powerful than $\mathsf{PostBQP} = \mathsf{PP}$, the class of problems solvable with postselected quantum computation [1].

Another consequence concerns Projected Entangled Pair States, or PEPS, a natural extension of matrix product states to two and higher spatial dimensions, which can be described as the ground state of certain frustration-free local Hamiltonians [34]. A characterization of the computational power of PEPS was given in [28], and can be summarized as follows: let O_{PEPS} be a quantum oracle that, given the description of a PEPS, outputs the PEPS (so the output is quantum). Then $\mathsf{BQP}^{O_{PEPS}}_{\|,\text{classical}} = \mathsf{PostBQP} = \mathsf{PP}$, where (following Aaronson [1]) the subscript denotes that only classical nonadaptive queries to the oracle are allowed. Moreover, let PQP be the set of problems solvable by a quantum computer with *unbounded error*; then it is straightforwardly shown that $\mathsf{PQP}^{O_{PEPS}}_{\|,\text{classical}} = \mathsf{PP}$ as well (see Appendix F).

On the other hand, suppose we have an oracle O_{LH} that given the description of a local Hamiltonian, outputs a ground state of the Hamiltonian. Then our results show that $\mathsf{PreciseQMA} = \mathsf{PSPACE} \subseteq \mathsf{PQP}^{O_{LH}}_{\|,\text{classical}}$. This shows that in the setting of unbounded-error quantum computation, PEPS do not capture the full computational complexity of general local Hamiltonian ground states unless $\mathsf{PP} = \mathsf{PSPACE}$. We leave open the problem of determining the complexity of $\mathsf{BQP}^{O_{LH}}_{\|,\text{classical}}$.

Lastly, we are able to strengthen our characterization to show that PreciseQMA contains PSPACE (see Appendix C), even when restricted to having perfect completeness. This allows us to prove that testing if a local Hamiltonian is frustration-free is a PSPACE-complete problem (Appendix D). We note that if the local Hamiltonian is promised to have a ground state energy of at least $1/\operatorname{poly}$ if it is frustrated, then this is the Quantum Satisfiability problem defined by Bravyi, which is known to be QMA_1 complete [10, 17]. Our results show that if the promise gap is removed then we instead get PSPACE-completeness.

2 Preliminaries

2.1 Quantum circuits

We will assume a working knowledge of quantum information. For an introduction, see [26].

A *quantum circuit* consists of a series of quantum gates each taken from some universal gateset, such as the gateset consisting of Hadamard and Toffoli gates [29]. For functions $f, g : \mathbb{N} \to \mathbb{N}$, we say a family of quantum circuits $\{Q_x\}_{x\in\{0,1\}^*}$ is *f-time g-space uniformly generated* if there exists a deterministic classical Turing machine that on input $x \in \{0,1\}^n$ and $i > 0$ outputs the i-th gate of Q_x within time $f(n)$ and workspace $g(n)$ [26].

Our restriction to a specific gateset is without loss of generality, even for logarithmic space algorithms: there exists a deterministic algorithm that given any unitary quantum gate U and a parameter ϵ, outputs a sequence of at most $\operatorname{polylog}(1/\epsilon)$ gates from any universal quantum gateset that approximates U to precision ϵ in space $\mathcal{O}(\log(1/\epsilon))$ and time $\operatorname{polylog}(1/\epsilon)$ [33]. This improves the Solovay-Kitaev theorem, which guarantees a space bound of $\operatorname{polylog}(1/\epsilon)$; see e.g., [26].

2.2 Space-bounded computation

For our model of unitary quantum space-bounded computation, we consider a quantum system with purely classical control, because there are no intermediate quantum measurements to condition future operations on. Specifically, we use the following definition (see Appendix A for more details):

▶ **Definition 4.** Let $k(n)$ be a function satisfying $\Omega(\log(n)) \leq k(n) \leq \operatorname{poly}(n)$. A promise problem $L = (L_{yes}, L_{no})$ is in $\mathsf{Q_U SPACE}[k(n)](c, s)$ if there exists a $\operatorname{poly}(|x|)$-time $\mathcal{O}(k)$-space uniformly generated family of quantum circuits $\{Q_x\}_{x\in\{0,1\}^*}$, where each circuit $Q_x = U_{x,T}U_{x,T-1}\cdots U_{x,1}$ has $T = 2^{\mathcal{O}(k)}$ gates, and acts on $\mathcal{O}(k(|x|))$ qubits, such that:
If $x \in L_{yes}$:

$$\left\langle 0^k \middle| Q_x^\dagger |1\rangle\langle 1|_{out} Q_x \middle| 0^k \right\rangle \geq c. \tag{1}$$

Whereas if $x \in L_{no}$:

$$\left\langle 0^k \middle| Q_x^\dagger |1\rangle\langle 1|_{out} Q_x \middle| 0^k \right\rangle \leq s. \tag{2}$$

Here *out* denotes a single qubit we measure at the end of the computation; no intermediate measurements are allowed. Furthermore, we require c and s to be computable in classical $\mathcal{O}(k(n))$-space.

For the rest of the paper we will always assume that $\Omega(\log(n)) \leq k(n) \leq \operatorname{poly}(n)$.

The bound $T = 2^{\mathcal{O}(k)}$ on the circuit size comes from that any classical Turing machine generating Q_x using space $\mathcal{O}(k(|x|))$ has at most $2^{\mathcal{O}(k)}$ configurations. We note that $2^{\mathcal{O}(k)}$

gates suffice to approximate any gate on $\mathcal{O}(k)$ qubits to high accuracy (see e.g. [26, Chapter 4]). The poly($|x|$) time bound on the classical control can be assumed without loss of generality; see Appendix A.

▶ **Definition 5.** $\mathsf{BQ_USPACE}[k] = \mathsf{Q_USPACE}[k](2/3, 1/3)$.

▶ **Theorem 6** (Watrous [35, 36])**.** $\mathsf{BQ_USPACE}[\mathrm{poly}] = \mathsf{PSPACE}$.

We now define space- and time-bounded analogues of QMA:

▶ **Definition 7.** We say a promise problem $L = (L_{yes}, L_{no})$ is in (t, k)-*bounded* $\mathsf{QMA}_m(c, s)$ if there exists a t-time and $(k + m)$-space uniformly generated family of quantum circuits $\{V_x\}_{x\in\{0,1\}^*}$, each of size at most $t(|x|)$, acting on $k(|x|) + m(|x|)$ qubits, so that:

If $x \in L_{yes}$ there exists an m-qubit state $|\psi\rangle$ such that:

$$\left(\langle\psi| \otimes \left\langle 0^k\right|\right) V_x^\dagger |1\rangle\langle 1|_{out} V_x \left(|\psi\rangle \otimes \left|0^k\right\rangle\right) \geq c. \tag{3}$$

Whereas if $x \in L_{no}$, for all m-qubit states $|\psi\rangle$ we have:

$$\left(\langle\psi| \otimes \left\langle 0^k\right|\right) V_x^\dagger |1\rangle\langle 1|_{out} V_x \left(|\psi\rangle \otimes \left|0^k\right\rangle\right) \leq s. \tag{4}$$

out denotes a single qubit measured at the end of the computation; no intermediate measurements are allowed. Here c and s are computable in classical $\mathcal{O}(t)$-time and $\mathcal{O}(k + m)$-space.

▶ **Definition 8.** $\mathsf{QMA} = (\mathrm{poly}, \mathrm{poly})$-*bounded* $\mathsf{QMA}_{\mathrm{poly}}(2/3, 1/3)$.

▶ **Definition 9.** $\mathsf{PreciseQMA} = \bigcup_{c\in(0,1]} (\mathrm{poly}, \mathrm{poly})$-*bounded* $\mathsf{QMA}_{\mathrm{poly}}(c, c - 2^{-\mathrm{poly}})$.

2.3 Other definitions and results

We use the following definition of *efficient encodings* of matrices:

▶ **Definition 10.** Let M be a $2^k \times 2^k$ matrix, and $\mathcal{A}$ be a classical algorithm (e.g. a Turing machine) specified using n bits. We say that $\mathcal{A}$ is an *efficient encoding* of M if on input $i \in \{0, 1\}^k$, $\mathcal{A}$ outputs the indices and contents of the non-zero entries of the i-th row, using at most poly(n) time and $\mathcal{O}(k)$ workspace (not including the output size). Note that as a consequence M has at most poly(n) nonzero entries in each row.

We will often specify a matrix M in the input by giving an efficient encoding of M. The size of the encoding is then the input size, which we will usually indicate by n.

▶ Remark. It is not difficult to see that every $n \times n$ matrix has an efficient encoding of size $\mathcal{O}(n^2)$, since it is straightforward to construct a classical $\mathcal{O}(\log n)$-space circuit that on input i, j outputs the (i, j)-entry of the matrix.

In our results we will implicitly assume the existence of algorithms that compute some common functions on n-bit numbers, such as $\sin, \cos, \arcsin, \arccos$ and exponentiation, to within $1/\mathrm{poly}(n)$ accuracy in classical $\mathcal{O}(\log n)$ space. Algorithms for these tasks have been designed by Reif [27].[1]

Finally, we will need new results from the Hamiltonian simulation literature:

[1] Reif's algorithms take only $\mathcal{O}(\log\log n \log\log\log n)$ space, but we only need the $\mathcal{O}(\log n)$ bound.

▶ **Theorem 11** ([6, 7, 8]). *Suppose we are given as input the size-n efficient encoding of a $2^{k(n)} \times 2^{k(n)}$ Hermitian matrix H. Then treated as a Hamiltonian, the time evolution $\exp(-iHt)$ can be simulated using $\mathrm{poly}(n, k, \|H\|_{max}, t, \log(1/\epsilon))$ operations and $\mathcal{O}(k + \log(t/\epsilon))$ space.*

While the space complexity was not explicitly stated in [6, 7, 8], it can be seen from the analysis (see e.g. [7]). The crucial thing to notice in Theorem 11 is the polylogarithmic scaling in the error ϵ; this implies that we can obtain polynomial precision in $\exp(-iHt)$ using only polynomially many operations. Also note that the maximum eigenvalue of H, $\|H\|$, satisfies $\|H\| \leq \mathrm{poly}(n)\|H\|_{max}$.

3 The Well-Conditioned Matrix Inversion Problem

We begin with a formal statement of the problem:

▶ **Definition 12** ($k(n)$-*Well-conditioned Matrix Inversion*)**.** Given as input is the size-n efficient encoding of a $2^{k(n)} \times 2^{k(n)}$ positive semidefinite matrix H with a known upper bound $\kappa = 2^{\mathcal{O}(k(n))}$ on the condition number, so that $\kappa^{-1}I \preceq H \preceq I$, and $s, t \in \{0,1\}^{k(n)}$. It is promised that either $|H^{-1}(s,t)| \geq b$ or $|H^{-1}(s,t)| \leq a$ for some constants $0 \leq a < b \leq 1$; determine which is the case.

▶ **Theorem 13.** *For $\Omega(\log(n)) \leq k(n) \leq \mathrm{poly}(n)$, $\mathcal{O}(k(n))$-Well-conditioned Matrix Inversion is complete for $\mathsf{BQ_USPACE}[\mathcal{O}(k(n))]$ under classical reductions using $\mathrm{poly}(n)$ time and $\mathcal{O}(k(n))$ space.*

Proof. We begin by giving a new space efficient algorithm for this matrix inversion problem:

▶ **Theorem 14.** *Fix functions $k(n)$, $\kappa(n)$, and $\epsilon(n)$. Suppose we are given the size-n efficient encoding of a $2^{k(n)} \times 2^{k(n)}$ PSD matrix H such that $\kappa^{-1}I \preceq H \preceq I$. We are also given $\mathrm{poly}(n)$-time $\mathcal{O}(k + \log(\kappa/\epsilon))$-space uniform quantum circuits U_a and U_b acting on k qubits and using at most T gates. Let $U_a|0\rangle^{\otimes k(n)} = |a\rangle$ and $U_b|0\rangle^{\otimes k(n)} = |b\rangle$. The following tasks can be performed with $\mathrm{poly}(n)$-time $O(k + \log(\kappa/\epsilon))$-space uniformly generated quantum circuits with $\mathrm{poly}(T, k, \kappa, 1/\epsilon)$ gates and $\mathcal{O}(k + \log(\kappa/\epsilon))$ qubits:*

1. *With at least constant probability, output an approximation of the quantum state $H^{-1}|b\rangle/\|H^{-1}|b\rangle\|$ up to error ϵ.*
2. *Approximate $\|H^{-1}|b\rangle\|$ to precision ϵ.*
3. *Approximate $|\langle a|H^{-1}|b\rangle|$ to precision ϵ.*

These circuits do not require intermediate measurements.

In fact our algorithm is much stronger: to solve $k(n)$-*Well-conditioned Matrix Inversion* we merely need to approximate $|\langle s|H^{-1}|t\rangle|$ to constant precision, while Theorem 14 actually gives an approximation to precision $2^{-\mathcal{O}(k)}$ in $\mathcal{O}(k(n))$ unitary quantum space. Moreover our algorithm does not require s and t to be computational basis states. We can also approximate $\langle s|H^{-1}|t\rangle$ (without the absolute value), since if we choose $|a\rangle = (|s\rangle + |t\rangle)/\sqrt{2}$ and $|b\rangle = (|s\rangle + i|t\rangle)/\sqrt{2}$ we have

$$\langle s|H^{-1}|t\rangle = \langle a|H^{-1}|a\rangle - i\langle b|H^{-1}|b\rangle + (i-1)(\langle s|H^{-1}|s\rangle + \langle t|H^{-1}|t\rangle)/2 \tag{5}$$

and e.g. $\langle a|H^{-1}|a\rangle = |\langle a|H^{-1}|a\rangle|$ becaue H is positive semidefinite.

We note that we can modify our definition of unitary quantum space-bounded classes to include computing functions, for instance by adding a write-only one-way output tape of qubits to the Turing machine (see the discussion in Appendix A), that are measured at

the end of the computation. The error reduction result (Corollary 29) later in our work allows the total error to be reasonably controlled. With such a modification we can output an approximation to the entire matrix inverse in unitary quantum logspace. We will not pursue this modified model further in this work.

Proof. We first briefly summarize the algorithm of Ta-Shma [30], which is based on the linear systems solver of Harrow, Hassidim and Lloyd [18]. Ta-Shma shows that an $n \times n$ matrix with condition number at most $\mathrm{poly}(n)$ can be inverted by a quantum logspace algorithm with intermediate measurements; in our language this corresponds to solving $\mathcal{O}(\log n)$-*Well-conditioned Matrix Inversion.*

Our algorithm and Ta-Shma's share the same initial procedure. In particular it is shown:

▶ **Lemma 15** (Implicit in [18, 30]). *There is a* poly*-time $\mathcal{O}(k + \log(\kappa/\epsilon'))$-space uniform quantum unitary transformation W_H over $k + \ell = \mathcal{O}(k + \log(\kappa/\epsilon'))$ qubits and using* $\mathrm{poly}(k, \kappa/\epsilon')$ *gates, such that for any k-qubit input state $|b\rangle$,*

$$W_H(|0\rangle^{\otimes \ell} \otimes |b\rangle) = \alpha|0\rangle_{out} \otimes |\psi_b\rangle + \sqrt{1-\alpha^2}|1\rangle_{out} \otimes |\psi_b'\rangle, \tag{6}$$

where $|\psi_b\rangle$ and $|\psi_b'\rangle$ are normalized states such that $\| |\psi_b\rangle - |0\rangle^{\otimes \ell - 1} \otimes \frac{H^{-1}|b\rangle}{\|H^{-1}|b\rangle\|} \| \leq \epsilon'$, α is a positive number satisfying $|\alpha - \frac{\|H^{-1}|b\rangle\|}{\kappa}| \leq \epsilon'$, and "out" is a 1-qubit register.

This lemma can be obtained by combining the Hamiltonian simulation algorithms of Berry et al. (Theorem 11) with the analysis of Harrow, Hassidim and Lloyd [18]; a version without the time bound is implicit in the proof of [30, Theorem 6.3]. For completeness, we sketch the proof below.

Proof sketch. Decompose $|b\rangle$ into the eigenbasis of H: $|b\rangle = \sum_\lambda a_\lambda |v_\lambda\rangle$, where λ are eigenvalues of H and $H|v_\lambda\rangle = \lambda|v_\lambda\rangle$. The following procedure satisfies Lemma 15 (all steps are approximate):

1. Perform phase estimation on the operator $\exp(iH)$ and state $|b\rangle$ to compute the eigenvalues of H into an ancilliary register, obtaining the state $\sum_\lambda a_\lambda |v_\lambda\rangle|\lambda\rangle$.
2. Implement the unitary transformation $|\lambda\rangle|0\rangle \to |\lambda\rangle[(\kappa\lambda)^{-1}|0\rangle + (\sqrt{1-(\kappa\lambda)^{-2}}|1\rangle)]$, to obtain the state $\sum_\lambda a_\lambda |v_\lambda\rangle|\lambda\rangle[(\kappa\lambda)^{-1}|0\rangle + (\sqrt{1-(\kappa\lambda)^{-2}}|1\rangle)]$.
3. Uncompute the eigenvalues λ by running phase estimation in reverse, obtaining the state $\sum_\lambda a_\lambda |v_\lambda\rangle|0\rangle^{\ell-1}[(\kappa\lambda)^{-1}|0\rangle + (\sqrt{1-(\kappa\lambda)^{-2}}|1\rangle)]$. Note that $\sum_\lambda a_\lambda |v_\lambda\rangle|0\rangle^{\ell-1}(\kappa\lambda)^{-1}|0\rangle = \frac{1}{\kappa}H^{-1}|b\rangle$.

An appropriate error analysis of this procedure is the technical bulk of the proof; we refer the reader to [18]. For Step 1, Ta-Shma showed how to implement $\exp(iH)$ in $\mathcal{O}(k + \log(1/\epsilon))$ space [30, Theorem 4.1] (their proof works for general matrices with efficient encodings); recent sparse Hamiltonian simulation algorithms (Theorem 11) give a time efficient way to do this. ◀

Intuitively, Lemma 15 gives a space-efficient quantum algorithm that produces a state proportional to $H^{-1}|b\rangle$ with probability at least $1/\kappa$. Our goal is to amplify the probability from $1/\kappa$ to a constant, to produce a state with constant overlap to the state $|0\rangle_{out} \otimes |\psi_b\rangle$ together with an estimate for $\alpha \approx \|H^{-1}|b\rangle\|$. From here our algorithm differs from Ta-Shma's and uses a combination of amplitude amplification and phase estimation. This sidesteps both the somewhat involved analysis and intermediate measurements of Ta-Shma's algorithm.

Specifically, consider the two projectors

$$\Pi_0 = |0\rangle\langle 0|^{\otimes \ell} \otimes |b\rangle\langle b|, \quad \Pi_1 = W_H^\dagger(|0\rangle\langle 0|_{out} \otimes I)W_H. \tag{7}$$

Π_0 projects onto the initial subspace, while Π_1 projects onto the initial states that would be accepted by the final measurement. The rotation $R = -(I - 2\Pi_1)(I - 2\Pi_0)$ has eigenvalues $e^{\pm i2\sin^{-1}\alpha}$ with eigenvectors $|\psi_+\rangle$, such that $|0\rangle^{\otimes\ell} \otimes |b\rangle = (|\psi_+\rangle + |\psi_-\rangle)/\sqrt{2}$ is a uniform superposition of the two eigenvectors. Therefore phase estimation on the operator R and input state $|0\rangle^{\otimes\ell} \otimes |b\rangle$ suffices to give an estimate of α. Furthermore both eigenvectors have constant overlap with $W_H^\dagger|\psi_b\rangle$, so applying W_H to the residual state of phase estimation allows us to complete the first task as well.

We have addressed the first two tasks in Theorem 14. For the third task (approximating $|\langle a|H^{-1}|b\rangle|$), we can choose $\Pi_1' = W_H^\dagger(|0\rangle\langle 0|_{out} \otimes I)(I_{anc} \otimes |a\rangle\langle a|)(|0\rangle\langle 0|_{out} \otimes I)W_H$ instead, and phase estimation on $R = -(I - 2\Pi_1)(I - 2\Pi_0')$ will give an estimate for $|\langle a|H^{-1}|b\rangle|$. See the full version of the paper for the complete proof. ◀

We establish that $k(n)$-*Well-conditioned Matrix Inversion* is $\mathsf{BQ_USPACE}[\mathcal{O}(k)]$-hard using a similar argument to Harrow, Hassidim, and Lloyd [18], in which given a quantum circuit acting on $k(n)$ qubits we construct a efficiently encoded well-conditioned $2^{\mathcal{O}(k)} \times 2^{\mathcal{O}(k)}$ matrix H, so that a single element of H^{-1} is proportional to the success probability of the circuit. See the full version of our paper for a detailed proof. ◀

4 The Minimum Eigenvalue Problem

Our second characterization of unitary quantum space is based on the following problem:

▶ **Definition 16** ($k(n)$-*Minimum Eigenvalue* problem)**.** Given as input is the size-n efficient encoding of a $2^{k(n)} \times 2^{k(n)}$ PSD matrix H, such that $\|H\|_{max} = \max_{s,t}|H(s,t)|$ is at most a constant. Let λ_{min} be the minimum eigenvalue of H. It is promised that either $\lambda_{min} \leq a$ or $\lambda_{min} \geq b$, where $a(n)$ and $b(n)$ are numbers such that $b - a > 2^{-\mathcal{O}(k(n))}$. Output 1 if $\lambda_{min} \leq a$, and output 0 otherwise.

▶ **Theorem 17.** *For* $\Omega(\log(n)) \leq k(n) \leq \mathrm{poly}(n)$, $\mathcal{O}(k(n))$*-Minimum Eigenvalue is complete for* $\mathsf{BQ_USPACE}[\mathcal{O}(k(n))]$ *under classical reductions using* $\mathrm{poly}(n)$ *time and* $\mathcal{O}(k(n))$ *space.*

In the process of proving this result, we will also show the following equivalence:

▶ **Theorem 18.** $\mathsf{BQ_USPACE}[\mathcal{O}(k(n))]$ *is equivalent to the class of problems characterized by having quantum Merlin Arthur proof systems running in polynomial time,* $\mathcal{O}(k(n))$ *witness size and space, and* $2^{-\mathcal{O}(k(n))}$ *completeness-soundness gap. Or in other words,*

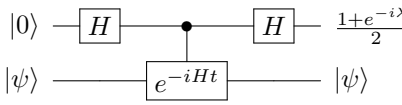
$$\mathsf{BQ_USPACE}[\mathcal{O}(k(n))] = \bigcup_{c-s \geq 2^{-\mathcal{O}(k(n))}} (\mathrm{poly}, \mathcal{O}(k(n)))\text{-bounded } \mathsf{QMA}_{\mathcal{O}(k(n))}(c,s)$$

Our proof will consist of three steps. Lemma 19 will show that $k(n)$-*Minimum Eigenvalue* is in the generalized $\mathsf{PreciseQMA}$ class defined in Theorem 18. Lemma 20 will show that this generalized $\mathsf{PreciseQMA}$ class is contained in $\mathsf{BQ_USPACE}[k(n)]$. Finally, Lemma 21 will show that $\mathsf{BQ_USPACE}[k(n)]$-hardness of $k(n)$-*Minimum Eigenvalue*.

▶ **Lemma 19.** $k(n)$*-Minimum Eigenvalue is contained in* $(\mathrm{poly}, \mathcal{O}(k(n)))$*-bounded* $\mathsf{QMA}_{\mathcal{O}(k(n))}(c,s)$ *for some* c, s *such that* $c - s > 2^{-\mathcal{O}(k(n))}$.

Proof. We are given the size-n efficient encoding of a $2^{k(n)} \times 2^{k(n)}$ PSD matrix H, and it is promised that the smallest eigenvalue λ_{min} of H is either at most a or at least b. Merlin would like to convince us that $\lambda_{min} \leq a$; he will send us a purported k-qubit eigenstate $|\psi\rangle$ of H

with eigenvalue λ_{min}. Choose $t = \pi/(\text{poly}(n)\|H\|_{max}) \leq \pi/\|H\|$; then all eigenvalues of Ht lie in the range $[0, \pi]$, and the output of phase estimation on $\exp(-iHt)$ will be unambiguous. We perform, on ψ, phase estimation of $\exp(-iHt)$ with one bit of precision:



(8)

Here the H gates on the first qubit are Hadamard gates (and have nothing to do with the matrix H). Theorem 11 gives an implementation of $\exp(-iHt)$ up to error $\epsilon = 2^{-\Theta(k(n))}$ using $\text{poly}(n)$ operations and $\mathcal{O}(k(n))$ space.

In Circuit (8) we've assumed $|\psi\rangle$ is an eigenstate of H with eigenvalue λ. If we measure the control qubit at the end, the probability we obtain 0 is $(1 + \cos(\lambda t))/2$. Therefore if ψ is a eigenstate with eigenvalue at most a, we can verify this with probability at least $c = (1 + \cos(at))/2 - \epsilon$, where ϵ is the error in the implementation of $\exp(-iHt)$. Otherwise if $\lambda_{min} \geq b$, no state ψ will be accepted with probability more than $s = (1 + \cos(bt))/2 + \epsilon$. The separation between c and s is at least

$$(\cos(at) - \cos(bt)) - 2\epsilon = 2\sin\left(\frac{(a+b)t}{2}\right)\sin\left(\frac{(b-a)t}{2}\right) - 2\epsilon \geq 2^{-\mathcal{O}(k)} \tag{9}$$

since $\sin x = \Omega(x)$ for $x \in [0, 1]$, $(a + b)t \geq (b - a)t = 2^{-\mathcal{O}(k(n))}$, as long as we choose $\epsilon = 2^{-\Theta(k(n))}$ to be sufficiently small enough. This therefore gives a $(\text{poly}, \Theta(k(n)))$-*bounded* $\mathsf{QMA}_{\Theta(k)}(c, s)$ protocol for $c - s = 2^{-\mathcal{O}(k(n))}$, as desired. ◀

▶ **Lemma 20.** $\bigcup_{c-s \geq 2^{-\mathcal{O}(k)}} (\text{poly}, \mathcal{O}(k))$-*bounded* $\mathsf{QMA}_{\mathcal{O}(k)}(c, s) \subseteq \mathsf{BQ_USPACE}[k(n)]$.

Proof sketch. We only give a high level overview of the proof here; for the complete proof see Appendix B. The core of the proof is to develop and use new *space-efficient* QMA error reduction procedures. Our procedures are based on the "in-place" QMA amplification procedure of Marriott and Watrous [24], which allows the error in a QMA proof system to be reduced without requiring additional copies of the witness state. This was improved by Nagaj, Wocjan, and Zhang [25], whose phase-estimation based procedure reduces the error to 2^{-r} using only $\mathcal{O}\left(r \log \frac{1}{c-s}\right)$ additional qubits and $\mathcal{O}(r/(c-s))$ repetitions of the circuit and its inverse. We derive a procedure (Lemma 28) that gives the same error bounds while using only $\mathcal{O}\left(r + \log \frac{1}{c-s}\right)$ additional qubits, but still using only $\mathcal{O}(r/(c-s))$ repetitions of the circuit; the improved space bound will be required for our purposes[2].

Thus we can amplify the gap in our QMA protocols to still use $\mathcal{O}(k)$ space, but with completeness $1 - 2^{-\mathcal{O}(k)}$ and soundness $2^{-\mathcal{O}(k)}$. We can now replace the witness by the completely mixed state (or alternatively half of many EPR pairs), which gives us a computation with *no* witness such that the resulting completeness and soundness are both exponentially small, but are still separated by $2^{-\mathcal{O}(k)}$. Finally, we can once again apply our space-efficient amplification procedure to this witness-free protocol, obtaining a computation in $\mathsf{BQ_USPACE}[\mathcal{O}(k)]$. ◀

▶ **Lemma 21.** $\mathcal{O}(k(n))$-*Minimum Eigenvalue is* $\mathsf{BQ_USPACE}[k(n)]$-*hard under classical poly-time* $\mathcal{O}(k(n))$-*space reductions.*

[2] In recent work we improved this result to achieve such amplification using only $\log \frac{r}{c-s}$ extra space [15].

Proof sketch. Again we only give an overview; see the full version of the paper for the full proof. Recall that our uniformity condition on $k(n)$-*Minimum Eigenvalue* implies that every language in $k(n)$-*Minimum Eigenvalue* can be decided by a quantum circuit of size at most $2^{\mathcal{O}(k(n))}$. We first use our space-efficient error reduction procedure to amplify the gap; then we apply a variant of Kitaev's clock construction [23] to construct a Hamiltonian from this amplified circuit. We use a *binary* clock instead of a unary one to save space; since the number of gates is at most $2^{\mathcal{O}(k(n))}$, the clock only needs to be of size $\mathcal{O}(k(n))$, and the total dimension of the system is $2^{\mathcal{O}(k(n))}$ as required. Therefore the Hamiltonian is not local, but it remains sparse (with only a constant number of nonzero terms in each row). Kitaev's analysis then shows that we can obtain a gap inverse polynomial in the circuit size, or inverse exponential in $k(n)$. ◀

Proof of Theorems 17 and 18. Immediate from Lemmas 19, 20, and 21. ◀

Note the polynomial space case in Theorem 18 is Corollary 3, that PreciseQMA = PSPACE.

Finally, we end with two results particular to the polynomial space case. First of all, in the equality PreciseQMA = PSPACE, we can actually achieve perfect completeness ($c = 1$) for the QMA proof protocol, assuming the underlying gate set contains the Hadamard and Toffoli gates. Moreover for perfect completeness we do not require that $c - s > 2^{-\text{poly}}$:

▶ **Proposition 22.** *Let* $\mathsf{QMA}(c, s) = (\text{poly}, \text{poly})$*-bounded* $\mathsf{QMA}_{\text{poly}}(c, s)$*. Then*

$$\mathsf{PSPACE} = \mathsf{QMA}(1, 1 - 2^{-\text{poly}}) = \bigcup_{s<1} \mathsf{QMA}(1, s),$$

where we assume that the gateset we use contains the Hadamard and Toffoli gates. In the last term, the union is taken over all functions $s(n)$ such that $s(n) < 1$ for all n.

The containment $\mathsf{QMA}(1, s) \subseteq \mathsf{PSPACE}$ is known [19]. We give a proof in Appendix C.

Our second result concerns the QMA-complete Local Hamiltonian problem. We show that if we allow the promise gap to be exponentially small, then the problem becomes PSPACE-complete.

▶ **Definition 23** (*Precise k-Local Hamiltonian*)**.** Given as input is a k-local Hamiltonian $H = \sum_{j=1}^{r} H_j$ acting on n qubits, satisfying $r \in \text{poly}(n)$ and $\|H_j\| \leq \text{poly}(n)$, and numbers $a < b$ satisfying $b - a > 2^{-\text{poly}(n)}$. It is promised that the smallest eigenvalue of H is either at most a or at least b. Output 1 if the smallest eigenvalue of H is at most a, and output 0 otherwise.

▶ **Theorem 24.** *For any $3 \leq k \leq \mathcal{O}(\log(n))$, Precise k-Local Hamiltonian is* PreciseQMA*-complete, and hence* PSPACE*-complete.*

See Appendix D for a proof. Combined with the perfect completeness results of Appendix C, this will also give a proof that determining whether a local Hamiltonian is frustration-free is a PSPACE-complete problem (Theorem 35 in Appendix D).

5 Complete problems for time- and space-bounded classes

As we noted in the introduction, variants of the problems we consider are already known to be complete for other time-bounded quantum complexity classes. For example, consider the problem of inverting an efficiently encoded $2^{\mathcal{O}(k(n))} \times 2^{\mathcal{O}(k(n))}$ matrix with condition number at most $\kappa(n)$. If $\kappa(n), k(n) = \text{poly}(n)$, this problem is BQP-complete [18]. Theorem

13 says that this problem is instead $\mathsf{BQ_USPACE}[\mathcal{O}(k)]$-complete if $\kappa = 2^{\mathcal{O}(k)}$. Similarly, consider the problem of determining whether the minimum eigenvalue of an efficiently encoded $2^{\mathcal{O}(k(n))} \times 2^{\mathcal{O}(k(n))}$ matrix is at least b or at most a, with $b - a = g(n)$. If $g = 1/\operatorname{poly}$ and $k = \operatorname{poly}$ then this problem is QMA-complete [23, 2]. Theorem 17 says that this problem is instead $\mathsf{BQ_USPACE}[\mathcal{O}(k)]$-complete if $g = 2^{-\mathcal{O}(k)}$.

In both of the problems we consider, we have two parameters that we can vary: for matrix inversion, the condition number κ and the matrix size k; and for minimum eigenvalue, the promise gap size $g = b - a$ and the matrix size k. Varying these two parameters independently gives complete problems for quantum classes simultaneously bounded in time and space.

▶ **Theorem 25.** *Consider the class of problems solvable by a unitary quantum algorithm using* $\operatorname{poly}(T(n))$ *gates and* $\mathcal{O}(k(n))$ *space, where* $\Omega(\log(n)) \leq k(n) \leq T(n) \leq 2^{\mathcal{O}(k)} \leq 2^{\operatorname{poly}(n)}$. *This class has the following complete problem under classical* $\operatorname{poly}(n)$*-time and* $\mathcal{O}(k(n))$*-space reductions:*

Given as input is the size-n efficient encoding of a $2^{\mathcal{O}(k)} \times 2^{\mathcal{O}(k)}$ *positive semidefinite matrix* H *with a known upper bound* $\kappa = \operatorname{poly}(T)$ *on the condition number, so that* $\kappa^{-1} I \preceq H \preceq I$, *and* $s, t \in \{0,1\}^{k(n)}$. *It is promised that either* $|H^{-1}(s,t)| \geq b$ *or* $|H^{-1}(s,t)| \leq a$ *for some constants* $0 \leq a < b \leq 1$*; determine which is the case.*

▶ **Theorem 26.** *For functions* $k(n)$, $T(n)$ *satisfying* $\Omega(\log(n)) \leq k(n) \leq T(n) \leq 2^{\mathcal{O}(k)} \leq 2^{\operatorname{poly}(n)}$,

$$\bigcup_{c-s \geq \frac{1}{\operatorname{poly}(T)}} (\operatorname{poly}(n), \mathcal{O}(k))\text{-bounded } \mathsf{QMA}_{\mathcal{O}(k)}(c,s) = (\operatorname{poly}(T), \mathcal{O}(k))\text{-bounded } \mathsf{QMA}_{\mathcal{O}(k)}(2/3, 1/3)$$

Furthermore, the following problem is complete for this class under classical $\operatorname{poly}(n)$*-time and* $\mathcal{O}(k(n))$*-space reductions:*

Given as input is the size-n efficient encoding of a $2^{\mathcal{O}(k)} \times 2^{\mathcal{O}(k)}$ *PSD matrix* H, *such that* $\|H\|_{max} = \max_{s,t} |H(s,t)|$ *is at most a constant. Let* λ_{min} *be the minimum eigenvalue of* H. *It is promised that either* $\lambda_{min} \leq a$ *or* $\lambda_{min} \geq b$, *where* $a(n)$ *and* $b(n)$ *are numbers such that* $b - a \geq 1/\operatorname{poly}(T)$. *Output 1 if* $\lambda_{min} \leq a$, *and output 0 otherwise.*

We omit the proofs; they are straightforward generalizations of the proofs in our paper. These results interpolate between the time-bounded and space-bounded case: when $T = \operatorname{poly}(k)$ the time-bound dominates and we obtain a time-bounded class; while when $T = 2^{\mathcal{O}(k)}$ we obtain a space-bounded class. Note that when $T = 2^{\mathcal{O}(k)}$ then the complexity class in Theorem 26 is equal to $\mathsf{BQ_USPACE}[\mathcal{O}(k)]$, as shown in Theorem 18.

6 Open Problems

This work leaves open several questions that may lead to interesting follow-up work:

1. Can we use our $\mathsf{PreciseQMA} = \mathsf{PSPACE}$ result to prove upper or lower bounds for other complexity classes?
2. We have shown $\mathsf{PreciseQMA} = \mathsf{PSPACE}$. Ito, Kobayashi and Watrous have shown that QIP with doubly-exponentially small completeness-soundness gap is equal to EXP [19]. What about the power of QIP with exponentially small completeness-soundness gap?
3. In this paper we studied unitary quantum space complexity classes, and showed that $k(n)$-*Well-conditioned Matrix Inversion* and $k(n)$-*Minimum Eigenvalue* characterize unitary quantum space complexity. Can similar hardness results be shown for non-unitary quantum space complexity classes?

Acknowledgements. We are grateful to Andrew Childs, Sevag Gharibian, David Gosset, Aram Harrow, Hirotada Kobayashi, Robin Kothari, Tomoyuki Morimae, Harumichi Nishimura, Martin Schwarz, John Watrous, and Xiaodi Wu for helpful conversations, to John Watrous for comments on a preliminary draft, and to anonymous referees for suggestions.

References

1 Scott Aaronson. Quantum computing, postselection, and probabilistic polynomial-time. *Proceedings of the Royal Society A*, 461(2063):3473–3482, 2005.

2 Dorit Aharonov and Amnon Ta-Shma. Adiabatic quantum state generation and statistical zero knowledge. In *Proceedings of the 35th Annual ACM Symposium on the Theory of Computing (STOC)*, pages 20–29, 2003.

3 Eric W. Allender and Klaus W. Wagner. Counting hierarchies: polynomial time and constant depth circuits. In G. Rozenberg and A. Salomaa, editors, *Current trends in Theoretical Computer Science*, pages 469–483. World Scientific, 1993.

4 Sanjeev Arora and Boaz Barak. *Computational Complexity: A Modern Approach*. Cambridge University Press, New York, NY, USA, 2009.

5 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.

6 Dominic W. Berry, Andrew M. Childs, Richard Cleve, Robin Kothari, and Rolando D. Somma. Exponential improvement in precision for simulating sparse hamiltonians. In David B. Shmoys, editor, *Symposium on Theory of Computing, STOC 2014, New York, NY, USA, May 31 - June 03, 2014*, pages 283–292. ACM, 2014. doi:10.1145/2591796.2591854.

7 Dominic W. Berry, Andrew M. Childs, Richard Cleve, Robin Kothari, and Rolando D. Somma. Simulating Hamiltonian dynamics with a truncated Taylor series. *Physical Review Letters*, 114:090502, 2015.

8 Dominic W. Berry, Andrew M. Childs, and Robin Kothari. Hamiltonian simulation with nearly optimal dependence on all parameters. In *Proceedings of the 56th IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 792–809, 2015. URL: quant-ph/1501.01715.

9 Allan Borodin, Stephen A. Cook, and Nicholas Pippenger. Parallel computation for well-endowed rings and space-bounded probabilistic machines. *Information and Control*, 58(1-3):113–136, 1983. doi:10.1016/S0019-9958(83)80060-6.

10 Sergey Bravyi. Efficient algorithm for a quantum analogue of 2-sat. arXiv preprint quant-ph/0602108, 2006. URL: quant-ph/0602108.

11 Andrew Childs. On the relationship between continuous- and discrete-time quantum walk. *Communications in Mathematical Physics*, 294:581–603, 2010.

12 Stephen A. Cook. A taxonomy of problems with fast parallel algorithms. *Information and Control*, 64(1-3):2–21, 1985. doi:10.1016/S0019-9958(85)80041-3.

13 Dean Doron, Amir Sarid, and Amnon Ta-Shma. On approximating the eigenvalues of stochastic matrices in probabilistic logspace. Electronic Colloquium on Computational Complexity (ECCC) preprint TR16-120, 2016.

14 Dean Doron and Amnon Ta-Shma. On the problem of approximating the eigenvalues of undirected graphs in probabilistic logspace. In *Proceedings of the 42nd International Colloquium on Automata, Languages and Programming (ICALP)*, pages 419–431, 2015.

15 Bill Fefferman, Hirotada Kobayashi, Cedric Yen-Yu Lin, Tomoyuki Morimae, and Harumichi Nishimura. Space-efficient error reduction for unitary quantum computations. In *Proceedings of the 43rd International Colloquium on Automata, Languages and Programming (ICALP)*, pages 14:1–14:14, 2016.

16 François Le Gall. Solving Laplacian systems in logarithmic space. arXiv preprint 1608.01426, 2016.

17 David Gosset and Daniel Nagaj. Quantum 3-SAT is QMA1-complete. In *Proceedings of the 54th IEEE Annual Symposium on Foundations of Computer Science (FOCS)*, pages 756–765, 2013.

18 Aram W Harrow, Avinatan Hassidim, and Seth Lloyd. Quantum algorithm for linear systems of equations. *Physical Review letters*, 103(15):150502, 2009.

19 Tsuyoshi Ito, Hirotada Kobayashi, and John Watrous. Quantum interactive proofs with weak error bounds. In *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference (ITCS)*, pages 266–275, 2012.

20 Richard Jozsa, Barbara Kraus, Akimasa Miyake, and John Watrous. Matchgate and space-bounded quantum computations are equivalent. *Proceedings of the Royal Society A*, 466(2115):809–830, 2010. doi:10.1098/rspa.2009.0433.

21 Richard Jozsa and Akimasa Miyake. Matchgates and classical simulation of quantum circuits. *Proceedings of the Royal Society A*, 464(2100):3089–3106, 2008. doi:10.1098/rspa.2008.0189.

22 Julia Kempe and Oded Regev. 3-local Hamiltonian is QMA-complete. *Quantum Information & Computation*, 3(3):258–264, 2003.

23 A. Yu. Kitaev, A. H. Shen, and M. N. Vyalyi. *Classical and Quantum Computation*. American Mathematical Society, Boston, MA, USA, 2002.

24 Chris Marriott and John Watrous. Quantum arthur-merlin games. *Computational Complexity*, 14(2):122–152, 2005. doi:10.1007/s00037-005-0194-x.

25 Daniel Nagaj, Pawel Wocjan, and Yong Zhang. Fast amplification of QMA. *Quantum Information & Computation*, 9(11):1053–1068, 2011. URL: http://www.rintonpress.com/xxqic9/qic-9-1112/1053-1068.pdf.

26 M. A. Nielsen and I. L. Chuang. *Quantum Information and Computation*. Cambridge University Press, Cambridge, UK, 2000.

27 John H. Reif. Logarithmic depth circuits for algebraic functions. *SIAM J. Comput.*, 15(1):231–242, 1986. doi:10.1137/0215017.

28 Norbert Schuch, Michael M. Wolf, Frank Verstraete, and J. Ignacio Cirac. Computational complexity of projected entangled pair states. *Physical Review Letters*, 98:140506, 2007.

29 Yaoyun Shi. Both Toffoli and controlled-NOT need little help to do universal quantum computing. *Quantum Information & Computation*, 3(1):84–92, 2003. URL: http://dl.acm.org/citation.cfm?id=2011508.2011515.

30 Amnon Ta-Shma. Inverting well conditioned matrices in quantum logspace. 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 881–890. ACM, 2013. doi:10.1145/2488608.2488720.

31 Barbara M. Terhal and David P. DiVincenzo. Classical simulation of noninteracting-fermion quantum circuits. *Physical Review A*, 65(3):032325, 2002. doi:10.1103/PhysRevA.65.032325.

32 Leslie G. Valiant. Quantum circuits that can be simulated classically in polynomial time. *SIAM J. Comput.*, 31(4):1229–1254, 2002. doi:10.1137/S0097539700377025.

33 Dieter van Melkebeek and Thomas Watson. Time-space efficient simulations of quantum computations. *Theory of Computing*, 8:1–51, 2012.

34 F. Verstraete and J. I. Cirac. Renormalization algorithms for quantum-many body systems in two and higher dimensions. arXiv preprint cond-mat/0407066, 2004. URL: cond-mat/0407066.

35 John Watrous. Space-bounded quantum complexity. *J. Comput. Syst. Sci.*, 59(2):281–326, 1999. doi:10.1006/jcss.1999.1655.

36 John Watrous. On the complexity of simulating space-bounded quantum computations. *Computational Complexity*, 12(1):48–84, 2003.

37 John Watrous. Quantum computational complexity. In Robert A. Meyers, editor, *Encyclopedia of Complexity and Systems Science*, pages 7174–7201. Springer, 2009. doi:10.1007/978-0-387-30440-3_428.

A More details on space-bounded computation

For this section, it would be helpful to keep in mind that we always assume the space bound $k(n)$ always satisfies $\Omega(\log(n)) \leq k(n) \leq \text{poly}(n)$.

We start with the definitions of classical bounded space computation. In discussion of space-bounded classes, we usually consider Turing machines with two tapes, a read-only input tape and a work tape; only the space used on the work tape is counted. For $k : \mathbb{N} \to \mathbb{N}$, a function $f : \{0,1\}^* \to \{0,1\}^*$ is said to be computable in $k(n)$ space if any bit of $f(x)$ can be computed by a deterministic Turing machine using space $\mathcal{O}(k(|x|))$ on the work tape. For example, L is the class of functions that can be computed in $\mathcal{O}(\log n)$ space. We now discuss quantum space-bounded complexity classes; for a fuller discussion see [37]. A straightforward way to define quantum space-bounded classes is to consider a Turing machine with three tapes: a read-only classical input tape, a classical work tape, and a quantum work tape (with two heads) consisting of qubits. This is the model considered in [30] and [36], except that they allow intermediate measurements (and [36] allows even more general quantum operations). In this work we consider only computations with no intermediate measurements: we can therefore impose that there are no measurements on the quantum work tape until the register reaches a specified end state, following which a single measurement is performed on the quantum tape and the algorithm accepts or rejects according to the measurement. Therefore the operations performed by the algorithm will not depend on the quantum tape, since there is no way to read information out of it until the end of the algorithm.

Instead of working with Turing machines, in quantum computation it is much more customary (and convenient) to work with quantum circuits. For the setup above, since the operations on the quantum tape are completely classically controlled, we can equivalently consider a quantum circuit generated by a classical space-bounded Turing machine that computes the quantum gates one-by-one and applies them in sequence. If the classical Turing machine is $\mathcal{O}(k(n))$-space bounded, it has at most $2^{\mathcal{O}(k)}$ configurations, and therefore there are at most $2^{\mathcal{O}(k)}$ quantum gates in the circuit.

Moreover, the $\mathcal{O}(k)$-space bounded classical Turing machine can be replaced by a classical circuit on $\mathcal{O}(k)$ bits, such that there is a $\text{poly}(n)$-time $\mathcal{O}(k)$-space Turing machine that on input i generates the i-th gate of the circuit (see e.g. [4, Section 6.8]). The classical circuit can then be bundled into the quantum circuit, and we obtain a quantum circuit with at most $2^{\mathcal{O}(k)}$ gates, such that each individual gate can be generated in classical $\text{poly}(n)$-time and $\mathcal{O}(k)$-space. This justifies the definition of the complexity class $\mathsf{Q_U SPACE}[k(n)](c,s)$:

▶ **Definition 27.** Let $k(n)$ be a function satisfying $\Omega(\log(n)) \leq k(n) \leq \text{poly}(n)$. A promise problem $L = (L_{yes}, L_{no})$ is in $\mathsf{Q_U SPACE}[k(n)](c,s)$ if there exists a $\text{poly}(|x|)$-time $\mathcal{O}(k)$-space uniformly generated family of quantum circuits $\{Q_x\}_{x\in\{0,1\}^*}$, where each circuit $Q_x = U_{x,T}U_{x,T-1}\cdots U_{x,1}$ has $T = 2^{\mathcal{O}(k)}$ gates, and acts on $\mathcal{O}(k(|x|))$ qubits, such that:
If $x \in L_{yes}$:

$$\left\langle 0^k \middle| Q_x^\dagger |1\rangle\langle 1|_{out} Q_x \middle| 0^k \right\rangle \geq c. \tag{10}$$

Whereas if $x \in L_{no}$:

$$\left\langle 0^k \right| Q_x^\dagger |1\rangle\langle 1|_{out} Q_x \left| 0^k \right\rangle \leq s. \tag{11}$$

Here *out* denotes a single qubit we measure at the end of the computation; no intermediate measurements are allowed. Furthermore, we require c and s to be computable in classical $\mathcal{O}(k(n))$-space.

B Proof of Lemma 20

B.1 In-place gap amplification of QMA protocols with phase estimation

We start out by proving the following lemma, which proves "in-place" gap amplification of QMA using phase estimation (see also the similar result of Nagaj et. al, Lemma 36 in Appendix E).

▶ **Lemma 27.** *For any functions $t, k, r > 0$,*

$$(t,k)\text{-bounded } \mathsf{QMA}_m(c,s) \subseteq \left(\mathcal{O}\left(\frac{t2^r}{c-s}\right), \mathcal{O}\left(k + r + \log\left(\frac{1}{c-s}\right)\right)\right)\text{-bounded } \mathsf{QMA}_m(1-2^{-r}, 2^{-r}).$$

Proof. Let $L = (L_{yes}, L_{no})$ be a promise problem in $\mathsf{QMA}(c,s)$ and $\{V_x\}_{x\in\{0,1\}^*}$ the corresponding uniform family of verification circuits. Define the projectors:

$$\Pi_0 = I_m \otimes \left|0^k\right\rangle\left\langle 0^k\right|, \quad \Pi_1 = V_x^\dagger \left(|1\rangle\langle 1|_{out} \otimes I_{m+k-1}\right) V_x \tag{12}$$

and the corresponding reflections $R_0 = 2\Pi_0 - I, R_1 = 2\Pi_1 - I$. Define $\phi_c = \arccos\sqrt{c}/\pi$ and $\phi_s = \arccos\sqrt{s}/\pi$ (recalling that these functions can be computed to precision $\mathcal{O}(c-s)$ in space $\mathcal{O}(\log[1/(c-s)])$. Now consider the following procedure:

1. Perform phase estimation of the operator R_1R_0 on the state $|\psi\rangle \otimes \left|0^k\right\rangle$, with precision $\mathcal{O}(c-s)$ and failure probability 2^{-r}.
2. Output YES if the phase is at most $(\phi_c + \phi_s)/2$; otherwise output NO.

Phase estimation of an operator U up to precision a and failure probability ϵ requires $\alpha := \lceil \log_2(1/a) \rceil + \log_2[2 + 1/(2\epsilon)]$ additional ancilla qubits and $2^\alpha = \mathcal{O}(1/(a\epsilon))$ applications of the control-U operation (see e.g. [26]). Thus, the above procedure can be implemented by a circuit of size $\mathcal{O}(2^r t/(c-s))$ using $\mathcal{O}(r + \log[1/(c-s)])$ extra ancilla qubits. Using the standard analysis of in-place QMA error reduction [24, 25], it can be shown that this procedure has completeness probability at least $1 - 2^{-r}$ and soundness at most 2^{-r}. ◀

In Appendix E we will prove the following stronger error reduction lemma that gives the same space bound but uses less time. This better time bound will be required for proving Lemma 21.

▶ **Lemma 28.** *For any functions $t, k, r > 0$,*

$$(t,k)\text{-bounded } \mathsf{QMA}_m(c,s) \subseteq \left(\mathcal{O}\left(\frac{rt}{c-s}\right), \mathcal{O}\left(k + r + \log\left(\frac{1}{c-s}\right)\right)\right)\text{-bounded } \mathsf{QMA}_m(1-2^{-r}, 2^{-r}).$$

Thus, we get the following corollaries:

▶ **Corollary 29.** *For any $r = \mathcal{O}(k)$, $\mathsf{QUSPACE}[k](c, c - 2^{-\mathcal{O}(k)}) \subseteq \mathsf{QUSPACE}[\Theta(k)](1 - 2^{-r}, 2^{-r})$.*

This corollary shows that error reduction is possible for unitary quantum $\mathcal{O}(k)$-space bounded classes, as long as the completeness-soundness gap is at least $2^{-\mathcal{O}(k)}$.

Proof. This follows from Lemma 28 by taking $m = 0$, $s = c - 2^{-\Theta(k)}$, and $r = \Theta(k)$. ◀

▶ **Corollary 30.**

(t,k)-*bounded* $\mathsf{QMA}_m(c, c-2^{-\Theta(k)}) \subseteq \left(\mathcal{O}\left(t2^{\Theta(k)}\right), \mathcal{O}(k)\right)$-*bounded* $\mathsf{QMA}_m(1-2^{-(m+2)}, 2^{-(m+2)})$.

Proof. This follows from Lemma 28 by taking $s = c - 2^{-\Theta(k)}$ and $r = m+2$. ◀

B.2 Removing the witness of an amplified QMA protocol

▶ **Theorem 31.** *For any function* $t = 2^{\mathcal{O}(k+m)}$,

$$(t,k)\text{-bounded } \mathsf{QMA}_m(1-2^{-(m+2)}, 2^{-(m+2)}) \subseteq \mathsf{Q_U SPACE}[k+m](3/4 \cdot 2^{-m}, 1/4 \cdot 2^{-m}).$$

Proof. The proof is very similar to that of [24, Theorem 3.6]. For any functions m, k, consider a problem $L \in (t,k)$-*bounded* $\mathsf{QMA}_m(1-2^{-(m+2)}, 2^{-(m+2)})$, and let $\{V'_x\}_{x\in\{0,1\}^*}$ be a uniform family of verification circuits for L with completeness $1-2^{-(m+2)}$ and soundness $2^{-(m+2)}$.

For convenience, define the $2^m \times 2^m$ matrix:

$$Q_x := \left(I_{2^m} \otimes \langle 0^p|\right) V_x'^{\dagger} |1\rangle\langle 1|_{out} V'_x \left(I_{2^m} \otimes |0^p\rangle\right). \tag{13}$$

Q_x is positive semidefinite, and $\langle\psi|Q_x|\psi\rangle$ is the acceptance probability of V'_x on witness ψ. Thus

$$x \in L_{yes} \Rightarrow \mathrm{tr}[Q_x] \geq 1 - 2^{-(m+2)} \geq 3/4 \tag{14}$$

since the trace is at least the largest eigenvalue, and $m \geq 0$; likewise,

$$x \in L_{no} \Rightarrow \mathrm{tr}[Q_x] \leq 2^m \cdot 2^{-(m+2)} = 1/4 \tag{15}$$

since the trace is the sum of the 2^m eigenvalues, each of which is at most $2^{-(m+2)}$.

Therefore our problem reduces to determining whether the trace of Q_x is at least $3/4$ or at most $1/4$. Now we show that using the totally mixed state $2^{-m}I_m$ (alternatively, preparing m EPR pairs and taking a qubit from each pair) as the witness of the verification procedure encoded by Q_x, succeeds with the desired completeness and soundness bounds. The acceptance probability is given by $\mathrm{tr}(Q_x 2^{-m} I_m) = 2^{-m}\,\mathrm{tr}(Q_x)$, which is at least $2^{-m}\cdot 3/4$ if $x \in L_{yes}$, and at most $2^{-m} \cdot 1/4$ if $x \in L_{no}$. Thus we have reduced the problem L to determining if a quantum computation with *no* witness, acting on $k+m$ qubits, accepts with probability at least $3/4 \cdot 2^{-m}$ or at most $s' = 1/4 \cdot 2^{-m}$. ◀

We can finally finish the proof of Lemma 20.

Proof of Lemma 20. This follows from Corollary 30, Theorem 31, and Corollary 29. ◀

C Achieving Perfect Completeness for PreciseQMA

We now consider the problem of achieving perfect completeness for PreciseQMA. Specifically, we will show the following:

▶ **Proposition 32.** *Let* $\mathsf{QMA}(c,s) = (\mathrm{poly}, \mathrm{poly})$-*bounded* $\mathsf{QMA}_{\mathrm{poly}}(c,s)$. *Then*

$$\mathsf{PSPACE} = \mathsf{QMA}(1, 1-2^{-\mathrm{poly}}) = \bigcup_{s<1} \mathsf{QMA}(1,s),$$

where we assume that the gateset we use contains the Hadamard and Toffoli gates. In the last term, the union is taken over all functions $s(n)$ *such that* $s(n) < 1$ *for all* n.

Since PSPACE = PreciseQMA, this proposition shows that any PreciseQMA protocol can be reduced to a different PreciseQMA protocol with perfect completeness, i.e. in the YES case Arthur accepts Merlin's witness with probability 1. The reduction is rather roundabout, however, and it would be interesting to see if a more direct reduction can be found.

The second equality follows from the first equality and the result by [19] that $\mathsf{QMA}(1,s) \subseteq \mathsf{PSPACE}$. We will therefore only prove the first equality.

Looking back at Circuit 8, we see that we *almost* have perfect completeness in our protocol already - if the Hamiltonian simulation of e^{-iHt} could be done without error, then indeed the protocol has perfect completeness. Our strategy will be perform a different unitary that can be performed exactly, but, like e^{-iHt}, also allows us to use phase estimation to distinguish the eigenvalues of H.

Given a sparse Hamiltonian H (with at most d nonzero entries per row) and a number $X \geq \max_{j,\ell} |H_{j\ell}|$ that upper bounds the absolute value of entries of H, Andrew Childs defined an efficiently implementable quantum walk [6, 11]. Each step of the quantum walk is a unitary U with eigenvalues $e^{i\tilde{\lambda}}$, where

$$\tilde{\lambda} = \arcsin \frac{\lambda}{Xd} \tag{16}$$

with λ representing eigenvalues of H. Note that the YES case $\lambda = 0$ corresponds to $\tilde{\lambda} = 0$, and the NO case $\lambda \geq 2^{-g(n)}$ corresponds to $\tilde{\lambda} \geq 2^{-g(n)}/(Xd)$ since $\arcsin x \geq x$ for $|x| \leq 1$. In the latter case the $\tilde{\lambda}$ can be at most exponentially small, and therefore the stripped down version of phase estimation still suffices to tell the two cases apart with exponentially small probability.

We now note that the Hamiltonian H we obtain from the hardness reduction from PSPACE (Lemma 21) is of a very special form. Specifically, since $\mathsf{BQ_USPACE}[\mathrm{poly}] = \mathsf{PSPACE}$, we can assume the verifier circuit V_x is deterministic, so it has completeness 1 and soundness 0. Moreover, all of its gates are classical, and hence all entries of the Kitaev clock Hamiltonian H are 0, $\pm 1/2$, or 1.

For the matrix H satisfying the above, U can be implemented exactly with a standard gateset; perfect completeness of the protocol will then follow. If H is a $N \times N$ matrix (where $N = 2^n$), U is (see presentation in [8, Section 3.1 and Lemma 10]) a unitary defined on the enlarged Hilbert space $\mathbb{C}^{2N} \otimes \mathbb{C}^{2N} = (\mathbb{C}^N \otimes \mathbb{C}^2) \otimes (\mathbb{C}^N \otimes \mathbb{C}^2)$, as follows:

$$U = ST(I_{2N} \otimes (I_{2N} - 2|0\rangle\langle 0|_{2N}))T^\dagger \tag{17}$$

where the $2N$ subscript indicates a register of dimension $2N$, the unitary S swaps the two registers, and the unitary T is defined by

$$T = \sum_{j=0}^{N-1} \sum_{b \in \{0,1\}} (|j\rangle\langle j| \otimes |b\rangle\langle b|) \otimes |\varphi_{jb}\rangle\langle 0|_{2N} \tag{18}$$

with $|\varphi_{j1}\rangle = |0\rangle_N |1\rangle$ and

$$|\varphi_{jb}\rangle = \frac{1}{\sqrt{d}} \sum_{\ell \in F_j} |\ell\rangle \left(\sqrt{\frac{H^*_{j\ell}}{X}} |0\rangle + \sqrt{1 - \frac{|H^*_{j\ell}|}{X}} |1\rangle \right), \tag{19}$$

where F_j index the nonzero entries in the j-th row. Recall that for any j, ℓ, $H_{j\ell} = 0, \pm 1/2$, or 1, and hence we can take $X = 1$. If we furthermore assume d is a power of 2 (which we can always do by adding indices of zero entries to F_j), it is straightforward to see that

both S and T can be implemented using just Hadamard gates and classical gates (Pauli-X, controlled-X, and Toffoli gates) - the latter of which can be implemented using just Toffoli gates and access to a qubit in the $|1\rangle$ state (which can be provided by the prover). Therefore U can be exactly implemented in any gateset that allows Hadamard gates and Toffoli gates to be implemented exactly.

D Precise Local Hamiltonian Problem

▶ **Definition 32** (*Precise k-Local Hamiltonian*)**.** Given as input is a k-local Hamiltonian $H = \sum_{j=1}^{r} H_j$ acting on n qubits, satisfying $r \in \text{poly}(n)$ and $\|H_j\| \leq \text{poly}(n)$, and numbers $a < b$ satisfying $b - a > 2^{-\text{poly}(n)}$. It is promised that the smallest eigenvalue of H is either at most a or at least b. Output 1 if the smallest eigenvalue of H is at most a, and output 0 otherwise.

We then have the following theorem:

▶ **Theorem 24.** *For any $3 \leq k \leq \mathcal{O}(\log(n))$, Precise k-Local Hamiltonian is* PreciseQMA*-complete, and hence* PSPACE*-complete.*

Proof. This proof follows straightforwardly by adapting the proof of [23] and [22]. The proof of containment in PreciseQMA is identical to the containment of the usual Local Hamiltonian problem in QMA; see [23] for details.

To show PreciseQMA-hardness, we note that for a QMA-verification procedure with T gates, completeness c and soundness s, [22] reduces this to a 3-local Hamiltonian with lowest eigenvalue no more than $(1-c)/(T+1)$ in the YES case, or no less than $(1-s)/T^3$ in the NO case. For this to specify a valid *Precise Local Hamiltonian* problem we need that

$$\frac{1-s}{T^3} - \frac{1-c}{T+1} > 2^{-\text{poly}(n)}. \tag{20}$$

Recalling that we showed that perfect completeness can be assumed for PreciseQMA-hard problems, we can take $c = 1$, $s = 1 - 2^{-\text{poly}(n)}$ and the above inequality trivially holds. Hence any problem in PSPACE can be reduced to a *Precise* 3*-Local Hamiltonian* problem. ◀

In fact, even just testing if a k-Local Hamiltonian is frustration-free is PSPACE-complete:

▶ **Definition 34** (*Frustration-Free k-Local Hamiltonian*)**.** Given as input is a k-local Hamiltonian $H = \sum_{j=1}^{r} H_j$ acting on n qubits, satisfying $r \in \text{poly}(n)$, each term H_j is positive semidefinite, and $\|H_j\| \leq \text{poly}(n)$. Output 1 if the smallest eigenvalue of H is zero, and output 0 otherwise.

▶ **Theorem 35.** *Frustration-Free k-Local Hamiltonian is* PSPACE*-complete.*

Proof. The containment in PSPACE follows from the proof of the containment of the usual Local Hamiltonian problem in QMA [23], along with Proposition 22. PSPACE-hardness follows from the proof of Theorem 24, by taking $c = 1$ in the proof. ◀

E In-place gap amplification

In this appendix we will prove Lemma 28. To do so we first start out with the following weaker result:

▶ **Lemma 36** (Implicit in Nagaj, Wocjan, and Zhang [25]). *For any functions $t, k, r > 0$,*

$$(t,k)\text{-bounded } \mathsf{QMA}_m(c,s) \subseteq \left(\mathcal{O}\left(\frac{rt}{c-s}\right), \mathcal{O}\left(k + r\log\left(\frac{1}{c-s}\right)\right)\right)\text{-bounded } \mathsf{QMA}_m(1-2^{-r}, 2^{-r}).$$

Proof. Let $L = (L_{yes}, L_{no})$ be a promise problem in $\mathsf{QMA}(c,s)$ and $\{V_x\}_{x\in\{0,1\}^*}$ the corresponding uniform family of verification circuits. Define the projectors:

$$\Pi_0 = I_m \otimes \left|0^k\right\rangle\!\left\langle 0^k\right|, \quad \Pi_1 = V_x^\dagger \left(|1\rangle\!\langle 1|_{out} \otimes I_{m+k-1}\right) V_x \tag{21}$$

and the corresponding reflections:

$$R_0 = 2\Pi_0 - I, \quad R_1 = 2\Pi_1 - I. \tag{22}$$

Define $\phi_c = \arccos\sqrt{c}/\pi$ and $\phi_s = \arccos\sqrt{s}/\pi$ (recalling that these functions can be computed to precision $\mathcal{O}(c-s)$ in space $\mathcal{O}(\log[1/(c-s)])$. Consider the following procedure:

1. Perform r trials of phase estimation of the operator R_1R_0 on the state $|\psi\rangle \otimes \left|0^k\right\rangle$, with precision $\mathcal{O}(c-s)$ and $1/16$ failure probability.
2. If the median of the r results is at most $(\phi_c+\phi_s)/2$, output YES; otherwise output NO.

Phase estimation of an operator U up to precision a and failure probability ϵ requires $\alpha := \lceil \log_2(1/a)\rceil + \log_2[2 + 1/(2\epsilon)]$ additional ancilla qubits and $2^\alpha = \mathcal{O}(1/(a\epsilon))$ applications of the control-U operation (see e.g. [26]). Thus, the above procedure, which uses r applications of phase estimation to precision $\mathcal{O}(c-s)$, can be implemented by a circuit of size $\mathcal{O}(rt/(c-s))$ using $\mathcal{O}(r\log[1/(c-s)])$ extra ancilla qubits. Using the standard analysis of in-place QMA error reduction [24, 25], it can be seen that this procedure has completeness probability at least $1-2^{-r}$ and soundness at most 2^{-r}. ◀

We can now prove Lemma 28, which we restate below:

▶ **Lemma 28.** *For any functions $t, k, r > 0$,*

$$(t,k)\text{-bounded } \mathsf{QMA}_m(c,s)r \subseteq \left(\mathcal{O}\left(\frac{rt}{c-s}\right), \mathcal{O}\left(k + r + \log\left(\frac{1}{c-s}\right)\right)\right)\text{-bounded } \mathsf{QMA}_m(1-2^{-r}, 2^{-r}).$$

Proof.

$$\begin{aligned}(t,k)\text{-bounded } \mathsf{QMA}_m(c,s) &\subseteq \left(\mathcal{O}\left(\frac{t}{c-s}\right), \mathcal{O}\left(k + \log\left(\frac{1}{c-s}\right)\right)\right)\text{-bounded } \mathsf{QMA}_m(3/4, 1/4)\\ &\subseteq \left(\mathcal{O}\left(\frac{rt}{c-s}\right), \mathcal{O}\left(k + r + \log\left(\frac{1}{c-s}\right)\right)\right)\text{-bounded } \mathsf{QMA}_m(1-2^{-r}, 2^{-r})\end{aligned}$$

where the first line follows by taking $r = 2$ in Lemma 27, and the second line uses Lemma 36. ◀

F Proof sketch of $\mathsf{PQP}^{O_{PEPS}}_{\|,\text{classical}} = \mathsf{PP}$

Since $\mathsf{PP} \subseteq \mathsf{BQP}^{\mathsf{O_{PEPS}}}_{\|,\text{classical}} \subseteq \mathsf{PQP}^{\mathsf{O_{PEPS}}}_{\|,\text{classical}}$ [28], we only need to show that $\mathsf{PQP}^{O_{PEPS}}_{\|,\text{classical}} \subseteq \mathsf{PP}$. In [28] it was noted that all PEPS can be seen as the output of a quantum circuit followed by a postselected measurement. Therefore $\mathsf{PQP}^{O_{PEPS}}_{\|,\text{classical}}$ corresponds to the problems that can be decided by a quantum circuit, followed by a postselected measurement (since the queries to O_{PEPS} are classical and nonadaptive, we can compose them into one single postselection),

followed by a measurement. In the YES case the measurement outputs 1 with probability at least c, whereas in the NO case the measurement outputs 1 with probability at most s, with $c > s$. The standard counting argument placing BQP inside PP then applies to this case as well; see for instance [1, Propositions 2 and 3].

Matrix Completion and Related Problems via Strong Duality*†

Maria-Florina Balcan[1], Yingyu Liang[2], David P. Woodruff[3], and Hongyang Zhang[4]

1 **Carnegie Mellon University, Pittsburgh, USA**
ninamf@cs.cmu.edu
2 **University of Wisconsin-Madison, Madison, USA**
yliang@cs.wisc.edu
3 **Carnegie Mellon University, Pittsburgh, USA**
dwoodruf@cs.cmu.edu
4 **Carnegie Mellon University, Pittsburgh, USA**
hongyanz@cs.cmu.edu

Abstract

This work studies the *strong duality of non-convex matrix factorization problems*: we show that under certain dual conditions, these problems and its dual have the same optimum. This has been well understood for convex optimization, but little was known for non-convex problems. We propose a novel analytical framework and show that under certain dual conditions, the optimal solution of the matrix factorization program is the same as its bi-dual and thus the global optimality of the non-convex program can be achieved by solving its bi-dual which is convex. These dual conditions are satisfied by a wide class of matrix factorization problems, although matrix factorization problems are hard to solve in full generality. This analytical framework may be of independent interest to non-convex optimization more broadly.

We apply our framework to two prototypical matrix factorization problems: matrix completion and robust Principal Component Analysis (PCA). These are examples of efficiently recovering a hidden matrix given limited reliable observations of it. Our framework shows that exact recoverability and strong duality hold with nearly-optimal sample complexity guarantees for matrix completion and robust PCA.

1998 ACM Subject Classification G.1.6 Optimization

Keywords and phrases Non-Convex Optimization, Strong Duality, Matrix Completion, Robust PCA, Sample Complexity

Digital Object Identifier 10.4230/LIPIcs.ITCS.2018.5

1 Introduction

Non-convex matrix factorization problems have been an emerging object of study in theoretical computer science [37, 30, 53, 45], optimization [58, 50], machine learning [11, 23, 21, 36, 42, 57], and many other domains. In theoretical computer science and optimization, the study of such models has led to significant advances in provable algorithms that converge to local

* A full version of the paper is available at https://arxiv.org/abs/1704.08683

† This work was supported in part by NSF grants NSF CCF-1422910, NSF CCF-1535967, NSF CCF-1451177, NSF IIS-1618714, NSF CCF-1527371, a Sloan Research Fellowship, a Microsoft Research Faculty Fellowship, DMS-1317308, Simons Investigator Award, Simons Collaboration Grant, and ONR-N00014-16-1-2329.

minima in linear time [37, 30, 53, 2, 3]. In machine learning, matrix factorization serves as a building block for large-scale prediction and recommendation systems, e.g., the winning submission for the Netflix prize [41]. Two prototypical examples are matrix completion and robust Principal Component Analysis (PCA).

This work develops a novel framework to analyze a class of non-convex matrix factorization problems with strong duality, which leads to exact recoverability for matrix completion and robust Principal Component Analysis (PCA) via the solution to a convex problem. The matrix factorization problems can be stated as finding a target matrix $\mathbf{X}^*$ in the form of $\mathbf{X}^* = \mathbf{AB}$, by minimizing the objective function $H(\mathbf{AB}) + \frac{1}{2}\|\mathbf{AB}\|_F^2$ over factor matrices $\mathbf{A} \in \mathbb{R}^{n_1 \times r}$ and $\mathbf{B} \in \mathbb{R}^{r \times n_2}$ with a known value of $r \ll \min\{n_1, n_2\}$, where $H(\cdot)$ is some function that characterizes the desired properties of $\mathbf{X}^*$.

Our work is motivated by several promising areas where our analytical framework for non-convex matrix factorizations is applicable. The first area is low-rank matrix completion, where it has been shown that a low-rank matrix can be exactly recovered by finding a solution of the form $\mathbf{AB}$ that is consistent with the observed entries (assuming that it is incoherent) [37, 53, 23]. This problem has received a tremendous amount of attention due to its important role in optimization and its wide applicability in many areas such as quantum information theory and collaborative filtering [30, 61, 7]. The second area is robust PCA, a fundamental problem of interest in data processing that aims at recovering both the low-rank and the sparse components exactly from their superposition [13, 43, 27, 62, 61, 59], where the low-rank component corresponds to the product of $\mathbf{A}$ and $\mathbf{B}$ while the sparse component is captured by a proper choice of function $H(\cdot)$, e.g., the ℓ_1 norm [13, 6]. We believe our analytical framework can be potentially applied to other non-convex problems more broadly, e.g., matrix sensing [54], dictionary learning [52], weighted low-rank approximation [45, 42], and deep linear neural network [39], which may be of independent interest.

Without assumptions on the structure of the objective function, direct formulations of matrix factorization problems are NP-hard to optimize in general [31, 60]. With standard assumptions on the structure of the problem and with sufficiently many samples, these optimization problems can be solved efficiently, e.g., by convex relaxation [14, 18]. Some other methods run local search algorithms given an initialization close enough to the global solution in the basin of attraction [37, 30, 53, 21, 38]. However, these methods have sample complexity significantly larger than the information-theoretic lower bound; see Table 1 for a comparison. The problem becomes more challenging when the number of samples is small enough that the sample-based initialization is far from the desired solution, in which case the algorithm can run into a local minimum or a saddle point.

Another line of work has focused on studying the loss surface of matrix factorization problems, providing positive results for approximately achieving global optimality. One nice property in this line of research is that there is no spurious local minima for specific applications such as matrix completion [23], matrix sensing [11], dictionary learning [52], phase retrieval [51], linear deep neural networks [39], etc. However, these results are based on concrete forms of objective functions. Also, even when any local minimum is guaranteed to be globally optimal, in general it remains NP-hard to escape high-order saddle points [5], and additional arguments are needed to show the achievement of a local minimum. Most importantly, all existing results rely on strong assumptions on the sample size.

1.1 Our Results

Our work studies the exact recoverability problem for a variety of non-convex matrix factorization problems. The goal is to provide a unified framework to analyze a large class

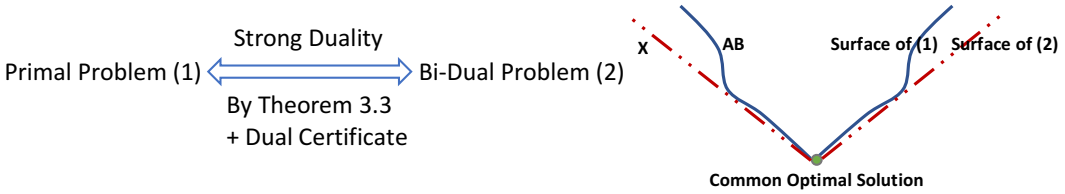


Figure 1 Strong duality of matrix factorizations.

of matrix factorization problems, and to achieve efficient algorithms. Our main results show that although matrix factorization problems are hard to optimize in general, *under certain dual conditions the duality gap is zero*, and thus the problem can be converted to an equivalent convex program. The main theorem of our framework is the following.

Theorem 4. (Strong Duality. Informal.) *Under certain dual conditions, strong duality holds for the non-convex optimization problem*

$$(\widetilde{\mathbf{A}}, \widetilde{\mathbf{B}}) = \operatorname*{argmin}_{\mathbf{A}\in\mathbb{R}^{n_1\times r}, \mathbf{B}\in\mathbb{R}^{r\times n_2}} F(\mathbf{A},\mathbf{B}) = H(\mathbf{AB}) + \frac{1}{2}\|\mathbf{AB}\|_F^2, \quad H(\cdot) \text{ is convex and closed}, \tag{1}$$

where "the function $H(\cdot)$ is closed" means that for each $\alpha \in \mathbb{R}$, the sub-level set $\{\mathbf{X} \in \mathbb{R}^{n_1\times n_2} : H(\mathbf{X}) \le \alpha\}$ is a closed set. In other words, problem (1) and its bi-dual problem

$$\widetilde{\mathbf{X}} = \operatorname*{argmin}_{\mathbf{X}\in\mathbb{R}^{n_1\times n_2}} H(\mathbf{X}) + \|\mathbf{X}\|_{r*}, \tag{2}$$

have exactly the same optimal solutions in the sense that $\widetilde{\mathbf{A}}\widetilde{\mathbf{B}} = \widetilde{\mathbf{X}}$, where $\|\mathbf{X}\|_{r*}$ is a convex function defined by $\|\mathbf{X}\|_{r*} = \max_{\mathbf{M}} \langle \mathbf{M}, \mathbf{X}\rangle - \frac{1}{2}\|\mathbf{M}\|_r^2$ and $\|\mathbf{M}\|_r^2 = \sum_{i=1}^r \sigma_i^2(\mathbf{M})$ is the sum of the first r largest squared singular values.

Theorem 4 connects the non-convex program (1) to its convex counterpart via strong duality; see Figure 1. We mention that strong duality rarely happens in the non-convex optimization region: low-rank matrix approximation [44] and quadratic optimization with two quadratic constraints [10] are among the few paradigms that enjoy such a nice property. Given strong duality, the computational issues of the original problem can be overcome by solving the convex bi-dual problem (2).

The positive result of our framework is complemented by a lower bound to formalize the hardness of the above problem in general. Assuming that the random 4-SAT problem is hard [45], we give a strong negative result for deterministic algorithms. If also $\mathsf{BPP} = \mathsf{P}$ (see Section 6 for a discussion), then the same conclusion holds for randomized algorithms succeeding with probability at least $2/3$.

Theorem 9. (Hardness Statement. Informal.) *Assuming that random 4-SAT is hard on average, there is a problem in the form of (1) such that any deterministic algorithm achieving $(1+\epsilon)\mathsf{OPT}$ in the objective function value with $\epsilon \le \epsilon_0$ requires $2^{\Omega(n_1+n_2)}$ time, where OPT is the optimum and $\epsilon_0 > 0$ is an absolute constant. If $\mathsf{BPP} = \mathsf{P}$, then the same conclusion holds for randomized algorithms succeeding with probability at least $2/3$.*

Our framework only requires the dual conditions in Theorem 4 to be verified. We will show that two prototypical problems, matrix completion and robust PCA, obey the conditions. They belong to the linear inverse problems of form (1) with a proper choice of function $H(\cdot)$, which aim at exactly recovering a hidden matrix $\mathbf{X}^*$ with $\mathsf{rank}(\mathbf{X}^*) \le r$ given a limited number of linear observations of it.

For matrix completion, the linear measurements are of the form $\{\mathbf{X}^*_{ij} : (i,j) \in \Omega\}$, where Ω is the support set which is uniformly distributed among all subsets of $[n_1] \times [n_2]$

Table 1 Comparison of matrix completion methods. Here $\kappa = \sigma_1(\mathbf{X}^*)/\sigma_r(\mathbf{X}^*)$ is the condition number of $\mathbf{X}^* \in \mathbb{R}^{n_1 \times n_2}$, ϵ is the accuracy such that the output $\widetilde{\mathbf{X}}$ obeys $\|\widetilde{\mathbf{X}} - \mathbf{X}^*\|_F \leq \epsilon$, $n_{(1)} = \max\{n_1, n_2\}$ and $n_{(2)} = \min\{n_1, n_2\}$. *The first line of ours is an information-theoretic upper bound and the second line is a polynomial-time approach.*

Work	Sample Complexity	μ-Incoherence
[37]	$\mathcal{O}\left(\kappa^4\mu^2 r^{4.5} n_{(1)} \log n_{(1)} \log\left(\frac{r\|\mathbf{X}^*\|_F}{\epsilon}\right)\right)$	Condition (3)
[30]	$\mathcal{O}\left(\mu r n_{(1)}(r + \log\left(\frac{n_{(1)}\|\mathbf{X}^*\|_F}{\epsilon}\right)\frac{\|\mathbf{X}^*\|_F^2}{\sigma_r^2}\right)$	Condition (3)
[23]	$\mathcal{O}(\max\{\mu^6\kappa^{16}r^4, \mu^4\kappa^4 r^6\} n_{(1)} \log^2 n_{(1)})$	$\|\mathbf{X}^*_{i\cdot}\|_2 \leq \frac{\mu}{\sqrt{n_{(2)}}}\|\mathbf{X}^*\|_F$
[53]	$\mathcal{O}(r n_{(1)}\kappa^2 \max\left\{\mu \log n_{(2)}, \sqrt{\frac{n_{(1)}}{n_{(2)}}}\mu^2 r^6 \kappa^4\right\}$	Condition (3)
[65]	$\mathcal{O}(\mu r^2 n_{(1)}\kappa^2 \max(\mu, \log n_{(1)}))$	Condition (3)
[20]	$\mathcal{O}\left(\left(\mu^2 r^4\kappa^2 + \mu r \log\left(\frac{\|\mathbf{X}^*\|_F}{\epsilon}\right)\right) n_{(1)} \log\left(\frac{\|\mathbf{X}^*\|_F}{\epsilon}\right)\right)$	Condition (3)
[64]	$\mathcal{O}\left(\mu r^3 n_{(1)} \log n_{(1)} \log\left(\frac{1}{\epsilon}\right)\right)$	Condition (3)
[40]	$\mathcal{O}\left(n_{(2)} r \sqrt{\frac{n_{(1)}}{n_{(2)}}}\kappa^2 \max\left\{\mu \log n_{(2)}, \mu^2 r \sqrt{\frac{n_{(1)}}{n_{(2)}}}\kappa^4\right\}\right)$	Similar to (3) and (12)
[17]	$\mathcal{O}(\max\{\mu\kappa n_{(1)} r \log n, \mu^2 r^2 \kappa^2 n_{(1)}\})$	Condition (3)
[25]	$\mathcal{O}(\mu r n_{(1)} \log^2 n_{(1)})$	Conditions (3) and (12)
[18]	$\mathcal{O}(\mu r n_{(1)} \log^2 n_{(1)})$	Condition (3)
Ours	$\mathcal{O}(\mu r n_{(1)} \log n_{(1)})$	Condition (3)
	$\mathcal{O}(\kappa^2 \mu r n_{(1)} \log(n_{(1)}) \log_{2\kappa}(n_{(1)}))$	Condition (3)
Lower Bound[1] [15]	$\Omega(\mu r n_{(1)} \log n_{(1)})$	Condition (3)

of cardinality m. With strong duality, we can either study the exact recoverability of the primal problem (1), or investigate the validity of its convex dual (or bi-dual) problem (2). Here we study the former with tools from geometric functional analysis. Recall that in the analysis of matrix completion, one typically requires a μ-incoherence condition for a given rank-r matrix $\mathbf{X}^*$ with skinny SVD $\mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$ [46, 15]:

$$\|\mathbf{U}^T\mathbf{e}_i\|_2 \leq \sqrt{\frac{\mu r}{n_1}}, \qquad \text{and} \qquad \|\mathbf{V}^T\mathbf{e}_i\|_2 \leq \sqrt{\frac{\mu r}{n_2}}, \qquad \text{for all } i \tag{3}$$

where $\mathbf{e}_i$'s are vectors with i-th entry equal to 1 and other entries equal to 0. The incoherence condition claims that information spreads throughout the left and right singular vectors and is quite standard in the matrix completion literature. Under this standard condition, we have the following results.

Theorems 5, 7, and 6. (Matrix Completion. Informal.) *$\mathbf{X}^* \in \mathbb{R}^{n_1 \times n_2}$ is the unique matrix of rank at most r that is consistent with the m measurements with high probability, provided $m = \mathcal{O}(\mu(n_1+n_2)r\log(n_1+n_2))$ and $\mathbf{X}^*$ satisfies incoherence (3). In addition, there exists a convex optimization for matrix completion in the form of (2) that exactly recovers $\mathbf{X}^*$ with high probability, provided that $m = \mathcal{O}(\kappa^2\mu(n_1+n_2)r\log(n_1+n_2)\log_{2\kappa}(n_1+n_2))$, where κ is the condition number of $\mathbf{X}^*$.*

To the best of our knowledge, our result is the first to connect convex matrix completion to non-convex matrix completion, two parallel lines of research that have received significant attention in the past few years. Table 1 compares our result with prior results.

[1] This lower bound is information-theoretic.

For robust PCA, instead of studying exact recoverability of problem (1) as for matrix completion, we investigate problem (2) directly. The robust PCA problem is to decompose a given matrix $\mathbf{D} = \mathbf{X}^* + \mathbf{S}^*$ into the sum of a low-rank component $\mathbf{X}^*$ and a sparse component $\mathbf{S}^*$ [1]. We obtain the following theorem for robust PCA.

Theorem 8. (Robust PCA. Informal.) *There exists a convex optimization formulation for robust PCA in the form of problem* (2) *that exactly recovers the incoherent matrix* $\mathbf{X}^* \in \mathbb{R}^{n_1 \times n_2}$ *and* $\mathbf{S}^* \in \mathbb{R}^{n_1 \times n_2}$ *with high probability, even if* $\mathsf{rank}(\mathbf{X}^*) = \Theta\left(\frac{\min\{n_1,n_2\}}{\mu \log^2 \max\{n_1,n_2\}}\right)$ *and the size of the support of* $\mathbf{S}^*$ *is* $m = \Theta(n_1 n_2)$*, where the support set of* $\mathbf{S}^*$ *is uniformly distributed among all sets of cardinality* m*, and the incoherence parameter* μ *satisfies constraints* (3) *and* $\|\mathbf{X}^*\|_\infty \leq \sqrt{\frac{\mu r}{n_1 n_2}}\sigma_r(\mathbf{X}^*)$.

The bounds in Theorem 8 match the best known results in the robust PCA literature when the supports of $\mathbf{S}^*$ are uniformly sampled [13], while our assumption is arguably more intuitive; see Section 5. Note that our results hold even when $\mathbf{X}^*$ is close to full rank and a constant fraction of the entries have noise. Independently of our work, Ge et al. [22] developed a framework to analyze the loss surface of low-rank problems, and applied the framework to matrix completion and robust PCA. Their bounds are: for matrix completion, the sample complexity is $\mathcal{O}(\kappa^6 \mu^4 r^6 (n_1+n_2)\log(n_1+n_2))$; for robust PCA, the outlier entries are deterministic and the number that the method can tolerate is $\mathcal{O}\left(\frac{n_1 n_2}{\mu r \kappa^5}\right)$. Zhang et al. [63] also studied the robust PCA problem using non-convex optimization, where the outlier entries are deterministic and the number of outliers that their algorithm can tolerate is $\mathcal{O}\left(\frac{n_1 n_2}{r\kappa}\right)$. The strong duality approach is unique to our work.

1.2 Our Techniques

Reduction to Low-Rank Approximation. Our results are inspired by the low-rank approximation problem:

$$\min_{\mathbf{A}\in\mathbb{R}^{n_1\times r},\mathbf{B}\in\mathbb{R}^{r\times n_2}} \frac{1}{2}\|-\widetilde{\mathbf{\Lambda}} - \mathbf{AB}\|_F^2. \tag{4}$$

We know that all local solutions of (4) are globally optimal (see Lemma 1) and that strong duality holds for any given matrix $-\widetilde{\mathbf{\Lambda}} \in \mathbb{R}^{n_1 \times n_2}$ [26]. To extend this property to our more general problem (1), our main insight is to reduce problem (1) to the form of (4) using the ℓ_2-regularization term. While some prior work attempted to apply a similar reduction, their conclusions either depended on unrealistic conditions on local solutions, e.g., all local solutions are rank-deficient [28, 26], or their conclusions relied on strong assumptions on the objective functions, e.g., that the objective functions are twice-differentiable [29]. Instead, our general results formulate strong duality via the existence of a dual certificate $\widetilde{\mathbf{\Lambda}}$. For concrete applications, the existence of a dual certificate is then converted to mild assumptions, e.g., that the number of measurements is sufficiently large and the positions of measurements are randomly distributed. We will illustrate the importance of randomness below.

The Blessing of Randomness. The desired dual certificate $\widetilde{\mathbf{\Lambda}}$ may not exist in the deterministic world. A hardness result [45] shows that for the problem of weighted low-rank approximation, which can be cast in the form of (1), without some randomization in the measurements made on the underlying low rank matrix, it is NP-hard to achieve a good objective value, not to mention to achieve strong duality. A similar phenomenon was observed for deterministic matrix completion [32]. Thus we should utilize such randomness to analyze the

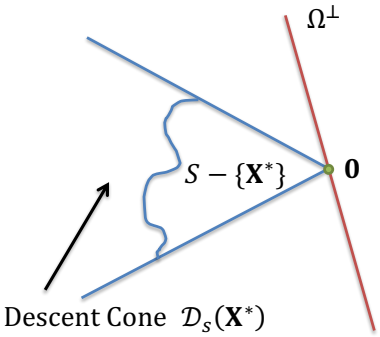


Figure 2 Feasibility.

existence of a dual certificate. For matrix completion, the assumption that the measurements are random is standard, under which, the angle between the space Ω (the space of matrices which are consistent with observations) and the space $\mathcal{T}$ (the space of matrices which are low-rank) is small with high probability, namely, $\mathbf{X}^*$ is almost the unique low-rank matrix that is consistent with the measurements. Thus, our dual certificate can be represented as another form of a convergent Neumann series concerning the projection operators on the spaces Ω and $\mathcal{T}$. The remainder of the proof is to show that such a construction obeys the dual conditions.

To prove the dual conditions for matrix completion, we use the fact that the subspace Ω and the complement space $\mathcal{T}^\perp$ are almost orthogonal when the sample size is sufficiently large. This implies the projection of our dual certificate on the space $\mathcal{T}^\perp$ has a very small norm, which exactly matches the dual conditions.

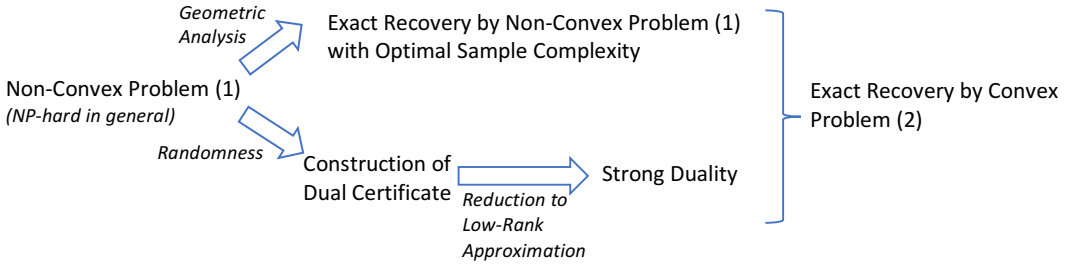
Non-Convex Geometric Analysis. Strong duality implies that the primal problem (1) and its bi-dual problem (2) have exactly the same solutions in the sense that $\widetilde{\mathbf{A}}\widetilde{\mathbf{B}} = \widetilde{\mathbf{X}}$. Thus, to show exact recoverability of linear inverse problems such as matrix completion and robust PCA, it suffices to study either the non-convex primal problem (1) or its convex counterpart (2). Here we do the former analysis for matrix completion. We mention that traditional techniques [15, 46, 16] for convex optimization break down for our non-convex problem, since the subgradient of a non-convex objective function may not even exist [12]. Instead, we apply tools from geometric functional analysis [55] to analyze the geometry of problem (1). Our non-convex geometric analysis is in stark contrast to prior techniques of convex geometric analysis [56] where convex combinations of non-convex constraints were used to define the Minkowski functional (e.g., in the definition of atomic norm) while our method uses the non-convex constraint itself.

For matrix completion, problem (1) has two hard constraints: a) the rank of the output matrix should be no larger than r, as implied by the form of $\mathbf{AB}$; b) the output matrix should be consistent with the sampled measurements, i.e., $\mathcal{P}_\Omega(\mathbf{AB}) = \mathcal{P}_\Omega(\mathbf{X}^*)$. We study the feasibility condition of problem (1) from a geometric perspective: $\widetilde{\mathbf{A}}\widetilde{\mathbf{B}} = \mathbf{X}^*$ is the unique feasible solution to problem (1) if and only if starting from $\mathbf{X}^*$, the rank of $\mathbf{X}^* + \mathbf{D}$ increases for all directions $\mathbf{D}$'s in the constraint set $\Omega^\perp = \{\mathbf{D} \in \mathbb{R}^{n_1 \times n_2} : \mathcal{P}_\Omega(\mathbf{X}^* + \mathbf{D}) = \mathcal{P}_\Omega(\mathbf{X}^*)\}$ (a.k.a. the feasibility condition). This can be geometrically interpreted as the requirement that the descent cone $\mathcal{D}_\mathcal{S}(\mathbf{X}^*) = \{t(\mathbf{X} - \mathbf{X}^*) \in \mathbb{R}^{n_1 \times n_2} : \mathsf{rank}(\mathbf{X}) \leq r,\ t \geq 0\}$ and the constraint set $\Omega^\perp$ must intersect uniquely at $\mathbf{0}$ (see Figure 2), which means $\mathbf{X}^*$ is the unique matrix that satisfies the constraints a) and b). This is shown by the following tangent cone argument.

Let $\mathcal{S}$ be the set of all matrices with rank at most r around the underlying matrix $\mathbf{X}^*$. In the tangent cone argument, by definition, $\mathcal{D}_{\mathcal{S}}(\mathbf{X}^*)$ is a subset of the tangent cone of $\mathcal{S}$ at $\mathbf{X}^*$. The latter cone of interest has a very nice form, namely, it is just the space $\mathcal{T}$ mentioned above (the space of matrices which are low-rank). Now leverage results from prior work which imply $\mathcal{T} \cap \Omega^{\perp} = \{\mathbf{0}\}$ with a large enough sample size. Namely, among all matrices of the form $\mathbf{X}^* + \mathbf{D}$, $\mathbf{D} = \mathbf{0}$ is the only matrix such that $\mathsf{rank}(\mathbf{X}^* + \mathbf{D}) \leq r$ and $\mathbf{X}^* + \mathbf{D}$ is consistent with the observations.

Using this argument, we can show that the sample size needed for exact recovery in matrix completion matches the known lower bound up to a constant factor.

Putting Things Together. We summarize our new analytical framework with the following figure.



Other Techniques. An alternative method is to investigate the exact recoverability of problem (2) via standard convex analysis. We find that the sub-differential of our induced function $\|\cdot\|_{r*}$ is very similar to that of the nuclear norm. With this observation, we prove the validity of robust PCA in the form of (2) by combining this property of $\|\cdot\|_{r*}$ with standard techniques from [13].

2 Preliminaries

We will use calligraphy to represent a set, bold capital letters to represent a matrix, bold lower-case letters to represent a vector, and lower-case letters to represent scalars. Specifically, we denote by $\mathbf{X}^* \in \mathbb{R}^{n_1 \times n_2}$ the underlying matrix. We use $\mathbf{X}_{:t} \in \mathbb{R}^{n_1 \times 1}$ ($\mathbf{X}_{t:} \in \mathbb{R}^{1 \times n_2}$) to indicate the t-th column (row) of $\mathbf{X}$. The entry in the i-th row, j-th column of $\mathbf{X}$ is represented by $\mathbf{X}_{ij}$. The condition number of $\mathbf{X}$ is $\kappa = \sigma_1(\mathbf{X})/\sigma_r(\mathbf{X})$. We let $n_{(1)} = \max\{n_1, n_2\}$ and $n_{(2)} = \min\{n_1, n_2\}$. For a function $H(\mathbf{M})$ on an input matrix $\mathbf{M}$, its conjugate function H^* is defined by $H^*(\mathbf{\Lambda}) = \max_{\mathbf{M}} \langle \mathbf{\Lambda}, \mathbf{M} \rangle - H(\mathbf{M})$. Furthermore, let H^{**} denote the conjugate function of H^*.

We will frequently use $\mathsf{rank}(\mathbf{X}) \leq r$ to constrain the rank of $\mathbf{X}$. This can be equivalently represented as $\mathbf{X} = \mathbf{AB}$, by restricting the number of columns of $\mathbf{A}$ and rows of $\mathbf{B}$ to be r. For norms, we denote by $\|\mathbf{X}\|_F = \sqrt{\sum_{ij} \mathbf{X}_{ij}^2}$ the Frobenius norm of matrix $\mathbf{X}$. Let $\sigma_1(\mathbf{X}) \geq \sigma_2(\mathbf{X}) \geq ... \geq \sigma_r(\mathbf{X})$ be the non-zero singular values of $\mathbf{X}$. The nuclear norm (a.k.a. trace norm) of $\mathbf{X}$ is defined by $\|\mathbf{X}\|_* = \sum_{i=1}^{r} \sigma_i(\mathbf{X})$, and the operator norm of $\mathbf{X}$ is $\|\mathbf{X}\| = \sigma_1(\mathbf{X})$. Denote by $\|\mathbf{X}\|_\infty = \max_{ij} |\mathbf{X}_{ij}|$. For two matrices $\mathbf{A}$ and $\mathbf{B}$ of equal dimensions, we denote by $\langle \mathbf{A}, \mathbf{B} \rangle = \sum_{ij} \mathbf{A}_{ij} \mathbf{B}_{ij}$. We denote by $\partial H(\mathbf{X}) = \{\mathbf{\Lambda} \in \mathbb{R}^{n_1 \times n_2} : H(\mathbf{Y}) \geq H(\mathbf{X}) + \langle \mathbf{\Lambda}, \mathbf{Y} - \mathbf{X} \rangle \text{ for any } \mathbf{Y}\}$ the sub-differential of function H evaluated at $\mathbf{X}$.

We define the indicator function of convex set $\mathcal{C}$ by $\mathbf{I}_{\mathcal{C}}(\mathbf{X}) = \begin{cases} 0, & \text{if } \mathbf{X} \in \mathcal{C}; \\ +\infty, & \text{otherwise.} \end{cases}$ For any non-empty set $\mathcal{C}$, denote by $\mathsf{cone}(\mathcal{C}) = \{t\mathbf{X} : \mathbf{X} \in \mathcal{C},\ t \geq 0\}$.

We denote by Ω the set of indices of observed entries, and $\Omega^{\perp}$ its complement. Without confusion, Ω also indicates the linear subspace formed by matrices with entries in $\Omega^{\perp}$ being 0. We denote by $\mathcal{P}_{\Omega} : \mathbb{R}^{n_1 \times n_2} \to \mathbb{R}^{n_1 \times n_2}$ the orthogonal projector of subspace Ω. We will consider a single norm for these operators, namely, the operator norm denoted by $\|\mathcal{A}\|$ and defined by $\|\mathcal{A}\| = \sup_{\|\mathbf{X}\|_F = 1} \|\mathcal{A}(\mathbf{X})\|_F$. For any orthogonal projection operator $\mathcal{P}_{\mathcal{T}}$ to any subspace $\mathcal{T}$, we know that $\|\mathcal{P}_{\mathcal{T}}\| = 1$ whenever $\mathsf{dim}(\mathcal{T}) \neq 0$. For distributions, denote by $\mathcal{N}(0,1)$ a standard Gaussian random variable, $\mathsf{Uniform}(m)$ the uniform distribution of cardinality m, and $\mathsf{Ber}(p)$ the Bernoulli distribution with success probability p.

3 ℓ_2-Regularized Matrix Factorizations: A New Analytical Framework

In this section, we develop a novel framework to analyze a general class of ℓ_2-regularized matrix factorization problems. Our framework can be applied to different specific problems and leads to nearly optimal sample complexity guarantees. In particular, we study the ℓ_2-regularized matrix factorization problem

$$(\mathbf{P}) \qquad \min_{\mathbf{A} \in \mathbb{R}^{n_1 \times r}, \mathbf{B} \in \mathbb{R}^{r \times n_2}} F(\mathbf{A}, \mathbf{B}) = H(\mathbf{AB}) + \frac{1}{2}\|\mathbf{AB}\|_F^2,\ H(\cdot) \text{ is convex and closed.}$$

We show that under suitable conditions the duality gap between $(\mathbf{P})$ and its dual (bi-dual) problem is zero, so problem $(\mathbf{P})$ can be converted to an equivalent convex problem.

3.1 Strong Duality

We first consider an easy case where $H(\mathbf{AB}) = \frac{1}{2}\|\widehat{\mathbf{Y}}\|_F^2 - \langle \widehat{\mathbf{Y}}, \mathbf{AB} \rangle$ for a fixed $\widehat{\mathbf{Y}}$, leading to the objective function $\frac{1}{2}\|\widehat{\mathbf{Y}} - \mathbf{AB}\|_F^2$. For this case, we establish the following lemma.

▶ **Lemma 1.** *For any given matrix $\widehat{\mathbf{Y}}$, any local minimum of $f(\mathbf{A}, \mathbf{B}) = \frac{1}{2}\|\widehat{\mathbf{Y}} - \mathbf{AB}\|_F^2$ is globally optimal, given by $\mathsf{svd}_r(\widehat{\mathbf{Y}})$. The objective function $f(\mathbf{A}, \mathbf{B})$ around any saddle point has a negative second-order directional curvature. Moreover, $f(\mathbf{A}, \mathbf{B})$ has no local maximum.*[2]

The proof of Lemma 1 is basically to calculate the gradient of $f(\mathbf{A}, \mathbf{B})$ and let it equal to zero. Given this lemma, we can reduce $F(\mathbf{A}, \mathbf{B})$ to the form $\frac{1}{2}\|\widehat{\mathbf{Y}} - \mathbf{AB}\|_F^2$ for some $\widehat{\mathbf{Y}}$ plus an extra term:

$$\begin{aligned} F(\mathbf{A}, \mathbf{B}) &= \frac{1}{2}\|\mathbf{AB}\|_F^2 + H(\mathbf{AB}) = \frac{1}{2}\|\mathbf{AB}\|_F^2 + H^{**}(\mathbf{AB}) \\ &= \max_{\mathbf{\Lambda}} \frac{1}{2}\|\mathbf{AB}\|_F^2 + \langle \mathbf{\Lambda}, \mathbf{AB} \rangle - H^*(\mathbf{\Lambda}) \\ &= \max_{\mathbf{\Lambda}} \frac{1}{2}\| - \mathbf{\Lambda} - \mathbf{AB}\|_F^2 - \frac{1}{2}\|\mathbf{\Lambda}\|_F^2 - H^*(\mathbf{\Lambda}) \triangleq \max_{\mathbf{\Lambda}} L(\mathbf{A}, \mathbf{B}, \mathbf{\Lambda}), \end{aligned} \tag{5}$$

where we define $L(\mathbf{A}, \mathbf{B}, \mathbf{\Lambda}) \triangleq \frac{1}{2}\| - \mathbf{\Lambda} - \mathbf{AB}\|_F^2 - \frac{1}{2}\|\mathbf{\Lambda}\|_F^2 - H^*(\mathbf{\Lambda})$ as the Lagrangian of problem $(\mathbf{P})$,[3] and the second equality holds because H is closed and convex w.r.t. the

[2] Prior work studying the loss surface of low-rank matrix approximation assumes that the matrix $\widetilde{\mathbf{\Lambda}}$ is of full rank and does not have the same singular values [8]. In this work, we generalize this result by removing these two assumptions.

[3] One can easily check that $L(\mathbf{A}, \mathbf{B}, \mathbf{\Lambda}) = \min_{\mathbf{M}} L'(\mathbf{A}, \mathbf{B}, \mathbf{M}, \mathbf{\Lambda})$, where $L'(\mathbf{A}, \mathbf{B}, \mathbf{M}, \mathbf{\Lambda})$ is the Lagrangian of the constraint optimization problem $\min_{\mathbf{A}, \mathbf{B}, \mathbf{M}} \frac{1}{2}\|\mathbf{AB}\|_F^2 + H(\mathbf{M})$, s.t. $\mathbf{M} = \mathbf{AB}$. With a little abuse of notation, we call $L(\mathbf{A}, \mathbf{B}, \mathbf{\Lambda})$ the Lagrangian of the unconstrained problem $(\mathbf{P})$ as well.

argument $\mathbf{AB}$. For any fixed value of $\mathbf{\Lambda}$, by Lemma 1, any local minimum of $L(\mathbf{A},\mathbf{B},\mathbf{\Lambda})$ is globally optimal, because minimizing $L(\mathbf{A},\mathbf{B},\mathbf{\Lambda})$ is equivalent to minimizing $\frac{1}{2}\|-\mathbf{\Lambda}-\mathbf{AB}\|_F^2$ for a fixed $\mathbf{\Lambda}$.

The remaining part of our analysis is to choose a proper $\widetilde{\mathbf{\Lambda}}$ such that $(\widetilde{\mathbf{A}},\widetilde{\mathbf{B}},\widetilde{\mathbf{\Lambda}})$ is a primal-dual saddle point of $L(\mathbf{A},\mathbf{B},\mathbf{\Lambda})$, so that $\min_{\mathbf{A},\mathbf{B}} L(\mathbf{A},\mathbf{B},\widetilde{\mathbf{\Lambda}})$ and problem $(\mathbf{P})$ have the same optimal solution $(\widetilde{\mathbf{A}},\widetilde{\mathbf{B}})$. For this, we introduce the following condition, and later we will show that the condition holds with high probability.

▶ **Condition 2.** *For a solution $(\widetilde{\mathbf{A}},\widetilde{\mathbf{B}})$ to problem $(\mathbf{P})$, there exists an $\widetilde{\mathbf{\Lambda}}\in\partial_{\mathbf{X}}H(\mathbf{X})|_{\mathbf{X}=\widetilde{\mathbf{A}}\widetilde{\mathbf{B}}}$ such that*

$$-\widetilde{\mathbf{A}}\widetilde{\mathbf{B}}\widetilde{\mathbf{B}}^T=\widetilde{\mathbf{\Lambda}}\widetilde{\mathbf{B}}^T \quad \textit{and} \quad \widetilde{\mathbf{A}}^T(-\widetilde{\mathbf{A}}\widetilde{\mathbf{B}})=\widetilde{\mathbf{A}}^T\widetilde{\mathbf{\Lambda}}. \tag{6}$$

Explanation of Condition 2. We note that $\nabla_{\mathbf{A}}L(\mathbf{A},\mathbf{B},\mathbf{\Lambda})=\mathbf{ABB}^T+\mathbf{\Lambda B}^T$ and $\nabla_{\mathbf{B}}L(\mathbf{A},\mathbf{B},\mathbf{\Lambda})=\mathbf{A}^T\mathbf{AB}+\mathbf{A}^T\mathbf{\Lambda}$ for a fixed $\mathbf{\Lambda}$. In particular, if we set $\mathbf{\Lambda}$ to be the $\widetilde{\mathbf{\Lambda}}$ in (6), then $\nabla_{\mathbf{A}}L(\mathbf{A},\widetilde{\mathbf{B}},\widetilde{\mathbf{\Lambda}})|_{\mathbf{A}=\widetilde{\mathbf{A}}}=\mathbf{0}$ and $\nabla_{\mathbf{B}}L(\widetilde{\mathbf{A}},\mathbf{B},\widetilde{\mathbf{\Lambda}})|_{\mathbf{B}=\widetilde{\mathbf{B}}}=\mathbf{0}$. So Condition 2 implies that $(\widetilde{\mathbf{A}},\widetilde{\mathbf{B}})$ is either a saddle point or a local minimizer of $L(\mathbf{A},\mathbf{B},\widetilde{\mathbf{\Lambda}})$ as a function of $(\mathbf{A},\mathbf{B})$ for the fixed $\widetilde{\mathbf{\Lambda}}$.

The following lemma states that if it is a local minimizer, then strong duality holds.

▶ **Lemma 3** (Dual Certificate). *Let $(\widetilde{\mathbf{A}},\widetilde{\mathbf{B}})$ be a global minimizer of $F(\mathbf{A},\mathbf{B})$. If there exists a dual certificate $\widetilde{\mathbf{\Lambda}}$ satisfying Condition 2 and the pair $(\widetilde{\mathbf{A}},\widetilde{\mathbf{B}})$ is a local minimizer of $L(\mathbf{A},\mathbf{B},\widetilde{\mathbf{\Lambda}})$ for the fixed $\widetilde{\mathbf{\Lambda}}$, then strong duality holds. Moreover, we have the relation $\widetilde{\mathbf{A}}\widetilde{\mathbf{B}}=\mathsf{svd}_r(-\widetilde{\mathbf{\Lambda}})$.*

Proof. By the assumption of the lemma, $(\widetilde{\mathbf{A}},\widetilde{\mathbf{B}})$ is a local minimizer of $L(\mathbf{A},\mathbf{B},\widetilde{\mathbf{\Lambda}})=\frac{1}{2}\|-\widetilde{\mathbf{\Lambda}}-\mathbf{AB}\|_F^2+c(\widetilde{\mathbf{\Lambda}})$, where $c(\widetilde{\mathbf{\Lambda}})$ is a function that is independent of $\mathbf{A}$ and $\mathbf{B}$. So according to Lemma 1, $(\widetilde{\mathbf{A}},\widetilde{\mathbf{B}})=\operatorname{argmin}_{\mathbf{A},\mathbf{B}} L(\mathbf{A},\mathbf{B},\widetilde{\mathbf{\Lambda}})$, namely, $(\widetilde{\mathbf{A}},\widetilde{\mathbf{B}})$ globally minimizes $L(\mathbf{A},\mathbf{B},\mathbf{\Lambda})$ when $\mathbf{\Lambda}$ is fixed to $\widetilde{\mathbf{\Lambda}}$. Furthermore, $\widetilde{\mathbf{\Lambda}}\in\partial_{\mathbf{X}}H(\mathbf{X})|_{\mathbf{X}=\widetilde{\mathbf{A}}\widetilde{\mathbf{B}}}$ implies that $\widetilde{\mathbf{A}}\widetilde{\mathbf{B}}\in\partial_{\mathbf{\Lambda}}H^*(\mathbf{\Lambda})|_{\mathbf{\Lambda}=\widetilde{\mathbf{\Lambda}}}$ by the convexity of function H, meaning that $\mathbf{0}\in\partial_{\mathbf{\Lambda}}L(\widetilde{\mathbf{A}},\widetilde{\mathbf{B}},\mathbf{\Lambda})$. So $\widetilde{\mathbf{\Lambda}}=\operatorname{argmax}_{\mathbf{\Lambda}} L(\widetilde{\mathbf{A}},\widetilde{\mathbf{B}},\mathbf{\Lambda})$ due to the concavity of $L(\widetilde{\mathbf{A}},\widetilde{\mathbf{B}},\mathbf{\Lambda})$ w.r.t. variable $\mathbf{\Lambda}$. Thus $(\widetilde{\mathbf{A}},\widetilde{\mathbf{B}},\widetilde{\mathbf{\Lambda}})$ is a primal-dual saddle point of $L(\mathbf{A},\mathbf{B},\mathbf{\Lambda})$.

We now prove the strong duality. By the fact that $F(\mathbf{A},\mathbf{B})=\max_{\mathbf{\Lambda}} L(\mathbf{A},\mathbf{B},\mathbf{\Lambda})$ and that $\widetilde{\mathbf{\Lambda}}=\operatorname{argmax}_{\mathbf{\Lambda}} L(\widetilde{\mathbf{A}},\widetilde{\mathbf{B}},\mathbf{\Lambda})$, we have $F(\widetilde{\mathbf{A}},\widetilde{\mathbf{B}})=L(\widetilde{\mathbf{A}},\widetilde{\mathbf{B}},\widetilde{\mathbf{\Lambda}})\le L(\mathbf{A},\mathbf{B},\widetilde{\mathbf{\Lambda}}),\ \ \forall\mathbf{A},\mathbf{B}$, where the inequality holds because $(\widetilde{\mathbf{A}},\widetilde{\mathbf{B}},\widetilde{\mathbf{\Lambda}})$ is a primal-dual saddle point of L. So on the one hand, $\min_{\mathbf{A},\mathbf{B}}\max_{\mathbf{\Lambda}} L(\mathbf{A},\mathbf{B},\mathbf{\Lambda})=F(\widetilde{\mathbf{A}},\widetilde{\mathbf{B}})\le\min_{\mathbf{A},\mathbf{B}} L(\mathbf{A},\mathbf{B},\widetilde{\mathbf{\Lambda}})\le\max_{\mathbf{\Lambda}}\min_{\mathbf{A},\mathbf{B}} L(\mathbf{A},\mathbf{B},\mathbf{\Lambda})$. On the other hand, by weak duality, $\min_{\mathbf{A},\mathbf{B}}\max_{\mathbf{\Lambda}} L(\mathbf{A},\mathbf{B},\mathbf{\Lambda})\ge\max_{\mathbf{\Lambda}}\min_{\mathbf{A},\mathbf{B}} L(\mathbf{A},\mathbf{B},\mathbf{\Lambda})$. Therefore, we have $\min_{\mathbf{A},\mathbf{B}}\max_{\mathbf{\Lambda}} L(\mathbf{A},\mathbf{B},\mathbf{\Lambda})=\max_{\mathbf{\Lambda}}\min_{\mathbf{A},\mathbf{B}} L(\mathbf{A},\mathbf{B},\mathbf{\Lambda})$, i.e., strong duality holds. Hence,

$$\begin{aligned}\widetilde{\mathbf{A}}\widetilde{\mathbf{B}}&=\operatorname*{argmin}_{\mathbf{AB}} L(\mathbf{A},\mathbf{B},\widetilde{\mathbf{\Lambda}})=\operatorname*{argmin}_{\mathbf{AB}}\frac{1}{2}\|-\widetilde{\mathbf{\Lambda}}-\mathbf{AB}\|_F^2-\frac{1}{2}\|\widetilde{\mathbf{\Lambda}}\|_F^2-H^*(\widetilde{\mathbf{\Lambda}})\\&=\operatorname*{argmin}_{\mathbf{AB}}\frac{1}{2}\|-\widetilde{\mathbf{\Lambda}}-\mathbf{AB}\|_F^2=\mathsf{svd}_r(-\widetilde{\mathbf{\Lambda}}).\end{aligned}$$ ◀

This lemma then leads to the following theorem.

▶ **Theorem 4.** *Denote by $(\widetilde{\mathbf{A}},\widetilde{\mathbf{B}})$ the optimal solution of problem $(\mathbf{P})$. Define a matrix space $\mathcal{T}\triangleq\{\widetilde{\mathbf{A}}\mathbf{X}^T+\mathbf{Y}\widetilde{\mathbf{B}},\ \mathbf{X}\in\mathbb{R}^{n_2\times r},\ \mathbf{Y}\in\mathbb{R}^{n_1\times r}\}$. Then strong duality holds for problem $(\mathbf{P})$, provided that*

$$(1)\ \widetilde{\mathbf{\Lambda}}\in\partial H(\widetilde{\mathbf{A}}\widetilde{\mathbf{B}})\triangleq\Psi,\qquad (2)\ \mathcal{P}_{\mathcal{T}}(-\widetilde{\mathbf{\Lambda}})=\widetilde{\mathbf{A}}\widetilde{\mathbf{B}},\qquad (3)\ \|\mathcal{P}_{\mathcal{T}^\perp}\widetilde{\mathbf{\Lambda}}\|<\sigma_r(\widetilde{\mathbf{A}}\widetilde{\mathbf{B}}). \tag{7}$$

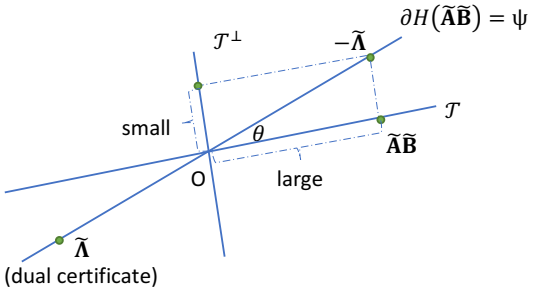


Figure 3 Geometry of dual condition (7) for general matrix factorization problems.

Proof. The proof idea is to construct a dual certificate $\widetilde{\mathbf{\Lambda}}$ so that the conditions in Lemma 3 hold. We note that $\widetilde{\mathbf{\Lambda}}$ should satisfy the following:

$$\begin{aligned}
&\text{(a)} \quad \widetilde{\mathbf{\Lambda}} \in \partial H(\widetilde{\mathbf{A}}\widetilde{\mathbf{B}}), \qquad \text{(by Condition 2)} \\
&\text{(b)} \quad (\widetilde{\mathbf{A}}\widetilde{\mathbf{B}} + \widetilde{\mathbf{\Lambda}})\widetilde{\mathbf{B}}^T = \mathbf{0} \quad \text{and} \quad \widetilde{\mathbf{A}}^T(\widetilde{\mathbf{A}}\widetilde{\mathbf{B}} + \widetilde{\mathbf{\Lambda}}) = \mathbf{0}, \qquad \text{(by Condition 2)} \\
&\text{(c)} \quad \widetilde{\mathbf{A}}\widetilde{\mathbf{B}} = \mathsf{svd}_r(-\widetilde{\mathbf{\Lambda}}). \qquad \text{(by the local minimizer assumption and Lemma 1)}
\end{aligned} \tag{8}$$

By the definition of $\mathcal{T}$ in the theorem statement, it turns out that for any matrix $\mathbf{M} \in \mathbb{R}^{n_1 \times n_2}$, we have $\mathcal{P}_{\mathcal{T}^\perp}\mathbf{M} = (\mathbf{I} - \widetilde{\mathbf{A}}\widetilde{\mathbf{A}}^\dagger)\mathbf{M}(\mathbf{I} - \widetilde{\mathbf{B}}\widetilde{\mathbf{B}}^\dagger)$ and so $\|\mathcal{P}_{\mathcal{T}^\perp}\mathbf{M}\| \le \|\mathbf{M}\|$, a fact that we will frequently use in the subsequent parts of the paper. Denote by $\mathcal{U}$ the left singular space of $\widetilde{\mathbf{A}}\widetilde{\mathbf{B}}$ and $\mathcal{V}$ the right singular space. Then the linear space $\mathcal{T}$ can be equivalently represented as $\mathcal{T} = \mathcal{U} + \mathcal{V}$ by the definition. Therefore, we have $\mathcal{T}^\perp = (\mathcal{U}+\mathcal{V})^\perp = \mathcal{U}^\perp \cap \mathcal{V}^\perp$. With this, we note that: (b) $(\widetilde{\mathbf{A}}\widetilde{\mathbf{B}}+\widetilde{\mathbf{\Lambda}})\widetilde{\mathbf{B}}^T = \mathbf{0}$ and $\widetilde{\mathbf{A}}^T(\widetilde{\mathbf{A}}\widetilde{\mathbf{B}}+\widetilde{\mathbf{\Lambda}}) = \mathbf{0}$ imply $\widetilde{\mathbf{A}}\widetilde{\mathbf{B}}+\widetilde{\mathbf{\Lambda}} \in \mathsf{Null}(\widetilde{\mathbf{A}}^T) = \mathsf{Col}(\widetilde{\mathbf{A}})^\perp$ and $\widetilde{\mathbf{A}}\widetilde{\mathbf{B}} + \widetilde{\mathbf{\Lambda}} \in \mathsf{Row}(\widetilde{\mathbf{B}})^\perp$ (so $\widetilde{\mathbf{A}}\widetilde{\mathbf{B}} + \widetilde{\mathbf{\Lambda}} \in \mathcal{T}^\perp$), and vice versa, where $\mathsf{Null}(\mathbf{Y})$, $\mathsf{Col}(\mathbf{Y})$, $\mathsf{Row}(\mathbf{Y})$ represent the null space, the row space, the column space of any given matrix $\mathbf{Y}$, respectively. And (c) $\widetilde{\mathbf{A}}\widetilde{\mathbf{B}} = \mathsf{svd}_r(-\widetilde{\mathbf{\Lambda}})$ implies that for an orthogonal decomposition $-\widetilde{\mathbf{\Lambda}} = \widetilde{\mathbf{A}}\widetilde{\mathbf{B}} + \mathbf{E}$, where $\widetilde{\mathbf{A}}\widetilde{\mathbf{B}} \in \mathcal{T}$, and $\mathbf{E} \in \mathcal{T}^\perp$, we have $\|\mathbf{E}\| < \sigma_r(\widetilde{\mathbf{A}}\widetilde{\mathbf{B}})$. Conversely, $\|\mathbf{E}\| < \sigma_r(\widetilde{\mathbf{A}}\widetilde{\mathbf{B}})$ and condition (b) imply $\widetilde{\mathbf{A}}\widetilde{\mathbf{B}} = \mathsf{svd}_r(-\widetilde{\mathbf{\Lambda}})$. Therefore, the dual conditions (a), (b), and (c) in (8) are equivalent to (1) $\widetilde{\mathbf{\Lambda}} \in \partial H(\widetilde{\mathbf{A}}\widetilde{\mathbf{B}}) \triangleq \Psi$; (2) $\mathcal{P}_{\mathcal{T}}(-\widetilde{\mathbf{\Lambda}}) = \widetilde{\mathbf{A}}\widetilde{\mathbf{B}}$; (3) $\|\mathcal{P}_{\mathcal{T}^\perp}\widetilde{\mathbf{\Lambda}}\| < \sigma_r(\widetilde{\mathbf{A}}\widetilde{\mathbf{B}})$, as desired. ◀

To show the dual condition in Theorem 4, intuitively, we need to show that the angle θ between subspace $\mathcal{T}$ and Ψ is small (see Figure 3) for a specific function $H(\cdot)$. In the following (see Section B), we will demonstrate applications that, with randomness, obey this dual condition with high probability.

4 Matrix Completion

In matrix completion, there is a hidden matrix $\mathbf{X}^* \in \mathbb{R}^{n_1 \times n_2}$ with rank r. We are given measurements $\{\mathbf{X}^*_{ij} : (i,j) \in \Omega\}$, where $\Omega \sim \mathsf{Uniform}(m)$, i.e., Ω is sampled uniformly at random from all subsets of $[n_1] \times [n_2]$ of cardinality m. The goal is to exactly recover $\mathbf{X}^*$ with high probability. Here we apply our unified framework in Section 3 to matrix completion, by setting $H(\cdot) = \mathbf{I}_{\{\mathbf{M}:\mathcal{P}_\Omega(\mathbf{M})=\mathcal{P}_\Omega(\mathbf{X}^*)\}}(\cdot)$.

A quantity governing the difficulties of matrix completion is the incoherence parameter μ. Intuitively, matrix completion is possible only if the information spreads evenly throughout the low-rank matrix. This intuition is captured by the incoherence conditions. Formally, denote by $\mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$ the skinny SVD of a fixed $n_1 \times n_2$ matrix $\mathbf{X}$ of rank r. Candès et

al. [13, 14, 46, 61] introduced the μ-incoherence condition (3) to the low-rank matrix $\mathbf{X}$. For conditions (3), it can be shown that $1 \leq \mu \leq \frac{n_{(1)}}{r}$. The condition holds for many random matrices with incoherence parameter μ about $\sqrt{r \log n_{(1)}}$ [40].

We have two positive results. The first result is an information-theoretic upper bound: with the standard incoherence condition (3), $\mathbf{X}^*$ is the unique matrix of rank at most r that is consistent with the observations. The proof is deferred to Appendix A.

▶ **Theorem 5** (Information-Theoretic Upper Bound). *Let $\Omega \sim \mathsf{Uniform}(m)$ be the support set uniformly distributed among all sets of cardinality m. Suppose that $m \geq c\mu n_{(1)} r \log n_{(1)}$ for an absolute constant c. Then $\mathbf{X}^*$ is the unique $n_1 \times n_2$ matrix of rank at most r with μ-incoherence condition (3) such that $\mathcal{P}_\Omega(\mathbf{X}) = \mathcal{P}_\Omega(\mathbf{X}^*)$, with probability at least $1 - n_{(1)}^{-10}$.*

Proof Sketch. We consider the feasibility of the matrix completion problem:

$$\text{Find a matrix } \mathbf{X} \in \mathbb{R}^{n_1 \times n_2} \text{ such that } \quad \mathcal{P}_\Omega(\mathbf{X}) = \mathcal{P}_\Omega(\mathbf{X}^*), \quad \mathsf{rank}(\mathbf{X}) \leq r. \tag{9}$$

Our proof first identifies a feasibility condition for problem (9), and then shows that $\mathbf{X}^*$ is the only matrix which obeys this feasibility condition when the sample size is large enough. More specifically, we note that $\mathbf{X}^*$ obeys the conditions in problem (9). Therefore, $\mathbf{X}^*$ is the only matrix which obeys condition (9) if and only if $\mathbf{X}^* + \mathbf{D}$ does not follow the condition for all $\mathbf{D}$, i.e., $\mathcal{D}_\mathcal{S}(\mathbf{X}^*) \cap \Omega^\perp = \{\mathbf{0}\}$, where $\mathcal{D}_\mathcal{S}(\mathbf{X}^*)$ is the descent cone of all low-rank matrices. We note that the descent cone $\mathcal{D}_\mathcal{S}(\mathbf{X}^*)$ is contained in the subspace $\mathcal{T}$ by the tool of geometry functional analysis. Thus by a well-known fact that $\mathcal{T} \cap \Omega^\perp = \{\mathbf{0}\}$ when the sample size is large, the proof is completed. ◀

We describe a simple finite-time inefficient algorithm given Theorem 5 in Section C. This positive result matches a lower bound from prior work, which claims that the sample complexity in Theorem 5 is optimal.

▶ **Theorem 6** (Information-Theoretic Lower Bound. [15], Theorem 1.7). *Denote by $\Omega \sim \mathsf{Uniform}(m)$ the support set uniformly distributed among all sets of cardinality m. Suppose that $m \leq c\mu n_{(1)} r \log n_{(1)}$ for an absolute constant c. Then there exist infinitely many $n_1 \times n_2$ matrices $\mathbf{X}'$ of rank at most r obeying μ-incoherence (3) such that $\mathcal{P}_\Omega(\mathbf{X}') = \mathcal{P}_\Omega(\mathbf{X}^*)$, with probability at least $1 - n_{(1)}^{-10}$.*

Our second positive result converts the feasibility problem in Theorem 5 to a convex optimization problem, which can be *efficiently* solved.

▶ **Theorem 7** (Efficient Matrix Completion). *Let $\Omega \sim \mathsf{Uniform}(m)$ be the support set uniformly distributed among all sets of cardinality m. Suppose $\mathbf{X}^*$ has condition number $\kappa = \sigma_1(\mathbf{X}^*)/\sigma_r(\mathbf{X}^*)$. Then there are absolute constants c and c_0 such that with probability at least $1 - c_0 n_{(1)}^{-10}$, the output of the convex problem*

$$\widetilde{\mathbf{X}} = \operatorname*{argmin}_{\mathbf{X}} \|\mathbf{X}\|_{r*}, \quad \text{s.t.} \quad \mathcal{P}_\Omega(\mathbf{X}) = \mathcal{P}_\Omega(\mathbf{X}^*), \tag{10}$$

is unique and exact, i.e., $\widetilde{\mathbf{X}} = \mathbf{X}^*$, provided that $m \geq c\kappa^2 \mu r n_{(1)} \log_{2\kappa}(n_{(1)}) \log(n_{(1)})$ and $\mathbf{X}^*$ obeys μ-incoherence (3).

Proof Sketch. We have shown in Theorem 5 that $(\widetilde{\mathbf{A}}, \widetilde{\mathbf{B}}) = \operatorname{argmin}_{\mathbf{A},\mathbf{B}} \frac{1}{2}\|\mathbf{AB}\|_F^2$, s.t. $\mathcal{P}_\Omega(\mathbf{AB}) = \mathcal{P}_\Omega(\mathbf{X}^*)$ exactly recovers $\mathbf{X}^*$, i.e., $\widetilde{\mathbf{A}}\widetilde{\mathbf{B}} = \mathbf{X}^*$, with the optimal sample complexity. So if strong duality holds, this non-convex optimization problem can be equivalently

converted to the convex program (10). Then Theorem 7 is straightforward from strong duality.

It now suffices to apply our unified framework in Section 3 to prove the strong duality. We show that the dual condition in Theorem 4 holds with high probability by the following arguments. Let $(\widetilde{\mathbf{A}}, \widetilde{\mathbf{B}})$ be a global solution to problem (10). For $H(\mathbf{X}) = \mathbf{I}_{\{\mathbf{M}\in\mathbb{R}^{n_1\times n_2}:\ \mathcal{P}_\Omega\mathbf{M}=\mathcal{P}_\Omega\mathbf{X}^*\}}(\mathbf{X})$, we have

$$\begin{aligned}\Psi &= \partial H(\widetilde{\mathbf{A}}\widetilde{\mathbf{B}}) \\ &= \{\mathbf{G}\in\mathbb{R}^{n_1\times n_2} : \langle \mathbf{G}, \widetilde{\mathbf{A}}\widetilde{\mathbf{B}}\rangle \geq \langle \mathbf{G}, \mathbf{Y}\rangle, \text{ for any } \mathbf{Y}\in\mathbb{R}^{n_1\times n_2} \text{ s.t. } \mathcal{P}_\Omega\mathbf{Y} = \mathcal{P}_\Omega\mathbf{X}^*\} \\ &= \{\mathbf{G}\in\mathbb{R}^{n_1\times n_2} : \langle \mathbf{G}, \mathbf{X}^*\rangle \geq \langle \mathbf{G}, \mathbf{Y}\rangle, \text{ for any } \mathbf{Y}\in\mathbb{R}^{n_1\times n_2} \text{ s.t. } \mathcal{P}_\Omega\mathbf{Y} = \mathcal{P}_\Omega\mathbf{X}^*\} = \Omega,\end{aligned}$$

where the third equality holds since $\widetilde{\mathbf{A}}\widetilde{\mathbf{B}} = \mathbf{X}^*$. Then we only need to show

$$(1)\ \widetilde{\mathbf{\Lambda}}\in\Omega, \qquad (2)\ \mathcal{P}_{\mathcal{T}}(-\widetilde{\mathbf{\Lambda}}) = \widetilde{\mathbf{A}}\widetilde{\mathbf{B}}, \qquad (3)\ \|\mathcal{P}_{\mathcal{T}^\perp}\widetilde{\mathbf{\Lambda}}\| < \frac{2}{3}\sigma_r(\widetilde{\mathbf{A}}\widetilde{\mathbf{B}}). \tag{11}$$

It is interesting to see that dual condition (11) can be satisfied if the angle θ between subspace Ω and subspace $\mathcal{T}$ is very small; see Figure 3. When the sample size $|\Omega|$ becomes larger and larger, the angle θ becomes smaller and smaller (e.g., when $|\Omega| = n_1n_2$, the angle θ is zero as $\Omega = \mathbb{R}^{n_1\times n_2}$). We show that the sample size $m \geq \Omega(\kappa^2\mu r n_{(1)}\log_{2\kappa}(n_{(1)})\log(n_{(1)}))$ is a sufficient condition for condition (11) to hold. ◀

5 Robust Principal Component Analysis

In this section, we develop our theory for robust PCA based on our framework. In the problem of robust PCA, we are given an observed matrix of the form $\mathbf{D} = \mathbf{X}^* + \mathbf{S}^*$, where $\mathbf{X}^*$ is the ground-truth matrix and $\mathbf{S}^*$ is the corruption matrix which is sparse. The goal is to recover the hidden matrices $\mathbf{X}^*$ and $\mathbf{S}^*$ from the observation $\mathbf{D}$. We set $H(\mathbf{X}) = \lambda\|\mathbf{D}-\mathbf{X}\|_1$.

To make the information spread evenly throughout the matrix, the matrix cannot have one entry whose absolute value is significantly larger than other entries. In this work, we make the following incoherence assumption for robust PCA:

$$\|\mathbf{X}^*\|_\infty \leq \sqrt{\frac{\mu r}{n_1n_2}}\sigma_r(\mathbf{X}^*). \tag{12}$$

Note that condition (12) has an intuitive explanation, namely, that the entries must scatter almost uniformly across the low-rank matrix.

We have the following results for robust PCA.

▶ **Theorem 8** (Robust PCA). *Suppose $\mathbf{X}^*$ is an $n_1\times n_2$ matrix of rank r, and obeys incoherence (3) and (12). Assume that the support set Ω of $\mathbf{S}^*$ is uniformly distributed among all sets of cardinality m. Then with probability at least $1 - cn_{(1)}^{-10}$, the output of the optimization problem*

$$(\widetilde{\mathbf{X}}, \widetilde{\mathbf{S}}) = \underset{\mathbf{X},\mathbf{S}}{\operatorname{argmin}} \|\mathbf{X}\|_{r*} + \lambda\|\mathbf{S}\|_1, \quad \text{s.t.} \quad \mathbf{D} = \mathbf{X} + \mathbf{S},$$

with $\lambda = \frac{\sigma_r(\mathbf{X}^*)}{\sqrt{n_{(1)}}}$ is exact, namely, $\widetilde{\mathbf{X}} = \mathbf{X}^*$ and $\widetilde{\mathbf{S}} = \mathbf{S}^*$, if $\mathsf{rank}(\mathbf{X}^*) \leq \rho_r\frac{n_{(2)}}{\mu\log^2 n_{(1)}}$ and $m \leq \rho_s n_1n_2$, where c, ρ_r, and ρ_s are all positive absolute constants, and function $\|\cdot\|_{r*}$ is given by (13).

The bounds on the rank of $\mathbf{X}^*$ and the sparsity of $\mathbf{S}^*$ in Theorem 8 match the best known results for robust PCA in prior work when we assume the support set of $\mathbf{S}^*$ is sampled uniformly [13].

6 Computational Aspects

Computational Efficiency. We discuss our computational efficiency given that we have strong duality. We note that the dual and bi-dual of primal problem (**P**) are given by

$$\begin{aligned} &(\textbf{Dual, D1}) \qquad \max_{\mathbf{\Lambda}\in\mathbb{R}^{n_1\times n_2}} -H^*(\mathbf{\Lambda}) - \frac{1}{2}\|\mathbf{\Lambda}\|_r^2, \quad \text{where } \|\mathbf{\Lambda}\|_r^2 = \sum_{i=1}^{r}\sigma_i^2(\mathbf{\Lambda}), \\ &(\textbf{Bi-Dual, D2}) \qquad \min_{\mathbf{M}\in\mathbb{R}^{n_1\times n_2}} H(\mathbf{M}) + \|\mathbf{M}\|_{r*}, \quad \text{where } \|\mathbf{M}\|_{r*} = \max_{\mathbf{X}}\langle \mathbf{M}, \mathbf{X}\rangle - \frac{1}{2}\|\mathbf{X}\|_r^2. \end{aligned} \tag{13}$$

Problems (**D1**) and (**D2**) can be solved efficiently due to their convexity. In particular, Grussler et al. [26] provided a computationally efficient algorithm to compute the proximal operators of functions $\frac{1}{2}\|\cdot\|_r^2$ and $\|\cdot\|_{r*}$. Hence, the Douglas-Rachford algorithm can find the global minimum up to an ϵ error in function value in time $\mathsf{poly}(1/\epsilon)$ [33].

Computational Lower Bounds. Unfortunately, strong duality does not always hold for general non-convex problems (**P**). Here we present a very strong lower bound based on the random 4-SAT hypothesis. This is by now a fairly standard conjecture in complexity theory [19] and gives us constant factor inapproximability of problem (**P**) for deterministic algorithms, even those running in exponential time.

If we additionally assume that $\mathsf{BPP} = \mathsf{P}$, where BPP is the class of problems which can be solved in probabilistic polynomial time, and P is the class of problems which can be solved in deterministic polynomial time, then the same conclusion holds for randomized algorithms. This is also a standard conjecture in complexity theory, as it is implied by the existence of certain strong pseudorandom generators or if any problem in deterministic exponential time has exponential size circuits [34]. Therefore, any subexponential time algorithm achieving a sufficiently small constant factor approximation to problem (**P**) in general would imply a major breakthrough in complexity theory.

The lower bound is proved by a reduction from the Maximum Edge Biclique problem [4].

▶ **Theorem 9** (Computational Lower Bound). *Assume Conjecture 20 (the hardness of Random 4-SAT). Then there exists an absolute constant $\epsilon_0 > 0$ for which any deterministic algorithm achieving $(1+\epsilon)\mathsf{OPT}$ in the objective function value for problem (**P**) with $\epsilon \le \epsilon_0$, requires $2^{\Omega(n_1+n_2)}$ time, where OPT is the optimum. If in addition, $\mathsf{BPP} = \mathsf{P}$, then the same conclusion holds for randomized algorithms succeeding with probability at least $2/3$.*

Proof Sketch. Theorem 9 is proved by using the hypothesis that random 4-SAT is hard, in order to show hardness of the Maximum Edge Biclique problem for deterministic algorithms. We then do a reduction from the Maximum Edge Biclique problem to our problem. ◀

Due to space constraints, we defer the proofs of Lemma 1, Theorem 8, some synthetic experiments, and other related work to our full version on arXiv. The proofs of other theorems/lemmas can be found in the appendices.

Acknowledgments. We thank Rong Ge, Zhouchen Lin, and Benjamin Recht for useful discussions. We would like to thank Rina Foygel for finding a bug in the proof of Theorem 7 in a previous version.

References

1. Alekh Agarwal, Sahand Negahban, and Martin J Wainwright. Noisy matrix decomposition via convex relaxation: Optimal rates in high dimensions. *The Annals of Statistics*, pages 1171–1197, 2012.
2. Naman Agarwal, Zeyuan Allen-Zhu, Brian Bullins, Elad Hazan, and Tengyu Ma. Finding approximate local minima for nonconvex optimization in linear time. In *ACM Symposium on Theory of Computing*, pages 1195–1199, 2017.
3. Zeyuan Allen-Zhu. Katyusha: The first direct acceleration of stochastic gradient methods. In *ACM Symposium on Theory of Computing*, 2017.
4. Christoph Ambühl, Monaldo Mastrolilli, and Ola Svensson. Inapproximability results for maximum edge biclique, minimum linear arrangement, and sparsest cut. *SIAM Journal on Computing*, 40(2):567–596, 2011.
5. Anima Anandkumar and Rong Ge. Efficient approaches for escaping higher order saddle points in non-convex optimization. *Annual Conference on Learning Theory*, pages 81–102, 2016.
6. Pranjal Awasthi, Maria-Florina Balcan, Nika Haghtalab, and Hongyang Zhang. Learning and 1-bit compressed sensing under asymmetric noise. In *Annual Conference on Learning Theory*, pages 152–192, 2016.
7. Maria-Florina Balcan and Hongyang Zhang. Noise-tolerant life-long matrix completion via adaptive sampling. In *Advances in Neural Information Processing Systems*, pages 2955–2963, 2016.
8. Pierre Baldi and Kurt Hornik. Neural networks and principal component analysis: Learning from examples without local minima. *Neural Networks*, 2(1):53–58, 1989.
9. Saugata Basu, Richard Pollack, and Marie Francoise Roy. On the combinatorial and algebraic complexity of quantifier elimination. *Journal of the ACM*, 43(6):1002–1045, 1996.
10. Amir Beck and Yonina C Eldar. Strong duality in nonconvex quadratic optimization with two quadratic constraints. *SIAM Journal on Optimization*, 17(3):844–860, 2006.
11. Srinadh Bhojanapalli, Behnam Neyshabur, and Nati Srebro. Global optimality of local search for low rank matrix recovery. In *Advances in Neural Information Processing Systems*, pages 3873–3881, 2016.
12. Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.
13. Emmanuel J. Candès, Xiaodong Li, Yi Ma, and John Wright. Robust principal component analysis? *Journal of the ACM*, 58(3):11, 2011.
14. Emmanuel J. Candès and Ben Recht. Exact matrix completion via convex optimization. *Foundations of Computational Mathematics*, 9(6):717–772, 2009.
15. Emmanuel J. Candès and Terence Tao. The power of convex relaxation: Near-optimal matrix completion. *IEEE Transactions on Information Theory*, 56(5):2053–2080, 2010.
16. Venkat Chandrasekaran, Benjamin Recht, Pablo A Parrilo, and Alan S Willsky. The convex geometry of linear inverse problems. *Foundations of Computational Mathematics*, 12(6):805–849, 2012.
17. Ji Chen and Xiaodong Li. Memory-efficient kernel PCA via partial matrix sampling and nonconvex optimization: a model-free analysis of local minima. *arXiv preprint arXiv:1711.01742*, 2017.
18. Yudong Chen. Incoherence-optimal matrix completion. *IEEE Transactions on Information Theory*, 61(5):2909–2923, 2015.
19. Uriel Feige. Relations between average case complexity and approximation complexity. In *Annual IEEE Conference on Computational Complexity*, page 5, 2002.
20. David Gamarnik, Quan Li, and Hongyi Zhang. Matrix completion from $O(n)$ samples in linear time. In *Annual Conference on Learning Theory*, 2017.

21 Rong Ge, Furong Huang, Chi Jin, and Yang Yuan. Escaping from saddle points – online stochastic gradient for tensor decomposition. In *Annual Conference on Learning Theory*, pages 797–842, 2015.

22 Rong Ge, Chi Jin, and Zheng Yi. No spurious local minima in nonconvex low rank problems: A unified geometric analysis. *International Conference on Machine Learning*, 2017.

23 Rong Ge, Jason D Lee, and Tengyu Ma. Matrix completion has no spurious local minimum. In *Advances in Neural Information Processing Systems*, pages 2973–2981, 2016.

24 Andreas Goerdt and André Lanka. An approximation hardness result for bipartite clique. In *Electronic Colloquium on Computational Complexity, Report*, volume 48, 2004.

25 D. Gross. Recovering low-rank matrices from few coefficients in any basis. *IEEE Transactions on Information Theory*, 57(3):1548–1566, 2011.

26 Christian Grussler, Anders Rantzer, and Pontus Giselsson. Low-rank optimization with convex constraints. *arXiv preprint arXiv:1606.01793*, 2016.

27 Quanquan Gu, Zhaoran Wang, and Han Liu. Low-rank and sparse structure pursuit via alternating minimization. In *International Conference on Artificial Intelligence and Statistics*, pages 600–609, 2016.

28 Benjamin Haeffele, Eric Young, and Rene Vidal. Structured low-rank matrix factorization: Optimality, algorithm, and applications to image processing. In *International Conference on Machine Learning*, pages 2007–2015, 2014.

29 Benjamin D Haeffele and René Vidal. Global optimality in neural network training. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 7331–7339, 2017.

30 Moritz Hardt. Understanding alternating minimization for matrix completion. In *IEEE Symposium on Foundations of Computer Science*, pages 651–660, 2014.

31 Moritz Hardt, Raghu Meka, Prasad Raghavendra, and Benjamin Weitz. Computational limits for matrix completion. In *Annual Conference on Learning Theory*, pages 703–725, 2014.

32 Moritz Hardt and Ankur Moitra. Algorithms and hardness for robust subspace recovery. *Annual Conference on Learning Theory*, 2013.

33 Bingsheng He and Xiaoming Yuan. On the $O(1/n)$ convergence rate of the douglas–rachford alternating direction method. *SIAM Journal on Numerical Analysis*, 50(2):700–709, 2012.

34 Russell Impagliazzo and Avi Wigderson. $P = BPP$ if E requires exponential circuits: Derandomizing the XOR lemma. In *ACM Symposium on the Theory of Computing*, pages 220–229, 1997.

35 Johannes Jahn. *Introduction to the theory of nonlinear optimization*. Springer Berlin Heidelberg, 2007.

36 Prateek Jain, Raghu Meka, and Inderjit S Dhillon. Guaranteed rank minimization via singular value projection. In *Advances in Neural Information Processing Systems*, pages 937–945, 2010.

37 Prateek Jain, Praneeth Netrapalli, and Sujay Sanghavi. Low-rank matrix completion using alternating minimization. In *ACM Symposium on Theory of Computing*, pages 665–674, 2013.

38 Chi Jin, Rong Ge, Praneeth Netrapalli, Sham M Kakade, and Michael I Jordan. How to escape saddle points efficiently. *International Conference on Machine Learning*, 2017.

39 Kenji Kawaguchi. Deep learning without poor local minima. In *Advances in Neural Information Processing Systems*, pages 586–594, 2016.

40 Raghunandan H Keshavan, Andrea Montanari, and Sewoong Oh. Matrix completion from a few entries. *IEEE Transactions on Information Theory*, 56(6):2980–2998, 2010.

41 Yehuda Koren, Robert Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems. *IEEE Computer*, 42(8):30–37, 2009.

42 Yuanzhi Li, Yingyu Liang, and Andrej Risteski. Recovery guarantee of weighted low-rank approximation via alternating minimization. In *International Conference on Machine Learning*, pages 2358–2367, 2016.

43 Praneeth Netrapalli, UN Niranjan, Sujay Sanghavi, Animashree Anandkumar, and Prateek Jain. Non-convex robust PCA. In *Advances in Neural Information Processing Systems*, pages 1107–1115, 2014.

44 Michael L Overton and Robert S Womersley. On the sum of the largest eigenvalues of a symmetric matrix. *SIAM Journal on Matrix Analysis and Applications*, 13(1):41–45, 1992.

45 Ilya Razenshteyn, Zhao Song, and David P. Woodruff. Weighted low rank approximations with provable guarantees. In *ACM Symposium on Theory of Computing*, pages 250–263, 2016.

46 Benjamin Recht. A simpler approach to matrix completion. *Journal of Machine Learning Research*, 12:3413–3430, 2011.

47 James Renegar. On the computational complexity and geometry of the first-order theory of the reals, part I: introduction. preliminaries. the geometry of semi-algebraic sets. the decision problem for the existential theory of the reals. *Journal of symbolic computation*, 13(3):255–300, 1992.

48 James Renegar. On the computational complexity and geometry of the first-order theory of the reals, part II: the general decision problem. preliminaries for quantifier elimination. *Journal of Symbolic Computation*, 13(3):301–328, 1992.

49 Reinhold Schneider and André Uschmajew. Convergence results for projected line-search methods on varieties of low-rank matrices via Lojasiewicz inequality. *SIAM Journal on Optimization*, 25(1):622–646, 2015.

50 Yuan Shen, Zaiwen Wen, and Yin Zhang. Augmented lagrangian alternating direction method for matrix separation based on low-rank factorization. *Optimization Methods and Software*, 29(2):239–263, 2014.

51 Ju Sun, Qing Qu, and John Wright. A geometric analysis of phase retrieval. In *IEEE International Symposium on Information Theory*, pages 2379–2383, 2016.

52 Ju Sun, Qing Qu, and John Wright. Complete dictionary recovery over the sphere I: Overview and the geometric picture. *IEEE Transactions on Information Theory*, 63(2):853–884, 2017.

53 Ruoyu Sun and Zhi-Quan Luo. Guaranteed matrix completion via nonconvex factorization. In *IEEE Symposium on Foundations of Computer Science*, pages 270–289, 2015.

54 Stephen Tu, Ross Boczar, Mahdi Soltanolkotabi, and Benjamin Recht. Low-rank solutions of linear matrix equations via procrustes flow. *International Conference on Machine Learning*, 2016.

55 Roman Vershynin. Lectures in geometric functional analysis, 2009. URL: `https://www.math.uci.edu/~rvershyn/papers/GFA-book.pdf`.

56 Roman Vershynin. Estimation in high dimensions: A geometric perspective. In *Sampling theory, a renaissance*, pages 3–66. Springer, 2015.

57 Yu-Xiang Wang and Huan Xu. Stability of matrix factorization for collaborative filtering. In *International Conference on Machine Learning*, pages 417–424, 2012.

58 Zaiwen Wen, Wotao Yin, and Yin Zhang. Solving a low-rank factorization model for matrix completion by a nonlinear successive over-relaxation algorithm. *Mathematical Programming Computation*, 4(4):333–361, 2012.

59 Xinyang Yi, Dohyung Park, Yudong Chen, and Constantine Caramanis. Fast algorithms for robust PCA via gradient descent. In *Advances in neural information processing systems*, pages 4152–4160, 2016.

60 Hongyang Zhang, Zhouchen Lin, and Chao Zhang. A counterexample for the validity of using nuclear norm as a convex surrogate of rank. In *European Conference on Machine*

Learning and Principles and Practice of Knowledge Discovery in Databases, volume 8189, pages 226–241, 2013.

61 Hongyang Zhang, Zhouchen Lin, and Chao Zhang. Completing low-rank matrices with corrupted samples from few coefficients in general basis. *IEEE Transactions on Information Theory*, 62(8):4748–4768, 2016.

62 Hongyang Zhang, Zhouchen Lin, Chao Zhang, and Edward Chang. Exact recoverability of robust PCA via outlier pursuit with tight recovery bounds. In *AAAI Conference on Artificial Intelligence*, pages 3143–3149, 2015.

63 Xiao Zhang, Lingxiao Wang, and Quanquan Gu. A nonconvex free lunch for low-rank plus sparse matrix recovery. *arXiv preprint arXiv:1702.06525*, 2017.

64 Tuo Zhao, Zhaoran Wang, and Han Liu. A nonconvex optimization framework for low rank matrix estimation. In *Advances in Neural Information Processing Systems*, pages 559–567, 2015.

65 Qinqing Zheng and John Lafferty. Convergence analysis for rectangular matrix completion using Burer-Monteiro factorization and gradient descent. *arXiv preprint arXiv:1605.07051*, 2016.

A Proof of Theorem 5

Theorem 5. (Information-Theoretic Upper Bound. Restated.) *Let $\Omega \sim \mathsf{Uniform}(m)$ be the support set, which is uniformly distributed among all sets of cardinality m. Suppose that $m \geq c\mu n_{(1)} r \log n_{(1)}$ for an absolute constant c. Then $\mathbf{X}^*$ is the unique $n_1 \times n_2$ matrix of rank at most r with μ-incoherence (3) such that $\mathcal{P}_\Omega(\mathbf{X}) = \mathcal{P}_\Omega(\mathbf{X}^*)$, with probability at least $1 - n_{(1)}^{-10}$.*

Proof. We note that the sampling model $\mathsf{Uniform}(m)$ is equivalent to the sampling model $\mathsf{Ber}(p)$ with $p = \Theta\left(\frac{m}{n_1 n_2}\right)$, which we will frequently use in the sequel. We consider the feasibility of the matrix completion problem:

$$\text{Find a matrix } \mathbf{X} \in \mathbb{R}^{n_1 \times n_2} \text{ such that } \quad \mathcal{P}_\Omega(\mathbf{X}) = \mathcal{P}_\Omega(\mathbf{X}^*), \quad \mathsf{rank}(\mathbf{X}) \leq r. \tag{14}$$

Our proof first identifies a feasibility condition for problem (14), and then shows that $\mathbf{X}^*$ is the only matrix that obeys this feasibility condition when the sample size is large enough. We denote by $\mathcal{S} = \{\mathbf{X} \in \mathbb{R}^{n_1 \times n_2} : \mathsf{rank}(\mathbf{X}) \leq r\}$, and define $\mathcal{D}_\mathcal{S}(\mathbf{X}^*) = \{t(\mathbf{X} - \mathbf{X}^*) \in \mathbb{R}^{n_1 \times n_2} : \mathsf{rank}(\mathbf{X}) \leq r,\ t \geq 0\}$. We have the following proposition for the feasibility of problem (14).

▶ **Proposition 10** (Feasibility Condition). *$\mathbf{X}^*$ is the unique feasible solution to problem (14) if $\mathcal{D}_\mathcal{S}(\mathbf{X}^*) \cap \Omega^\perp = \{\mathbf{0}\}$.*

Proof. Notice that problem (14) is equivalent to another feasibility problem

$$\text{Find a matrix } \mathbf{D} \in \mathbb{R}^{n_1 \times n_2} \text{ such that } \quad \mathsf{rank}(\mathbf{X}^* + \mathbf{D}) \leq r, \quad \mathbf{D} \in \Omega^\perp.$$

Suppose that $\mathcal{D}_\mathcal{S}(\mathbf{X}^*) \cap \Omega^\perp = \{\mathbf{0}\}$. Since $\mathsf{rank}(\mathbf{X}^* + \mathbf{D}) \leq r$ implies $\mathbf{D} \in \mathcal{D}_\mathcal{S}(\mathbf{X}^*)$, and note that $\mathbf{D} \in \Omega^\perp$, we have $\mathbf{D} = \mathbf{0}$, which means $\mathbf{X}^*$ is the unique feasible solution to problem (14). ◀

The remainder of the proof is to show $\mathcal{D}_\mathcal{S}(\mathbf{X}^*) \cap \Omega^\perp = \{\mathbf{0}\}$. To proceed, we note that the "escaping through a mesh" techniques for matrix sensing do not work for matrix completion since Ω is not drawn from the Grassmanian according to the Haar measure. To address this issue, we instead need the following lemmas. The first lemma claims that the tangent cone of the set $\mathcal{S}$ evaluated at $\mathbf{X}^*$ is slightly larger than the cone $\mathsf{cone}(\mathcal{S} - \{\mathbf{X}^*\})$.

▶ **Lemma 11** ([35], Theorem 4.8). *Let $\mathcal{S}$ be a non-empty subset of a real normed space. If $\mathcal{S}$ is star-shaped w.r.t. some $\mathbf{X}^* \in \mathcal{S}$, i.e., $t(\mathcal{S} - \{\mathbf{X}^*\}) \subseteq \mathcal{S} - \{\mathbf{X}^*\}$ for all $t \in [0,1]$, then it follows $\mathsf{cone}(\mathcal{S} - \{\mathbf{X}^*\}) \subseteq T(\mathcal{S}, \mathbf{X}^*)$, where $T(\mathcal{S}, \mathbf{X}^*)$ is the tangent cone of the set $\mathcal{S}$ at point $\mathbf{X}^*$ defined by $T(\mathcal{S}, \mathbf{X}^*) = \{\mathbf{\Xi} \in \mathbb{R}^{n_1 \times n_2} : \exists \mathbf{X}_n \subseteq \mathcal{S},\ (a_n) \subseteq \mathbb{R}^+ \text{ s.t. } \mathbf{X}_n \to \mathbf{X}^*,\ a_n(\mathbf{X}_n - \mathbf{X}^*) \to \mathbf{\Xi}\}$.*

The second lemma states that the tangent cone of $\mathcal{S}$ evaluated at $\mathbf{X}^*$ can be represented in a closed form.

▶ **Lemma 12** ([49], Theorem 3.2). *Let $\mathbf{X}^* = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$ be the skinny SVD of matrix $\mathbf{X}^*$. The tangent cone $T(\mathcal{S}, \mathbf{X}^*)$ of the set $\mathcal{S} = \{\mathbf{X} \in \mathbb{R}^{n_1 \times n_2} : \mathsf{rank}(\mathbf{X}) \le r\}$ at $\mathbf{X}^*$ is a linear subspace given by $T(\mathcal{S}, \mathbf{X}^*) = \{\mathbf{U}\mathbf{L}^T + \mathbf{M}\mathbf{V}^T :\ \mathbf{L} \in \mathbb{R}^{n_2 \times r}, \mathbf{M} \in \mathbb{R}^{n_1 \times r}\} \triangleq \mathcal{T}$.*

Now we are ready to prove Theorem 5. By Lemma 11 and 12, we have $\mathcal{D}_{\mathcal{S}}(\mathbf{X}^*) = \mathsf{cone}(\mathcal{S} - \{\mathbf{X}^*\}) \subseteq T(\mathcal{S}, \mathbf{X}^*) = \mathcal{T}$, where the first equality holds by the definition of $\mathcal{D}_{\mathcal{S}}(\mathbf{X}^*)$. So if $\mathcal{T} \cap \Omega^\perp = \{\mathbf{0}\}$, then $\mathcal{D}_{\mathcal{S}}(\mathbf{X}^*) \cap \Omega^\perp = \{\mathbf{0}\}$, meaning that $\mathbf{X}^*$ is the unique feasible solution to the problem (14). Thus the rest of proof is to find a sufficient condition for $\mathcal{T} \cap \Omega^\perp = \{\mathbf{0}\}$. We have the following lemma.

▶ **Lemma 13.** *Assume that $\Omega \sim \mathsf{Ber}(p)$ and the incoherence condition (3) holds. Then with probability at least $1 - n_{(1)}^{-10}$, we have $\|\mathcal{P}_{\Omega^\perp}\mathcal{P}_{\mathcal{T}}\| \le \sqrt{1 - p + \epsilon p}$, provided that $p \ge C_0 \epsilon^{-2}(\mu r \log n_{(1)})/n_{(2)}$, where C_0 is an absolute constant.*

Proof. If $\Omega \sim \mathsf{Ber}(p)$, we have, by Theorem 15, that with high probability $\|\mathcal{P}_{\mathcal{T}} - p^{-1}\mathcal{P}_{\mathcal{T}}\mathcal{P}_{\Omega}\mathcal{P}_{\mathcal{T}}\| \le \epsilon$, provided that $p \ge C_0 \epsilon^{-2} \frac{\mu r \log n_{(1)}}{n_{(2)}}$. Note, however, that since $\mathcal{I} = \mathcal{P}_{\Omega} + \mathcal{P}_{\Omega^\perp}$, $\mathcal{P}_{\mathcal{T}} - p^{-1}\mathcal{P}_{\mathcal{T}}\mathcal{P}_{\Omega}\mathcal{P}_{\mathcal{T}} = p^{-1}(\mathcal{P}_{\mathcal{T}}\mathcal{P}_{\Omega^\perp}\mathcal{P}_{\mathcal{T}} - (1-p)\mathcal{P}_{\mathcal{T}})$ and, therefore, by the triangle inequality $\|\mathcal{P}_{\mathcal{T}}\mathcal{P}_{\Omega^\perp}\mathcal{P}_{\mathcal{T}}\| \le \epsilon p + (1-p)$. Since $\|\mathcal{P}_{\Omega^\perp}\mathcal{P}_{\mathcal{T}}\|^2 \le \|\mathcal{P}_{\mathcal{T}}\mathcal{P}_{\Omega^\perp}\mathcal{P}_{\mathcal{T}}\|$, the proof is completed. ◀

We note that $\|\mathcal{P}_{\Omega^\perp}\mathcal{P}_{\mathcal{T}}\| < 1$ implies $\Omega^\perp \cap \mathcal{T} = \{\mathbf{0}\}$. The proof is completed. ◀

B Proof of Theorem 7

We have shown in Theorem 5 that the problem $(\widetilde{\mathbf{A}}, \widetilde{\mathbf{B}}) = \mathrm{argmin}_{\mathbf{A},\mathbf{B}} \frac{1}{2}\|\mathbf{A}\mathbf{B}\|_F^2, \text{s.t.} \mathcal{P}_{\Omega}(\mathbf{A}\mathbf{B}) = \mathcal{P}_{\Omega}(\mathbf{X}^*)$, exactly recovers $\mathbf{X}^*$, i.e., $\widetilde{\mathbf{A}}\widetilde{\mathbf{B}} = \mathbf{X}^*$, with the optimal sample complexity. So if strong duality holds, this non-convex optimization problem can be equivalently converted to the convex program (10). Then Theorem 7 is straightforward from strong duality.

It now suffices to apply our unified framework in Section 3 to prove the strong duality. We show that the dual condition in Theorem 4 holds with high probability. Let $(\widetilde{\mathbf{A}}, \widetilde{\mathbf{B}})$ be a global solution to problem (10). For $H(\mathbf{X}) = \mathbf{I}_{\{\mathbf{M} \in \mathbb{R}^{n_1 \times n_2}:\ \mathcal{P}_{\Omega}\mathbf{M} = \mathcal{P}_{\Omega}\mathbf{X}^*\}}(\mathbf{X})$, we have

$$\begin{aligned}
\Psi &= \partial H(\widetilde{\mathbf{A}}\widetilde{\mathbf{B}}) \\
&= \{\mathbf{G} \in \mathbb{R}^{n_1 \times n_2} : \langle \mathbf{G}, \widetilde{\mathbf{A}}\widetilde{\mathbf{B}} \rangle \ge \langle \mathbf{G}, \mathbf{Y} \rangle,\ \text{for any } \mathbf{Y} \in \mathbb{R}^{n_1 \times n_2} \text{ s.t. } \mathcal{P}_{\Omega}\mathbf{Y} = \mathcal{P}_{\Omega}\mathbf{X}^*\} \\
&= \{\mathbf{G} \in \mathbb{R}^{n_1 \times n_2} : \langle \mathbf{G}, \mathbf{X}^* \rangle \ge \langle \mathbf{G}, \mathbf{Y} \rangle,\ \text{for any } \mathbf{Y} \in \mathbb{R}^{n_1 \times n_2} \text{ s.t. } \mathcal{P}_{\Omega}\mathbf{Y} = \mathcal{P}_{\Omega}\mathbf{X}^*\} = \Omega,
\end{aligned}$$

where the third equality holds since $\widetilde{\mathbf{A}}\widetilde{\mathbf{B}} = \mathbf{X}^*$. Then we only need to show

$$(1)\ \widetilde{\mathbf{\Lambda}} \in \Omega, \quad (2)\ \mathcal{P}_{\mathcal{T}}(-\widetilde{\mathbf{\Lambda}}) = \widetilde{\mathbf{A}}\widetilde{\mathbf{B}}, \quad (3)\ \|\mathcal{P}_{\mathcal{T}^\perp}\widetilde{\mathbf{\Lambda}}\| < \frac{2}{3}\sigma_r(\widetilde{\mathbf{A}}\widetilde{\mathbf{B}}). \tag{15}$$

We have the following lemma.

▶ **Lemma 14.** *If we can construct an* $\mathbf{\Lambda}$ *such that*

$$\text{(a) } \mathbf{\Lambda} \in \Omega, \quad \text{(b) } \|\mathcal{P}_{\mathcal{T}}(-\mathbf{\Lambda}) - \widetilde{\mathbf{A}}\widetilde{\mathbf{B}}\|_F \le \sqrt{\frac{r}{3n_{(1)}^2}}\sigma_r(\widetilde{\mathbf{A}}\widetilde{\mathbf{B}}), \quad \text{(c) } \|\mathcal{P}_{\mathcal{T}^\perp}\mathbf{\Lambda}\| < \frac{1}{3}\sigma_r(\widetilde{\mathbf{A}}\widetilde{\mathbf{B}}), \tag{16}$$

then we can construct an $\widetilde{\mathbf{\Lambda}}$ *such that Eqn.* (15) *holds with probability at least* $1 - n_{(1)}^{-10}$.

Proof. To prove the lemma, we first claim the following theorem.

▶ **Theorem 15** ([14], Theorem 4.1)**.** *Assume that* Ω *is sampled according to the Bernoulli model with success probability* $p = \Theta(\frac{m}{n_1 n_2})$*, and incoherence condition* (3) *holds. Then there is an absolute constant* C_R *such that for* $\beta > 1$*, we have*

$$\|p^{-1}\mathcal{P}_{\mathcal{T}}\mathcal{P}_{\Omega}\mathcal{P}_{\mathcal{T}} - \mathcal{P}_{\mathcal{T}}\| \le C_R\sqrt{\frac{\beta\mu n_{(1)} r\log n_{(1)}}{m}} \triangleq \epsilon,$$

with probability at least $1 - 3n^{-\beta}$ *provided that* $C_R\sqrt{\frac{\beta\mu n_{(1)} r\log n_{(1)}}{m}} < 1$.

Suppose that Condition (16) holds. Let $\mathbf{Y} = \widetilde{\mathbf{\Lambda}} - \mathbf{\Lambda} \in \Omega$ be the perturbation matrix between $\mathbf{\Lambda}$ and $\widetilde{\mathbf{\Lambda}}$ such that $\mathcal{P}_{\mathcal{T}}(-\widetilde{\mathbf{\Lambda}}) = \widetilde{\mathbf{A}}\widetilde{\mathbf{B}}$. Such a $\mathbf{Y}$ exists by setting $\mathbf{Y} = \mathcal{P}_{\Omega}\mathcal{P}_{\mathcal{T}}(\mathcal{P}_{\mathcal{T}}\mathcal{P}_{\Omega}\mathcal{P}_{\mathcal{T}})^{-1}(\mathcal{P}_{\mathcal{T}}(-\mathbf{\Lambda}) - \widetilde{\mathbf{A}}\widetilde{\mathbf{B}})$. So $\|\mathcal{P}_{\mathcal{T}}\mathbf{Y}\|_F \le \sqrt{\frac{r}{3n_{(1)}^2}}\sigma_r(\widetilde{\mathbf{A}}\widetilde{\mathbf{B}})$. We now prove Condition (3) in Eqn. (15). Observe that

$$\|\mathcal{P}_{\mathcal{T}^\perp}\widetilde{\mathbf{\Lambda}}\| \le \|\mathcal{P}_{\mathcal{T}^\perp}\mathbf{\Lambda}\| + \|\mathcal{P}_{\mathcal{T}^\perp}\mathbf{Y}\| \le \frac{1}{3}\sigma_r(\widetilde{\mathbf{A}}\widetilde{\mathbf{B}}) + \|\mathcal{P}_{\mathcal{T}^\perp}\mathbf{Y}\|. \tag{17}$$

So we only need to show $\|\mathcal{P}_{\mathcal{T}^\perp}\mathbf{Y}\| \le \frac{1}{3}\sigma_r(\widetilde{\mathbf{A}}\widetilde{\mathbf{B}})$.

Before proceeding, we begin by introducing a normalized version $\mathcal{Q}_{\Omega} : \mathbb{R}^{n_1\times n_2} \to \mathbb{R}^{n_1\times n_2}$ of $\mathcal{P}_{\Omega}$: $\mathcal{Q}_{\Omega} = p^{-1}\mathcal{P}_{\Omega} - \mathcal{I}$. With this, we have $\mathcal{P}_{\mathcal{T}}\mathcal{P}_{\Omega}\mathcal{P}_{\mathcal{T}} = p\mathcal{P}_{\mathcal{T}}(\mathcal{I} + \mathcal{Q}_{\Omega})\mathcal{P}_{\mathcal{T}}$. Note that for any operator $\mathcal{P} : \mathcal{T} \to \mathcal{T}$, we have $\mathcal{P}^{-1} = \sum_{k\ge 0}(\mathcal{P}_{\mathcal{T}} - \mathcal{P})^k$ whenever $\|\mathcal{P}_{\mathcal{T}} - \mathcal{P}\| < 1$. So according to Theorem 15, the operator $p(\mathcal{P}_{\mathcal{T}}\mathcal{P}_{\Omega}\mathcal{P}_{\mathcal{T}})^{-1}$ can be represented as a *convergent* Neumann series $p(\mathcal{P}_{\mathcal{T}}\mathcal{P}_{\Omega}\mathcal{P}_{\mathcal{T}})^{-1} = \sum_{k\ge 0}(-1)^k(\mathcal{P}_{\mathcal{T}}\mathcal{Q}_{\Omega}\mathcal{P}_{\mathcal{T}})^k$, because $\|\mathcal{P}_{\mathcal{T}}\mathcal{Q}_{\Omega}\mathcal{P}_{\mathcal{T}}\| \le \epsilon < \frac{1}{2}$ once $m \ge C\mu n_{(1)} r\log n_{(1)}$ for a sufficiently large absolute constant C. We also note that $p(\mathcal{P}_{\mathcal{T}^\perp}\mathcal{Q}_{\Omega}\mathcal{P}_{\mathcal{T}}) = \mathcal{P}_{\mathcal{T}^\perp}\mathcal{P}_{\Omega}\mathcal{P}_{\mathcal{T}}$, because $\mathcal{P}_{\mathcal{T}^\perp}\mathcal{P}_{\mathcal{T}} = 0$. Thus

$$\begin{aligned}
\|\mathcal{P}_{\mathcal{T}^\perp}\mathbf{Y}\| &= \|\mathcal{P}_{\mathcal{T}^\perp}\mathcal{P}_{\Omega}\mathcal{P}_{\mathcal{T}}(\mathcal{P}_{\mathcal{T}}\mathcal{P}_{\Omega}\mathcal{P}_{\mathcal{T}})^{-1}(\mathcal{P}_{\mathcal{T}}(-\mathbf{\Lambda}) - \widetilde{\mathbf{A}}\widetilde{\mathbf{B}}))\| \\
&= \|\mathcal{P}_{\mathcal{T}^\perp}\mathcal{Q}_{\Omega}\mathcal{P}_{\mathcal{T}}p(\mathcal{P}_{\mathcal{T}}\mathcal{P}_{\Omega}\mathcal{P}_{\mathcal{T}})^{-1}((\mathcal{P}_{\mathcal{T}}(-\mathbf{\Lambda}) - \widetilde{\mathbf{A}}\widetilde{\mathbf{B}}))\| \\
&= \|\sum_{k\ge 0}(-1)^k\mathcal{P}_{\mathcal{T}^\perp}\mathcal{Q}_{\Omega}(\mathcal{P}_{\mathcal{T}}\mathcal{Q}_{\Omega}\mathcal{P}_{\mathcal{T}})^k((\mathcal{P}_{\mathcal{T}}(-\mathbf{\Lambda}) - \widetilde{\mathbf{A}}\widetilde{\mathbf{B}}))\| \\
&\le \sum_{k\ge 0}\|(-1)^k\mathcal{P}_{\mathcal{T}^\perp}\mathcal{Q}_{\Omega}(\mathcal{P}_{\mathcal{T}}\mathcal{Q}_{\Omega}\mathcal{P}_{\mathcal{T}})^k((\mathcal{P}_{\mathcal{T}}(-\mathbf{\Lambda}) - \widetilde{\mathbf{A}}\widetilde{\mathbf{B}}))\|_F \\
&\le \|\mathcal{Q}_{\Omega}\|\sum_{k\ge 0}\|\mathcal{P}_{\mathcal{T}}\mathcal{Q}_{\Omega}\mathcal{P}_{\mathcal{T}}\|^k\|\mathcal{P}_{\mathcal{T}}(-\mathbf{\Lambda}) - \widetilde{\mathbf{A}}\widetilde{\mathbf{B}}))\|_F \\
&\le \frac{4}{p}\|\mathcal{P}_{\mathcal{T}}(-\mathbf{\Lambda}) - \widetilde{\mathbf{A}}\widetilde{\mathbf{B}})\|_F \le \Theta\left(\frac{n_1 n_2}{m}\right)\sqrt{\frac{r}{3n_{(1)}^2}}\sigma_r(\widetilde{\mathbf{A}}\widetilde{\mathbf{B}}) \le \frac{1}{3}\sigma_r(\widetilde{\mathbf{A}}\widetilde{\mathbf{B}})
\end{aligned}$$

with high probability. The proof is completed. ◀

It thus suffices to construct a dual certificate $\mathbf{\Lambda}$ such that all conditions in (16) hold. To this end, partition $\Omega = \Omega_1 \cup \Omega_2 \cup ... \cup \Omega_b$ into b partitions of size q. By assumption, we may choose

$$q \geq \frac{128}{3} C\beta\kappa^2 \mu r n_{(1)} \log n_{(1)} \quad \text{and} \quad b \geq \frac{1}{2} \log_{2\kappa} \left(24^2 n_{(1)}^2 \kappa^2\right)$$

for a sufficiently large constant C. Let $\Omega_j \sim \mathsf{Ber}(q)$ denote the set of indices corresponding to the j-th partitions. Define $\mathbf{W}_0 = \widetilde{\mathbf{A}}\widetilde{\mathbf{B}}$ and set $\mathbf{\Lambda}_k = \frac{n_1 n_2}{q} \sum_{j=1}^{k} \mathcal{P}_{\Omega_j}(\mathbf{W}_{j-1})$, $\mathbf{W}_k = \widetilde{\mathbf{A}}\widetilde{\mathbf{B}} - \mathcal{P}_{\mathcal{T}}(\mathbf{\Lambda}_k)$ for $k = 1, 2, ..., b$. Then by Theorem 15,

$$\begin{aligned} \|\mathbf{W}_k\|_F &= \left\| \mathbf{W}_{k-1} - \frac{n_1 n_2}{q} \mathcal{P}_{\mathcal{T}} \mathcal{P}_{\Omega_k}(\mathbf{W}_{k-1}) \right\|_F \\ &= \left\| \left(\mathcal{P}_{\mathcal{T}} - \frac{n_1 n_2}{q} \mathcal{P}_{\mathcal{T}} \mathcal{P}_{\Omega_k} \mathcal{P}_{\mathcal{T}} \right) (\mathbf{W}_{k-1}) \right\|_F \leq \frac{1}{2\kappa} \|\mathbf{W}_{k-1}\|_F. \end{aligned}$$

So it follows that $\|\mathbf{W}_b\|_F \leq (2\kappa)^{-b} \|\mathbf{W}_0\|_F \leq (2\kappa)^{-b} \sqrt{r} \sigma_1(\widetilde{\mathbf{A}}\widetilde{\mathbf{B}}) \leq \sqrt{\frac{r}{24^2 n_{(1)}^2}} \sigma_r(\widetilde{\mathbf{A}}\widetilde{\mathbf{B}})$.

The following lemma together implies the strong duality of (10) straightforwardly.

▶ **Lemma 16.** *Under the assumptions of Theorem 7, the dual certification $\mathbf{W}_b$ obeys the dual condition* (16) *with probability at least $1 - n_{(1)}^{-10}$.*

Proof. It is well known that for matrix completion, the Uniform model $\Omega \sim \mathsf{Uniform}(m)$ is equivalent to the Bernoulli model $\Omega \sim \mathsf{Ber}(p)$, where each element in $[n_1] \times [n_2]$ is included with probability $p = \Theta(m/(n_1 n_2))$ independently. By the equivalence, we can suppose $\Omega \sim \mathsf{Ber}(p)$.

To prove Lemma 16, as a preliminary, we need the following lemmas.

▶ **Lemma 17** ([18], Lemma 2)**.** *Suppose $\mathbf{Z}$ is a fixed matrix. Suppose $\Omega \sim \mathsf{Ber}(p)$. Then with high probability, $\|(\mathcal{I} - p^{-1}\mathcal{P}_\Omega)\mathbf{Z}\| \leq C_0' \left(\frac{\log n_{(1)}}{p} \|\mathbf{Z}\|_\infty + \sqrt{\frac{\log n_{(1)}}{p}} \|\mathbf{Z}\|_{\infty,2} \right)$, where $C_0' > 0$ is an absolute constant and $\|\mathbf{Z}\|_{\infty,2} = \max \left\{ \max_i \sqrt{\sum_b \mathbf{Z}_{ib}^2}, \max_j \sqrt{\sum_a \mathbf{Z}_{aj}^2} \right\}$.*

▶ **Lemma 18** ([13], Lemma 3.1)**.** *Suppose $\Omega \sim \mathsf{Ber}(p)$ and $\mathbf{Z}$ is a fixed matrix. Then with high probability, $\|\mathbf{Z} - p^{-1}\mathcal{P}_{\mathcal{T}}\mathcal{P}_\Omega \mathbf{Z}\|_\infty \leq \epsilon \|\mathbf{Z}\|_\infty$, provided that $p \geq C_0 \epsilon^{-2} (\mu r \log n_{(1)})/n_{(2)}$ for some absolute constant $C_0 > 0$.*

▶ **Lemma 19** ([18], Lemma 3)**.** *Suppose that $\mathbf{Z}$ is a fixed matrix and $\Omega \sim \mathsf{Ber}(p)$. If $p \geq c_0 \mu r \log n_{(1)}/n_{(2)}$ for some c_0 sufficiently large, then by high probability, $\|(p^{-1}\mathcal{P}_{\mathcal{T}}\mathcal{P}_\Omega - \mathcal{P}_{\mathcal{T}})\mathbf{Z}\|_{\infty,2} \leq \frac{1}{2}\sqrt{\frac{n_{(1)}}{\mu r}} \|\mathbf{Z}\|_\infty + \frac{1}{2}\|\mathbf{Z}\|_{\infty,2}$.*

Observe that by Lemma 18, $\|\mathbf{W}_j\|_\infty \leq \left(\frac{1}{2}\right)^j \|\widetilde{\mathbf{A}}\widetilde{\mathbf{B}}\|_\infty$, and by Lemma 19, we have $\|\mathbf{W}_j\|_{\infty,2} \leq \frac{1}{2}\sqrt{\frac{n_{(1)}}{\mu r}} \|\mathbf{W}_{j-1}\|_\infty + \frac{1}{2}\|\mathbf{W}_{j-1}\|_{\infty,2}$. So

$$\begin{aligned} \|\mathbf{W}_j\|_{\infty,2} &\leq \left(\frac{1}{2}\right)^j \sqrt{\frac{n_{(1)}}{\mu r}} \|\widetilde{\mathbf{A}}\widetilde{\mathbf{B}}\|_\infty + \frac{1}{2}\|\mathbf{W}_{j-1}\|_{\infty,2} \\ &\leq j \left(\frac{1}{2}\right)^j \sqrt{\frac{n_{(1)}}{\mu r}} \|\widetilde{\mathbf{A}}\widetilde{\mathbf{B}}\|_\infty + \left(\frac{1}{2}\right)^j \|\widetilde{\mathbf{A}}\widetilde{\mathbf{B}}\|_{\infty,2}. \end{aligned}$$

Therefore, we have $\|\mathcal{P}_{\mathcal{T}^\perp}\mathbf{\Lambda}_b\| \leq \sum_{j=1}^{b} \|\frac{n_1 n_2}{q} \mathcal{P}_{\mathcal{T}^\perp}\mathcal{P}_{\Omega_j}\mathbf{W}_{j-1}\| = \sum_{j=1}^{b} \|\mathcal{P}_{\mathcal{T}^\perp}(\frac{n_1 n_2}{q}\mathcal{P}_{\Omega_j}\mathbf{W}_{j-1} - \mathbf{W}_{j-1})\| \leq \sum_{j=1}^{b} \|(\frac{n_1 n_2}{q}\mathcal{P}_{\Omega_j} - \mathcal{I})(\mathbf{W}_{j-1})\|$. Let p denote $\Theta\left(\frac{q}{n_1 n_2}\right)$. By Lemma 17,

$$\|\mathcal{P}_{\mathcal{T}^\perp}\mathbf{\Lambda}_b\| \le C_0'\frac{\log n_{(1)}}{p}\sum_{j=1}^{b}\|\mathbf{W}_{j-1}\|_\infty + C_0'\sqrt{\frac{\log n_{(1)}}{p}}\sum_{j=1}^{b}\|\mathbf{W}_{j-1}\|_{\infty,2}$$

$$\le C_0'\frac{\log n_{(1)}}{p}\sum_{j=1}^{b}\left(\frac{1}{2}\right)^j\|\widetilde{\mathbf{A}}\widetilde{\mathbf{B}}\|_\infty + C_0'\sqrt{\frac{\log n_{(1)}}{p}}\sum_{j=1}^{b}\left[j\left(\frac{1}{2}\right)^j\sqrt{\frac{n_{(1)}}{\mu r}}\|\widetilde{\mathbf{A}}\widetilde{\mathbf{B}}\|_\infty + \left(\frac{1}{2}\right)^j\|\widetilde{\mathbf{A}}\widetilde{\mathbf{B}}\|_{\infty,2}\right]$$

$$\le C_0'\frac{\log n_{(1)}}{p}\|\widetilde{\mathbf{A}}\widetilde{\mathbf{B}}\|_\infty + 2C_0'\sqrt{\frac{\log n_{(1)}}{p}}\sqrt{\frac{n_{(1)}}{\mu r}}\|\widetilde{\mathbf{A}}\widetilde{\mathbf{B}}\|_\infty + C_0'\sqrt{\frac{\log n_{(1)}}{p}}\|\widetilde{\mathbf{A}}\widetilde{\mathbf{B}}\|_{\infty,2}.$$

Setting $\widetilde{\mathbf{A}}\widetilde{\mathbf{B}} = \mathbf{X}^*$, we note the facts that (we assume WLOG $n_2 \ge n_1$)

$$\|\mathbf{X}^*\|_{\infty,2} = \max_i \|\mathbf{e}_i^T\mathbf{U}\mathbf{\Sigma}\mathbf{V}^T\|_2 \le \max_i \|\mathbf{e}_i^T\mathbf{U}\|\sigma_1(\mathbf{X}^*) \le \sqrt{\frac{\mu r}{n_1}}\sigma_1(\mathbf{X}^*) \le \sqrt{\frac{\mu r}{n_1}}\kappa\sigma_r(\mathbf{X}^*),$$

$$\|\mathbf{X}^*\|_\infty = \max_{ij}\langle\mathbf{X}^*, \mathbf{e}_i\mathbf{e}_j^T\rangle = \max_{ij}\langle\mathbf{U}\mathbf{\Sigma}\mathbf{V}^T, \mathbf{e}_i\mathbf{e}_j^T\rangle = \max_{ij}\langle\mathbf{e}_i^T\mathbf{U}\mathbf{\Sigma}, \mathbf{e}_j^T\mathbf{V}\rangle$$

$$\le \max_{ij}\|\mathbf{e}_i^T\mathbf{U}\mathbf{\Sigma}\mathbf{V}^T\|_2\|\mathbf{e}_j^T\mathbf{V}\|_2 \le \max_j\|\mathbf{X}^*\|_{\infty,2}\|\mathbf{e}_j^T\mathbf{V}\|_2 \le \frac{\mu r\kappa}{\sqrt{n_1 n_2}}\sigma_r(\mathbf{X}^*).$$

Substituting $p = \Theta\left(\frac{\kappa^2\mu r n_{(1)}\log(n_{(1)})\log_{2\kappa}(n_{(1)})}{n_1 n_2}\right)$, we obtain $\|\mathcal{P}_{\mathcal{T}^\perp}\widetilde{\mathbf{\Lambda}}\| < \frac{1}{3}\sigma_r(\mathbf{X}^*)$. The proof is completed. ◀

C Matrix Completion by Information-Theoretic Upper Bound

Theorem 5 formulates matrix completion as a feasibility problem. However, it is a priori unclear if there is an algorithm for finding $\mathbf{X}^*$ with $\mathcal{O}(\mu n_{(1)} r\log n_{(1)})$ sample complexity and incoherence (3) via solving the feasibility problem. To answer this question, we mention that matrix completion can be solved in finite time under these minimum assumptions, namely, we note that the feasibility problem is equivalent to finding a zero of the polynomial $\sum_{(i,j)\in\Omega}(\mathbf{e}_i^T\mathbf{A}\mathbf{B}\mathbf{e}_j - \mathbf{X}^*_{ij})^2 = 0$ w.r.t. the $(n_1+n_2)r$ unknowns $\mathbf{A}$ and $\mathbf{B}$. Since $\mathbf{A}$ can be assumed to be orthogonal, if the entries of $\mathbf{X}^*$ can be written down with $\mathsf{poly}(n)$ bits, then $\|\mathbf{B}\|_F \le \exp(\mathsf{poly}(n))$, which means if one rounds each of the entries of $\mathbf{B}$ to the nearest additive grid multiple of $1/\exp(\mathsf{poly}(n))$, then we will get a rank-k matrix $\mathbf{B}$ where each entry represents the true entry of the optimal $\mathbf{B}$ up to additive $1/\exp(\mathsf{poly}(n))$ error (of course one cannot write down $\mathbf{B}$ in some cases if the entries are irrational). Such an $\mathbf{A}$ and $\mathbf{B}$ can be found in $\exp((n_1+n_2)r)$ time [47, 48, 9]. This gives an exponential time algorithm to solve the feasibility problem in Theorem 5 for matrix completion.

D Proof of Theorem 9

Our computational lower bound for problem (**P**) assumes the hardness of random 4-SAT.

▶ **Conjecture 20** (Random 4-SAT). *Let $c > \ln 2$ be a constant. Consider a random 4-SAT formula on n variables in which each clause has 4 literals, and in which each of the $16n^4$ clauses is picked independently with probability c/n^3. Then any algorithm which always outputs 1 when the random formula is satisfiable, and outputs 0 with probability at least $1/2$ when the random formula is unsatisfiable, must run in $2^{c'n}$ time on some input, where $c' > 0$ is an absolute constant.*

Based on Conjecture 20, we have the following computational lower bound for problem (**P**). We show that problem (**P**) is in general hard for deterministic algorithms. If we additionally assume BPP = P, then the same conclusion holds for randomized algorithms with high probability.

Theorem 9. (Computational Lower Bound. Restated.) *Assume Conjecture 20. Then there exists an absolute constant $\epsilon_0 > 0$ for which any algorithm that achieves $(1+\epsilon)$OPT in objective function value for problem $(\mathbf{P})$ with $\epsilon \leq \epsilon_0$, and with constant probability, requires $2^{\Omega(n_1+n_2)}$ time, where OPT is the optimum. If in addition, BPP = P, then the same conclusion holds for randomized algorithms succeeding with probability at least $2/3$.*

Proof. Theorem 9 is proved by using the hypothesis that random 4-SAT is hard to show hardness of the Maximum Edge Biclique problem for deterministic algorithms.

▶ **Definition 21** (Maximum Edge Biclique)**.** The problem is
Input: An n-by-n bipartite graph G.
Output: A k_1-by-k_2 complete bipartite subgraph of G, such that $k_1 \cdot k_2$ is maximized.

[24] showed that under the random 4-SAT assumption there exist two constants $\epsilon_1 > \epsilon_2 > 0$ such that no efficient deterministic algorithm is able to distinguish between bipartite graphs $G(U, V, E)$ with $|U| = |V| = n$ which have a clique of size $\geq (n/16)^2(1+\epsilon_1)$ and those in which all bipartite cliques are of size $\leq (n/16)^2(1+\epsilon_2)$. The reduction uses a bipartite graph G with at least tn^2 edges with large probability, for a constant t.

Given a given bipartite graph $G(U, V, E)$, define $H(\cdot)$ as follows. Define the matrix $\mathbf{Y}$ and $\mathbf{W}$: $\mathbf{Y}_{ij} = 1$ if edge $(U_i, V_j) \in E$, $\mathbf{Y}_{ij} = 0$ if edge $(U_i, V_j) \notin E$; $\mathbf{W}_{ij} = 1$ if edge $(U_i, V_j) \in E$, and $\mathbf{W}_{ij} = \text{poly}(n)$ if edge $(U_i, V_j) \notin E$. Choose a large enough constant $\beta > 0$ and let $H(\mathbf{AB}) = \beta \sum_{ij} \mathbf{W}_{ij}^2 (\mathbf{Y}_{ij} - (\mathbf{AB})_{ij})^2$. Now, if there exists a biclique in G with at least $(n/16)^2(1+\epsilon_2)$ edges, then the number of remaining edges is at most $tn^2 - (n/16)^2(1+\epsilon_1)$, and so the solution to $\min H(\mathbf{AB}) + \frac{1}{2}\|\mathbf{AB}\|_F^2$ has cost at most $\beta[tn^2 - (n/16)^2(1+\epsilon_1)] + n^2$. On the other hand, if there does not exist a biclique that has more than $(n/16)^2(1+\epsilon_2)$ edges, then the number of remaining edges is at least $(n/16)^2(1+\epsilon_2)$, and so any solution to $\min H(\mathbf{AB}) + \frac{1}{2}\|\mathbf{AB}\|_F^2$ has cost at least $\beta[tn^2 - (n/16)^2(1+\epsilon_2)]$. Choose β large enough so that $\beta[tn^2 - (n/16)^2(1+\epsilon_2)] > \beta[tn^2 - (n/16)^2(1+\epsilon_1)] + n^2$. This combined with the result in [24] completes the proof for deterministic algorithms.

To rule out randomized algorithms running in time $2^{\alpha(n_1+n_2)}$ for some function α of n_1, n_2 for which $\alpha = o(1)$, observe that we can define a new problem which is the same as problem $(\mathbf{P})$ except the input description of H is padded with a string of 1s of length $2^{(\alpha/2)(n_1+n_2)}$. This string is irrelevant for solving problem $(\mathbf{P})$ but changes the input size to $N = \text{poly}(n_1, n_2) + 2^{(\alpha/2)(n_1+n_2)}$. By the argument in the previous paragraph, any deterministic algorithm still requires $2^{\Omega(n)} = N^{\omega(1)}$ time to solve this problem, which is super-polynomial in the new input size N. However, if a randomized algorithm can solve it in $2^{\alpha(n_1+n_2)}$ time, then it runs in $\text{poly}(N)$ time. This contradicts the assumption that BPP = P. This completes the proof. ◀

A Quasi-Random Approach to Matrix Spectral Analysis*†

Michael Ben-Or[1] and Lior Eldar[‡2]

1 **Hebrew University, Jerusalem, Israel**
benor@cs.huji.ac.il
2 **MIT, Cambridge, USA**
leldar@mit.edu

Abstract

Inspired by quantum computing algorithms for Linear Algebra problems [6, 14] we study how simulation on a classical computer of this type of "Phase Estimation algorithms" performs when we apply it to the Eigen-Problem of Hermitian matrices. The result is a completely new, efficient and stable, parallel algorithm to compute an approximate spectral decomposition of any Hermitian matrix. The algorithm can be implemented by Boolean circuits in $O(\log^2 n)$ parallel time with a total cost of $O(n^{\omega+1})$ Boolean operations. This Boolean complexity matches the best known $O(\log^2 n)$ parallel time algorithms, but unlike those algorithms our algorithm is (logarithmically) stable, so it may lead to actual implementations, allowing fast parallel computation of eigenvectors and eigenvalues in practice.

Previous approaches to solve the Eigen-Problem generally use randomization to avoid bad conditions - as we do. Our algorithm makes further use of randomization in a completely new way, taking random powers of a unitary matrix to randomize the phases of its eigenvalues. Proving that a tiny Gaussian perturbation and a random polynomial power are sufficient to ensure almost pairwise independence of the phases (mod 2π) is the main technical contribution of this work. It relies on the theory of low-discrepancy or quasi-random sequences - a theory, which to the best of our knowledge, has not been connected thus far to linear algebra problems. Hence, we believe that further study of this new connection will lead to additional improvements.

1998 ACM Subject Classification G.1.3 Numerical Linear Algebra

Keywords and phrases Eigenvectors, Eigenvalues, low-discrepancy sequence

Digital Object Identifier 10.4230/LIPIcs.ITCS.2018.6

1 Introduction

1.1 General

The eigen-problem of Hermitian matrices is the problem of computing the eigenvalues and eigenvectors of a Hermitian matrix. This problem is ubiquitous in computer science and engineering, and because of its relatively high computational complexity imposes a high computational load on most modern information processing systems.

* The full version of this paper is available on line at https://arxiv.org/abs/1505.08126
† This research project was supported in part by the Israeli Centers of Research Excellence (I-CORE) program (Center No. 4/11), by the Israeli Science Foundation (ISF) research grant 1446/09, by an EU FP7 ERC grant (no.280157), and by the EU FP7-ICT project QALGO (FET-Proactive Scheme).
‡ LE is thankful to the Templeton Foundation for their support of this work.

Eigenvalues and eigenvectors of an input Hermitian matrix, even specified to finite precision, can be irrational numbers. Hence, when computing them, one inherently needs to approximate them. This gives rise to a host of problems: spectral decomposition algorithms are often hard to analyze rigorously, and turn out to be unstable, and difficult to parallelize.

Thus, given a matrix A, we are usually interested not in its exact eigenvalues and eigenvectors, which may be very hard to compute, (and possibly very long to describe once computed), but rather in an approximate decomposition:

▶ **Definition 1** (Approximate Spectral Decomposition - $\mathrm{ASD}(\mathrm{A}, \delta)$)**.** Let A be some $n \times n$ Hermitian matrix. An approximate spectral decomposition of A, with accuracy parameter $\delta = 1/\mathrm{poly}(n)$ is a set of vectors $\{v_i\}_{i=1}^n, \|v_i\| = 1$ such that there exists a complete set of eigenvectors $\{w_i\}_{i=1}^n$ of a matrix A', $\|A' - A\| \leq \delta$ that satisfy:

$$\forall i \quad \|v_i - w_i\| \leq \delta.$$

For a general $n \times n$ matrix A one can consider the Hermitian matrix $A^H A$, in which case $\mathrm{ASD}(\mathrm{A^H A}, \delta)$ is an approximation of the *singular vectors* (and singular values) of A.

We note that the definition of ASD then corresponds to a "smooth analysis" of matrices: namely given input A, we do not find a spectral decomposition of A, but rather the decomposition of a matrix A', such that $\|A - A'\| \leq \delta$. We also point out, that the definition of ASD holds just as well in the case of nearly degenerate matrices: we do not require a one-to-one correspondence with the eigenvectors of A, which can be extremely hard to achieve, but rather to find some set of approximate eigenvectors, such that the corresponding weighted sum of rank-1 projections form an approximation of A.

When one considers an algorithm $\mathcal{A}$ for the ASD problem, one can examine its *arithmetic* complexity or *boolean* complexity. The arithmetic complexity is the minimal size arithmetic circuit C (namely each node computes addition, multiplication or division to unbounded accuracy) that implements $\mathcal{A}$, whereas the boolean complexity counts the number of boolean AND/OR gates of fan-in 2 required to implement $\mathcal{A}$.

Given the definition above, and following Demmel et al. [4] we consider an algorithm $\mathcal{A}$ to be log-stable (or stable for short), if there exists an arithmetic circuit C that implements $\mathcal{A}$ on $n \times n$ matrices, and a number $t = O(\log(n))$, such that each arithmetic computation in C uses at most t bits of precision, and the output of the circuit deviates from the output of the arithmetic circuit by at most $1/\mathrm{poly}(n)$. We note that when an algorithm is *stable* then its boolean complexity is equal to its arithmetic complexity up to a factor $O(\log(n))$. If, however, an algorithm is *unstable* then its boolean complexity could be larger by a factor of up to n. In the study of practical numerical linear algebra algorithms, one usually identifies algorithms that are stable with "practical", and algorithms that are not stable to be impractical. This usually, because the computing machines are restricted to representing numbers with a number of bits that is a small fraction of the size of the input.

In terms of parallelism, we will refer to the complexity class $\mathrm{NC^{(k)}}$ (see Definition 8) which is the set of all computational problems that can be solved by uniform Boolean circuits of size $\mathrm{poly}(n)$ in time $O(\log^k(n))$. Often, we will refer to the class $\mathrm{RNC^{(k)}}$, in which the parallel $\mathrm{NC^{(k)}}$ circuit is also allowed to accept uniform random bits. One would like an ASD algorithm to have minimal arithmetic or boolean complexity, and minimal parallel time. Ideally, one would also like this algorithm to be stable.

1.2 Main Contribution

Inspired by recent quantum computing algorithms [6, 14], we introduce a new perspective on the problem of computing the ASD that is based on low-discrepancy sequences. Roughly

speaking, low-discrepancy sequences are deterministic sequences which appear to be random, because they "visit" each small sub-cube the same number of times that a completely random sequence would, up to a small additive error.

▶ **Definition 2** (Multi-dimensional Discrepancy)**.** For integer s, put $I^s = [0,1)^s$. Given a sequence $x = (x_n)_{n=1}^N$, with $x_n \in I^s$ the discrepancy $D_N(x)$ is defined as:

$$D_N(x) = \sup_{B \in \mathcal{B}} \left\{ \left| \frac{1}{N} \sum_{n=1}^{N} \chi_B(x_n) - \mathrm{vol}(B) \right| \right\},$$

where $\chi_B(x_n)$ is an indicator function which is 1 if $x_n \in B$ and $\mathcal{B}$ is the set of all s-products of intervals $\prod_{i=1}^s [u_i, v_i]$, with $[u_i, v_i](\mathrm{mod}1) \subseteq [0,1)$.

We recast the ASD problem as a question about the discrepancy of a certain sequence related to the input matrix. Specifically, given a Hermitian matrix A with n unique eigenvalues $\{\lambda_i\}_{i\in[n]}$ the central object of interest is the sequence comprised of n-dimensional vectors of eigenvalue residuals:

$$S(A) = \left(\{\lambda_1 \cdot 1\}, \ldots, \{\lambda_n \cdot 1\}\right), \left(\{\lambda_1 \cdot 2\}, \ldots, \{\lambda_n \cdot 2\}\right), \ldots \left(\{\lambda_1 \cdot M\}, \ldots, \{\lambda_n \cdot M\}\right),$$

where $\{x\}$ is the fractional part of $x \in \mathbb{R}$, and $M = \mathrm{poly}(n)$ is some large integer. $S(A)$ is hence a sequence of length M in $[0,1)^n$. We would like $S(A)$ to have as small discrepancy as possible. Hence, in sharp contrast to previous algorithms, instead of the computational effort being concentrated on revealing "structure" in the matrix, our algorithm is actually focused on producing random-behaving dynamics.

The main application of our approach presented in this paper is a new stable and parallel algorithm for computing the ASD of any Hermitian matrix. We assume w.l.o.g. that the input matrix is positive-semidefinite (otherwise it can be scaled and shifted by appropriate multiple of identity) and claim:

▶ **Theorem 3.** *Let A be some $n \times n$ Hermitian matrix such that $0 \preceq A \preceq 0.9I$. Let $\delta = 1/\mathrm{poly}(n)$. Then $\mathrm{ASD}(\mathrm{A}, \delta) \in \mathrm{RNC}^{(2)}$, with circuit size $\tilde{O}(n^{\omega+1})$. The algorithm is* log*-stable.*

The boolean complexity of our algorithm is $O(n^{\omega+1})$. If however, one is interested in sampling a uniformly random eigenvector, it can be achieved in complexity $O(n^{\omega})$. [1]

1.3 Prior Art

There are numerous algorithms for computing the ASD of a matrix, relying most prominently on the QR decomposition [15]. For specific types of matrices, like tridiagonal matrices much faster algorithms are known [11], but here we consider the most general Hermitian case. We summarize the state of the art algorithms for this problems in terms of their complexity (boolean / arithmetic, serial / parallel) and compare them to our own:

[1] ω signifies the infimum over all constants c such that one can multiply two matrices in at most n^c arithmetic operations, and $O(\log(n))$ time.

	Arithmetic Complexity	Boolean Complexity	Parallel Time	Log-Stable	Comments
Csanky [7]	$\tilde{O}(n^{\omega+1})$	$\tilde{O}(n^{\omega+2})$	$\log^2(n)$	NO	
Demmel et al. [4]	$\tilde{O}(n^{\omega})$	$\tilde{O}(n^{\omega})(*)$	N/A	YES	$*$ Conjectured for a variant of the algorithm.
Bini et al., Reif [3, 11]	$\tilde{O}(n^{\omega})$	$\tilde{O}(n^{\omega+1})$	$O(\log^2(n))$	NO	Working with $\Omega(n)$ bit Integers
New	$\tilde{O}(n^{\omega+1})$	$\tilde{O}(n^{\omega+1})$	$\log^2(n)$	YES	

Comparing our algorithm to the best known $\mathrm{NC}^{(2)}$ algorithms, it is more efficient by a factor of n compared with Csanky's algorithm [7]. Notably, our algorithm is completely disjoint from Csanky's techniques - which rely on computing explicitly high powers of the input matrix, and computes the characteristic polynomial of the matrix using the Newton identities on the traces of those powers. This is an inherently unstable algorithm as it finds the eigenvalues by approximating the roots of the characteristic polynomial and small perturbation to the coefficients of the polynomial may lead to large deviations of the roots.

The algorithms of Demmel et al., Bini et al. and Reif, rely on efficient implementation of variants of the QR algorithm. Our asymptotic bounds are worse then Demmel et al. in terms of total arithmetic/boolean complexity, though we conjecture that this is an artifact of our proof strategy, and not an inherent problem (see the section on open problems), and in fact, we conjecture that a certain variant of the algorithm could probably achieve a boolean complexity of $O(n^{\omega})$. We note that the QR algorithm is not known to be parallelizable in a stable way, and hence the fast parallel algorithms of Bini et al. and Reif are not stable and probably impractical. In fact the QR decomposition has been shown, for standard implementations like the Given's or Householder method, to be P-complete [8] assuming the real-RAM model. Thus, it is unlikely to be stably-parallelizable unless $\mathrm{P} = \mathrm{NC}$. [2]

Thus, to the best our knowledge, our algorithm is the first parallel algorithm for the ASD of general Hermitian matrices that is both parallel and stable. In particular it achieves the smallest bit-complexity of any $\mathrm{RNC}^{(2)}$ algorithm to date. We conjecture that our approach may present a practical and parallel alternative to computing the ASD.

1.4 Overview of the Algorithm

To compute the ASD of a given matrix A, we first consider a similar problem of sampling uniformly an approximate eigenvector of A, where the eigenvalues of A are assumed to be well-separated. Clearly, if one can sample from this distribution in RNC^2, then by the coupon collector's bound concatenating $O(n\log(n))$ many parallel copies of this routine, one can sample all eigenvectors quickly with high probability. To do this, we require a definition of a Hermitian matrix that is δ-separated:

▶ **Definition 4** (δ-separated)**.** Let A be an $n \times n$ PSD matrix with eigenvalues $\lambda_1 > \lambda_2 > \ldots > \lambda_n \geq 0$. We say that A is δ-separated if $\lambda_j - \lambda_{j+1} \geq \delta$ for all $j < n$, and $\lambda_1 \leq 1/(2\pi) - \delta$.

Next, we introduce the notion of a separating integer w.r.t. a sequence of real numbers:

[2] We point out that the algorithm of Reif [11] achieves a QR factorization in parallel time $O(\log^2(n))$ in the arithmetic model, thus showing that QR is indeed parallelizable, but it relies on computations modulo large integers and therefore not stable and not practical.

▶ **Definition 5** (Separating Integer). Let $\bar{\lambda} = (\lambda_1, \ldots, \lambda_n) \in [0,1)^n$. For $\alpha > 4$ define $B_{in} \subseteq B_{out} \subseteq [0,1)$ as:

$$B_{out} = [-1/(4n), 1/(4n)](\text{mod}1) \quad \text{and} \quad B_{in}(\alpha) = [-1/(\alpha n), 1/(\alpha n)](\text{mod } 1),$$

A positive integer m is said to separate the k-th element of $\bar{\lambda}$ w.r.t. B_{in}, B_{out} if it satisfies:

- $\{m\lambda_k\} \in B_{in}(\alpha)$
- $\forall j \neq k \;\; \{m\lambda_j\} \notin B_{out}$

and finally define the notion of a separating integer w.r.t. a δ-separated matrix.

▶ **Definition 6.** Let A be a δ-separated matrix with eigenvalues $\bar{\lambda} = (\lambda_1, \ldots, \lambda_n)$. A positive integer m is said to separate k in A w.r.t. B_{in}, B_{out}, if m separates the k-th element of $\bar{\lambda}$ (namely, the k-th eigenvalue of A) w.r.t. B_{in}, B_{out}.

Following is a sketch of the main sampling routine. For complete details see Section 5. The routine accepts a separating integer m of the i-th eigenvalue of a δ-separated matrix A, a precision parameter δ and returns a δ approximation of the i-th eigenvector of A:

Algorithm 1 Filter(A, m, δ)

Input: $n \times n$ Hermitian matrix $A \succeq 0$, integer m, $\delta = 1/\text{poly}(n)$. A is δ-separated.

1. **Compute parameters:**
   $$p = 2n^2\lceil \ln(1/\delta) \rceil, \zeta = \delta^2/(2pm).$$
2. **Sample random unit vector:**
   Sample a standard complex Gaussian vector v, set $w_0 = v/\|v\|$.
3. **Approximate matrix exponent:**
   Compute a ζ Taylor approximation of $e^{2\pi i A}$, denoted by $\tilde{U}$.
4. **Raise to power:**
   Compute $\tilde{U}^m$ by repeated squaring.
5. **Generate matrix polynomial:**
   Compute $B = \left(\frac{I+\tilde{U}^m}{2}\right)^p$ by repeated squaring.
6. **Filter:**
   Compute $w = \frac{B \cdot w_0}{\|B \cdot w_0\|}$.
7. **Decide:**
   Set $z = A \cdot w$, $i_0 = \arg\max_{i \in [n]} |w_i|$ and compute $c = z_{i_0}/w_{i_0}$. If
   $$\|A \cdot w - c \cdot w\| \leq 3\delta\sqrt{n}$$
   return w, and otherwise reject.

In words - the algorithm samples a random vector and then multiplies it essentially by the matrix $B = ((I + e^{2\pi i A m})/2)^p$. After this "filtering" step, it evaluates whether or not the resulting vector is close to being an eigenvector of A, and keeps this vector if it is. To understand the behavior of the algorithm, it is insightful to consider the behavior in the eigenbasis of A.

$$w = \sum_i \alpha_i w_i,$$

where $\{w_i\}_{i\in[n]}$ is an orthonormal basis for A corresponding to eigenvalues $\{\lambda_i\}_{i\in[n]}$. If $\{m\lambda_i\}$, i.e. - the fractional part of $m\lambda_i$, is very close to 0 (i.e. inside B_{in}) and $\{m\lambda_j\}$ is $\sim 2\ln n/p$ far from 0 (i.e. outside B_{out}) for all $j \neq i$, then after multiplication by B and normalization, all eigenvectors w_j for $j \neq i$ are attenuated by factor $1/n^2$ relative to w_i, and hence the resulting vector is $1/n$ close to an *eigenvector* of λ_i.

Hence, a sufficient condition on the number m that would imply that $w = Filter(A, m, \delta)$ is an approximation of the i-th eigenvector is the following property: $\{m\lambda_i\}$ is very close to 0, and for all $j \neq i$ $\{m\lambda_j\}$ is bounded away from 0. This corresponds to the fact that m *separates* i in A, as assumed.

So to sample uniformly an approximate eigenvector, we would like to call $Filter(A, m, \delta)$ for $m \sim U[M]$ for $M = \text{poly}(n)$ and prove that m separates i where $i \sim U[n]$. The main observation here, is that this property holds if the sequence of residuals of integer multiples of the eigenvalues $S(A)$ defined above has the aforementioned *low discrepancy* property.

Most of the work in this study is devoted to achieving this property. Computationally, we achieve low-discrepancy of $S(A)$ simply by additive Gaussian perturbation prior to calling the sampling routine. We show that if we perturb a matrix using a Gaussian matrix $\mathcal{E}$ of variance $1/\text{poly}(n)$, then $S(A + \mathcal{E})$ has discrepancy which is $1/\text{poly}(n)$. Showing this is non-trivial because arbitrary vectors of eigenvalues $\lambda_1, \ldots, \lambda_n$ do not generate low-discrepancy sequences in general, and on the other hand we are also severely limited in our ability to perturb the eigenvalues without deviating too much from the original matrix. This is the subject of our main technical theorem 34, which may be of independent interest:

▶ **Theorem** (Informal). *Let A be an $n \times n$ Hermitian matrix, and $\mathcal{E}$ be a standard Gaussian matrix. For any $a > 0, b > 0$ there exists $M = M(a, b) = \text{poly}(n)$ such that w.p. at least $1 - n^{-b}$ the sequence of residuals of eigenvalue multiples of $A + n^{-a} \cdot \mathcal{E}$ of length M has discrepancy at most n^{-b}.*

Perturbing the input matrix has the additional benefit of making sure that A has a exactly n unique eigenvalues with high probability. This follows from a breakthrough theorem by Nguyen, Tao and Vu [9] which has provided a resolution of this long-standing open problem, which was considered unproven folklore until that point. This theorem allows us to handle general Hermitian matrices without extra conditions on the conditioning number of A or its eigenvalue spacing.

1.4.1 Comparison to the power method / QR algorithm

A natural benchmark by which to test the novelty of the proposed algorithm is the iterative power-method for computing the eigenvalues of a Hermitian matrix. In this method, one starts from some random vector b_0, and at each iteration k sets:

$$b_{k+1} = \frac{Ab_k}{\|Ab_k\|}.$$

Both the power method and our proposed scheme are similar in the sense that they attempt to extract the eigenvectors of the input matrix directly. Also, if two eigenvalues are ε-close in magnitude, for some $\varepsilon > 0$, then they require essentially the same exponent of A in the power method, and of e^{iA} in our scheme to distinguish between them. However, the similarity stops here. We maintain, that the power method is both conceptually different, and for general Hermitian matrices performs much worse, in terms of running time, compared with our proposed algorithm.

Conceptually, in the power method, we seek to leverage the difference in *magnitude* between adjacent eigenvalues in order to extract the eigenvectors. On the other hand, in our proposed scheme we recast the problem on the unit sphere $S^{(1)}$, where we are interested in the spacing of the residuals of integer multiples of the eigenvalues. Worded differently, our setting exploits the additive group structure of the eigenvalues modulo 1, whereas the power method distinguishes between them multiplicatively.

In the additive group setting, the advantage is that we can consider the discrepancy of the sequence of residuals, and analyze how quickly these residuals mimic a completely independent random distribution. Furthermore, in the additive setting there is inherent symmetry between the eigenvalues, as no eigenvalue is more likely to be sampled than another. This allows for a natural parallelization of the algorithm to extract simultaneously approximation of all eigenvectors.

The well-known QR algorithm for eigendecomposition [5] is the de-facto standard for computing the ASD, and is considered by some as a parallel version of the power-method. That algorithm applies an iterated sequence of of QR decompositions: At each step k we compute (where $A_1 = A$ - the input matrix)

$$A_k = Q_k R_k,$$

and then set

$$A_{k+1} = R_k Q_k.$$

The algorithm runs in time $\tilde{O}(n^3)$, by applying several pre-processing steps [5], and the fast variant of Demmel et al. in time $O(n^\omega)$. However, as stated above, the QR decomposition which is at the core of these methods is not known to be stably parallel.

1.5 Open Questions

We outline several open questions that may be interesting to research following this work:

1. Is it possible to attain a serial run-time of $O(n^\omega)$ for this algorithm? We conjecture that this is possible based on numerical evidence for a variant of this algorithm, yet we do not have a proof of this fact.
2. What other linear-algebra algorithms can be designed using our methods ? We would like these algorithms to improve on previous algorithms in either the stability, boolean complexity, parallel run-time, or all these parameters simultaneously.
3. Could one reduce the number of random bits required by the algorithm? Currently - we show that using $\tilde{O}(n^2)$ random bits - i.e. applying additive Gaussian perturbation results in a matrix whose eigenvalues seed a low-discrepancy sequence. However, can one do away with only $\tilde{O}(n)$ random bits - by applying a tri-diagonal perturbation to the matrix?

2 Preliminaries

2.1 Notation

A random variable x distributed according to distribution $\mathcal{D}$ is denoted by $x \sim \mathcal{D}$. We will use the letter D to denote the *discrepancy* of a sequence, and the calligraphy letter $\mathcal{D}$ to denote a distribution. For a matrix X, $\|X\|$ signifies the operator norm of X. For a set S, $U[S]$ is the uniform distribution on S. For integer $M > 0$ the set $[M]$ is the set of integers

$\{0, 1, \ldots, M-1\}$. For real number x, $\{x\}$ denotes the fractional part of x: $\{x\} = x - \lfloor x \rfloor$. For real number x, $\lceil x \rfloor \in [-1/2, 1/2)$ denotes the rounding error of x - i.e. $\min\{x - \lceil x \rceil, x - \lfloor x \rfloor\}$. $\mathbb{N}, \mathbb{Z}, \mathbb{C}$ signify the natural, integer, and complex numbers, respectively. For a matrix A, A^H is the Hermitian conjugate-transpose of A. For number $n > 0$ $\ln n$ denotes the natural logarithm, and $\log n$ denotes the binary logarithm. $\mu(\eta, \sigma^2)$ is the Gaussian measure with mean η and variance σ^2. An n-dimensional vector is σ-normal if its components are i.i.d. $\mu(0, \sigma^2)$. $U(n)$ is the set of $n \times n$ unitary matrices. For a Hermitian $n \times n$ matrix A, with eigenvalues $\{\lambda_i\}_{i=1}^n$, $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_n$ $\mathcal{L}(A) = (\lambda_1, \ldots, \lambda_n) \in \mathbb{R}^n$ denotes the vector of sorted eigenvalues of A. For a measurable subset $S \subseteq \mathbb{R}^n$ $\mathrm{vol}(S)$ denotes the volume of S. $\emptyset$ is the empty set. GUE is the global unitary ensemble of random matrices: these are Hermitian matrices whose upper-triangular entries are independently sampled as $\mu(0, 1)$.

2.2 Definitions

2.2.1 Complexity

▶ **Definition 7.** Let ω denote the infimum over all t such that any two $n \times n$ matrices can be multiplied using a number of products at most n^t, and time $O(\log(n))$.

The current best upper-bound on ω is 2.372 due to Williams [16].

▶ **Definition 8** (Class NC)**.** The class $\mathrm{NC}^{(k)}$ is the set of problems computed by uniform boolean circuits, with a polynomial number of gates, and depth at most $O(\log^k n)$.

Additionally, we will require the following fact:

▶ **Fact 9** ([1])**.** *There exists an algorithm for sorting n numbers in time $O(\log(n))$, using n processors.*

▶ **Definition 10** (Class RNC)**.** The class $\mathrm{RNC}^{(k)}$ is the set of problems that can be computed by uniform boolean circuits, with a polynomial number of gates, accepting a polynomial number of random bits, and depth at most $O(\log^k n)$.

For simplicity, we shall assume in this work that RNC circuits are allowed to accept t-bit numbers, sampled from a suitably truncated Gaussian distribution, and discretized to t-bits of precision.

2.2.2 Stable Computation

Following Demmel et al. [4] we define the notion of log-stability as one where truncating each binary arithmetic operation to $O(\log(n))$ bits of precision doesn't change the result by much:

▶ **Definition 11** ((t, δ)-stable randomized computation)**.** Let C denote a randomized arithmetic circuit, and $\mathcal{D}$ be its output distribution supported on $\mathbb{R}^n$. Let D denote the discretization of C to t bits as follows: each infinite-precision arithmetic operation is followed by rounding to t bits. Let $\mathcal{D}'$ denote the output distribution of D. C is said to be (t, δ)-stable if

$$\forall x \quad \exists y, \quad \mathcal{D}(x) = \mathcal{D}'(y) \text{ and } \|x - y\| \leq \delta.$$

▶ **Definition 12** (Log-stable computation)**.** Let C be a randomized arithmetic circuit that accepts n input numbers. C is said to be log-stable if for any $\delta = 1/\mathrm{poly}(n)$ it is (t, δ)-stable for some $t = O(\log(1/\delta))$.

3 Additive Perturbation

Matrix perturbation is a well-developed theory [13, 5] examining the behavior of eigen-values and eigen-vectors under additive perturbation, usually much smaller compared to the norm of the original matrix. While general eigenvalue problems are usually unstable against perturbation, for Hermitian matrices the situation is much better: the Bauer-Fike theorem [2] states that the perturbed eigenvalues can only deviate from the original eigenvalues by an amount corresponding to the relative strength of the perturbation. This holds regardless of whether the perturbation itself is Hermitian.

In particular, when the perturbed matrix A is δ-separated and the perturbation itself is Hermitian (GUE, for example) one can compute an explicit estimate for the behavior of the perturbed eigenvalues. We use here a quantitative estimate by [12]:

▶ **Fact 13** (Stability of well-separated eigenvalues under perturbation). *Let A be a δ-separated $n \times n$ Hermitian matrix with eigenvalues $\lambda_1 > \lambda_2 > \ldots > \lambda_n$, and corresponding orthonormal basis $\{v_i\}_{\in[n]}$. Let $\mathcal{E}$ be an additive perturbation of A satisfying $|\mathcal{E}_{i,j}| \leq \varepsilon$ for all i, j. Let $\tilde{\lambda}_i$ denote the i-th eigenvalue of $A + \mathcal{E}$. There exists a constant $c > 0$ satisfying:*

$$\forall i \in [n] \quad \tilde{\lambda}_i = \lambda_i + v_i^H \mathcal{E} v_i + \zeta_i, \quad |\zeta_i| \leq c\varepsilon^2/\delta.$$

In fact, if the perturbation $\mathcal{E}$ is GUE a stronger characterization is readily available:

▶ **Corollary 14.** *Let A be a δ-separated $n \times n$ Hermitian matrix with eigenvalues $\{\lambda_i\}_{i\in[n]}$, and corresponding orthonormal basis $\{v_i\}_{\in[n]}$. Let $\mathcal{E}$ be GUE. There exists $c > 0$ independent of n such that the eigenvalues $\{\lambda'_i\}_{i\in[n]}$ of the perturbed matrix $A' = A + \varepsilon \cdot \mathcal{E}$ are distributed as follows: they are sampled from $\mu(\lambda_i, \varepsilon^2)$, and added a number ζ_i satisfying w.p. $1 - 2^{-\Omega(n)}$:*

$$|\zeta_i| \leq cn \cdot \varepsilon^2/\delta$$

Proof. By Fact 13 the eigenvalues λ'_i behave as

$$\lambda'_i = \lambda_i + v_i^H \mathcal{E} v_i + \zeta_i, \quad |\zeta_i| \leq c\varepsilon^2 \max_{i,j} |\mathcal{E}_{i,j}|^2/\delta,$$

for some constant $c > 0$. The random matrix $\mathcal{E}$ is invariant under unitary conjugation so in particular, for the unitary matrix V whose columns are the v_i's we have:

$$V^H \mathcal{E} V \sim \mathcal{E}$$

which implies

$$\lambda'_i = \lambda_i + \mathcal{E}_{i,i} + \zeta_i,$$

where

$$|\zeta_i| \leq c \max_{i,j} \left|(V^H \mathcal{E} V)_{i,j}\right|^2/\delta \sim c\varepsilon^2 \max_{i,j} |\mathcal{E}_{i,j}|^2/\delta.$$

The standard Gaussian satisfies:

$$\mathsf{P}_\mu\left(|x| \geq 4\sqrt{n}\right) \leq 2^{-2n}.$$

Thus, by the union bound we have that $|\mathcal{E}_{i,j}| \leq 4\sqrt{n}$ for all i, j w.p. at least $1 - 2^{-n}$. Hence, w.p. at least $1 - 2^{-n}$ we have:

$$\forall i \in [n] \quad |\zeta_i| \leq c \max_{i,j} |\mathcal{E}_{i,j}|^2/\delta \leq 16cn \cdot \varepsilon^2/\delta, \qquad \blacktriangleleft$$

Our interest in additive perturbation, however, is not confined just to "stability" arguments. In fact, our main reason for using perturbation is to cause a scattering of the eigenvalues. The first step of our algorithm in fact applies additive perturbation to provide a minimal spacing between eigenvalues. Recently Nguyen et al. [9] have shown that applying additive perturbation to any Hermitian matrix using a the well-known Wigner ensemble, an ensemble of random matrices that generalize GUE, in fact causes the eigenvalues of the perturbed matrix to achieve a minimal inverse polynomial separation. We state their result:

▶ **Lemma 15** ([9], Theorem 2.6. Minimal eigenvalue spacing)**.** *Let $M_n = F_n + \varepsilon \cdot X_n$, where F_n is a real symmetric matrix, $\|F_n\|_2 \leq 1$, $\varepsilon = n^{-\gamma}$ for some constant $\gamma > 0$, and X_n is GUE - namely a random Hermitian matrix (see Section 2). Let $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_n$ denote the eigenvalues of M_n, and put $\alpha_i = \lambda_i - \lambda_{i+1}$ for all $i < n$. Then for any fixed $A > 0$ there exists $B = B(A, \gamma) > 0$, such that*

$$\max_{1 \leq i < n} \mathsf{P}\left(\alpha_i \leq n^{-B}\right) = \mathrm{O}(n^{-A}).$$

In particular [3] *for any $A > 0$ there exists $B > 0$ such that $\mathsf{P}\left(\min_{1 \leq i < n} \alpha_i \geq n^{-B}\right) = 1 - \mathrm{O}(n^{-A})$.*

Using the lemma above we define the number $B^* = B^*(\delta)$ as follows:

▶ **Definition 16.** For any $\delta = 1/\mathrm{poly}(n)$, let $B^*(\delta)$ denote the smallest number $B > 0$ such that for every F_n the matrix $M_n = F_n + \delta X_n$ satisfies:

$$\mathsf{P}\left(\min_{1 \leq i < n} \alpha_i \geq n^{-B}\right) \geq 0.99$$

4 Low-Discrepancy Sequences

4.1 Basic Introduction

Low discrepancy sequences (or "quasi-random" sequences) are a powerful tool in random sampling methods. Roughly speaking, these are deterministic sequences that visit any measurable subset B a number of times that is roughly proportional to the volume of B, up to some small additive error, called the discrepancy. See definition 2.

The definition of discrepancy naturally admits an interpretation in terms of probability:

▶ **Definition 17** (Discrepancy of a random variable)**.** Let x be a random variable on $[0,1)^s$. We define the discrepancy of x, $D(x)$ as follows:

$$D(x) = \max_{S \in \mathcal{B}} \left|\mathsf{P}_z(z \in S) - \mathrm{vol}(S)\right|.$$

By definition, if x is a sequence of length N of discrepancy $D_N(x)$, and z is a uniformly random element from x, then $D(z) = D_N(x)$.

Low-discrepancy sequences have much in common with random sampling, or the Monte-Carlo method, in the sense that they visit each cube a number of time that is roughly proportional to its volume, up to a small additive error. Yet, contrary to the Monte-Carlo method, such sequences are *not* random, but only appear to be random in the sense above.

[3] applying the union bound over all eigenvalues

There are deterministic s-dimensional sequences $x = \{x_i\}_{i=1}^N$ with discrepancy as low as

$$D_N(x) \leq C \cdot \frac{\log^s N}{N},$$

and matching lower-bounds (up to constant factors) on the smallest possible discrepancy are known for $s = 1$ [10]. Hence, usually one considers low-discrepancy sequences that are very long (N) compared to the dimension (s). In particular, in this work we will focus on attaining low-discrepancy sequences for dimension $s = 2$. 2-dimensional low-discrepancy sequences can be viewed as an approximation to pairwise independent uniform random variables on the interval $[0, 1)$. This property will be crucial in proving that we are able to isolate and "filter-out" single eigenvectors, and do so in a way that does not favor any particular eigenvector (see for example Lemma 36).

We mention, in passing, that the discrepancy upper-bound decays asymptotically almost as $O(1/N)$ (assuming small dimension s) whereas for a sequence $x = \{x_n\}_n$ where the x_n's are uniform independent samples (Monte-Carlo method) the discrepancy typically decays more slowly, behaving as $O(1/\sqrt{N})$ - and hence quasi-random sequences are often preferred as a method of numerical integration.

4.2 Some basic facts

We require a Lemma [2.5] due to Niederreiter [10].

▶ **Lemma 18** ([10]. Small point-wise distance implies similar discrepancy)**.** *Let $x_1, \ldots, x_N$, $y_1, \ldots, y_N$ denote two s-dimensional sequences for which $|x_{n,i} - y_{n,i}| \leq \varepsilon$, for all $n \in [N], i \in [s]$. Then the discrepancies of these sequences are related by:*

$$|D_N(x_1, \ldots, x_N) - D_N(y_1, \ldots, y_N)| \leq s \cdot \varepsilon. \tag{1}$$

We prove an additional fact:

▶ **Fact 19** (Monotonicity of discrepancy under addition of independent random variables)**.** *Let x be a random variable on $[0, 1)^s$ of discrepancy at most $D(x)$, and let y denote the random variable*

$$y = x + z (mod 1),$$

where z supported on $[0, 1)^s$ is a random variable independent of x. Then $D(y) \leq D(x)$.

The proof appears in the full version of the paper.

4.3 The Good Seed Problem

We will be interested in sequences $x = \{x_n\}_{n=1}^N$ where each x_n is an s-dimensional vector comprised of residuals of numbers as follows:

$$x_n = \{g \cdot n\},$$

where $g \in [0, 1)^s$ is some s-dimensional vector, called the *seed* of the sequence. Specifically, in our context, the vector g will represent the vector of eigenvalues of an $n \times n$ Hermitian matrix A whose spectrum we would like to analyze. Since it is unreasonable to assume that the input matrix has a spectrum that is a good seed, we will find a perturbation of the matrix $A' = A + \mathcal{E}$ such that $g' = \mathcal{L}(A')$ has a corresponding sequence, defined as above, with low-discrepancy.

Niederreiter has shown [10] that if g is sampled uniformly on $[N]^s$ then it is a good seed with high probability:

▶ **Lemma 20.** *Let s, N be an integers and $g \sim U([N]^s)$, and let $x = \{x_n\}_n$ denote the sequence whose n-th element is given by: $x_n = \{gn/N\}$. Then*

$$\mathsf{P}\left(D_N(x) \leq \frac{\log^s N}{N}\right) \geq 1 - 1/N.$$

For our application we require that $N = \text{poly}(n)$, and $s = 2$, in which case the above discrepancy is sufficiently low for our purposes. Yet, since it requires the normalized seed g/N to be essentially uniform on $[0,1)^n$, it implies that the corresponding matrix perturbation $\mathcal{E}$ added to A must be very strong - thereby loosing all connection to the input matrix.

4.4 Finding Reasonably-Good Seeds Locally

To bridge the gap between weak-perturbation and low-discrepancy we show a new lemma, which may be of independent interest: it allows to trade-off the extent to which g is random, and the discrepancy of the sequence generated by g. Specifically, we will show that if g/N is uniform on *cubes* of much smaller side-length, i.e. at least $1/\sqrt{N}$, then the resulting sequence has discrepancy $O(\log^s N/\sqrt{N})$. This is the subject of the following lemma:

▶ **Lemma 21.** *We are given integer N, with prime divisor M and an integer s. Let $g = (g_1, \ldots, g_s) \in N^s$, such that each coordinate g_i is independently chosen uniformly on some interval $I_i \subseteq [N]$ of size M. Let $x = x(g) = \{x_n\}_{n=1}^N$ be the following s-dimensional sequence of length N corresponding to residuals of g:*

$$x_n = \left\{\frac{g \cdot n}{N}\right\}.$$

Then $\mathsf{P}_g\left(D_N(x) \leq 2\log^s(M)/\sqrt{M}\right) \geq 1 - 1/\sqrt{M}$.

The proof appears in the full version of the paper.

4.5 Low-Discrepancy from Gaussian vectors

In the previous section we showed that sampling a vector of integers uniformly from an s-dimensional cube formed by the s-th fold product of an interval $M \subseteq [N]$ yields w.h.p. a sequences of discrepancy at most $1/\sqrt{M}$. In this section we adapt these theorems about good seeds for low-discrepancy sequences to the Gaussian measure: we show that sampling a vector $g = (g_1, \ldots, g_s)$ according to the Gaussian measure (e.g. "normal vector") with variance N^{-a} yields w.h.p. a sequence of discrepancy at most N^{-b} for some positive constants a, b. The proof of this is rather technical, and hinges on an approximation of the Gaussian measure of variance σ^2 by a convex combination of uniform distributions on intervals of size $\sigma/\text{poly}(N)$.

▶ **Theorem 22** (Approximating a Gaussian by a convex sum of uniform distributions)**.** *Let $g = (g_1, \ldots, g_n)$ be a vector $g \in \mathbb{R}^n$ sampled from the standard Gaussian measure. Then g is a convex combination of two distributions $\mathcal{D}_U, \mathcal{D}_V$ as follows: $(1-p)\mathcal{D}_U + p \cdot \mathcal{D}_V$, where $\mathcal{D}_U$ is the n-fold distribution of independent variables $z_1, \ldots, z_n$, and $p \leq 2n^2/m$. Each z_i is itself a convex combination of $m \geq 2n^2$ i.i.d. variables $\{w_j\}_{j=1}^m$, with $w_j \sim U[I_j]$, where I_j is some interval of the real line of size $|I_j| = 1/m$.*

The proof of this theorem is somewhat technical and appears in the full version of the paper. We now define a vector to be "almost" normal - in the sense that it is a small perturbation of a normal vector:

▶ **Definition 23** ((σ,ε)-normal vector)**.** A random vector v is (σ,ε)-normal if it is sampled as a σ-normal vector x to which we add a vector $e = e(x)$ of length at most $\sigma\varepsilon$.

We now state our main lemma of this section - that almost normal vectors yield seeds for low-discrepancy sequences:

▶ **Lemma 24** (Low-discrepancy sequence from almost normal vectors)**.** *Let $B > 0$, and $v = (v_1,\ldots,v_n)$ be some (σ,ε)-normal vector, for $\sigma = n^{-B}, \varepsilon \le n^{-0.9B}$. There exists $M \le n^{1.6B}$ such that for any $S = \{i_1,\ldots,i_s\} \subseteq [n]$, $|S| = s$ the distribution on s-dimensional sequence of length M:*

$$V_s \equiv \{(\{m\cdot v_{i_1}\},\ldots,\{m\cdot v_{i_s}\})\}_{m\in[M]}$$

satisfies $D_M(V_s) \le 4\log^s(n)\cdot n^{-0.1B}$.

Proof. Let P be the minimal prime which is at least $n^{0.3B}$, and put $M = P^5$. By Bertrand's postulate, for sufficiently large n we have that $M = P^5 \le n^{1.51B} \le n^{1.6B}$. For any $z \in [0,1)$ let z^M be the number closest to z in the grid m/M, $m \in [M]$.

Removal of non-independent component

Since v is (σ,ε)-normal then $v_i = X_i + Y_i$, where $X_i \sim (\eta_i,\sigma^2), |Y_i| \le \varepsilon\sigma$, and the X_i's are independent. Let V_S^X denote the sequence generated by taking only the X component of the seed vector v, i.e.:

$$V_S^X \equiv \{(\{m\cdot X_{i_1}\},\ldots,\{m\cdot X_{i_s}\})\}_{m\in[M]} \tag{2}$$

▶ **Fact 25.**

$$D_M(V_S) \le D_M(V_S^X) + s\cdot n^{-0.2B}$$

Proof. Consider the r.v.'s X_i, Y_i. By our assumption

$$\forall i\in[n] \quad |Y_i| \le \sigma\varepsilon = n^{-1.9B}. \tag{3}$$

Thus the difference between the residuals of v_i and X_i are small modulo 1:

$$\forall m\in[M],\ i\in[n] \quad |\lceil\{mv_i\} - \{mX_i\}\rfloor| \le m\cdot n^{-1.9B} \le Mn^{-1.9B} \le n^{-0.3B} \tag{4}$$

By Lemma 18, we can conclude that the discrepancy of our target sequence V_S follows tightly the discrepancy of V_S^X:

$$D_M(V_S) \le D_M(V_S^X) + s\cdot n^{-0.3B} \tag{5}$$

◀

Reducing Gaussian measure to uniform measure

Consider the vector derived by truncating each coordinate of the vector $(X_{i_1},\ldots,X_{i_s})$ to the nearest point on the M-grid:

$$X^M = (X_{i_1}^M,\ldots,X_{i_s}^M) = (\lfloor MX_{i_1}\rceil/M,\ldots,\lfloor MX_{i_s}\rceil/M).$$

Consider the discrepancy of the distribution on s-dimensional sequences formed by taking integer multiples of X^M. We claim:

▶ **Fact 26.**

$$\mathsf{P}_v\left(D_M(V_S^{X,M}) \le \log^s(n)\cdot n^{-0.1B}\right) \ge 1-3n^{-0.1B},$$

Proof. In Fact 22 choose as parameter $m = n^{0.2B+2}$. We get that w.p. at least $1-2n^2/m = 1-2n^{-0.2B}$ each X_i samples a convex mixture of variables $\{w_j\}_{j\in[m]}$ where

$$w_j \sim U(I_j), |I_j| = \sigma/m = n^{-1.2B-2} \tag{6}$$

Hence, w.p. at least $1-2n^{-0.2B}$ for all $i\in[n]$, the variable $M\cdot\{X_i^M\}$ is a convex mixture of uniform random variables on intervals $M\cdot I_j \subseteq [M]$, where

$$|M\cdot I_j| \ge \frac{\sigma M}{m} \ge n^{1.5B}\cdot n^{-1.2B-2} \ge M^{0.2}. \tag{7}$$

We apply Lemma 21 to the sequence of residuals of integer multiples, with the seed X^M:

$$V_S^{X,M} \equiv \left(\left\{mX_1^M\right\},\ldots,\left\{mX_s^M\right\}\right)_{m\in[M]}. \tag{8}$$

The lemma requires that each variable be distributed as: $MX_i^M \sim U[\mathcal{I}]$, where $\mathcal{I}$ is some interval of $[M]$, for integer $M>1$ satisfying: $|\mathcal{I}|\ge P,\quad P$ prime , $\quad P|N$. By our choice of parameters M has a prime divisor P equal to $M^{0.2}=P$. Hence, by Equation 7 we can satisfy the assumption of the lemma by choosing the parameters for Lemma 21 as follows: $N=M, M=P$. Hence, by Lemma 21, and accounting for the Gaussian-to-uniform approximation error we get:

$$\mathsf{P}_v\left(D_M(V_S^{X,M}) \le 2\log^s(n)\cdot n^{-0.1B}\right) \ge 1-n^{-0.1B}-2n^{-0.2B} \ge 1-3n^{-0.1B}. \tag{9}$$

◀

Treating the residual w.r.t. the M-grid

Define: the truncation error

$$\forall i\in[s]\quad r_i := X_i - X_i^M.$$

In Fact 25 we analyzed the error Y_i whose magnitude is negligible even w.r.t. $1/M$, and can thus be disregarded for any element of the sequence V_S. Unlike this, the residual error r_i cannot be disregarded because when multiplied by integers uniformly in $[M]$ it assumes magnitude $\Omega(1)$. Thus, it requires a different treatment.

▶ **Corollary 27.**

$$\mathsf{P}_v\left(D_M(V_S^X) \le 2\log^s(n)\cdot n^{-0.1B}\right) \ge 1-4n^{-0.1B}$$

Proof. Express the i-th element of the sequence using r_i:

$$\forall i\in[s]\quad \{X_i\cdot m\} = \left\{(X_i^M+r_i)\cdot m\right\} = \{\{mX_i^M\}+\{mr_i\}\} \tag{10}$$

The variable $V_S^{X,M}$ is the distribution on s-dimensional vectors formed by sampling the initial seed $\{X_i^M\}_{i\in[s]}$, a uniform random m and returning $(\{mX_1^M\},\ldots,\{mX_s^M\})\in[0,1)^s$. Hence, the variable $y\sim V_S^X$ can be written as

$$y = x + z(\mathrm{mod}1)$$

where $x \sim V_S^{X,M}$ and $z \sim \{(mr_1, \ldots, mr_s)\}$, where $m \sim U[M]$.

Let E denote the event in which X_i is sampled according to $w_j \sim U[\mathcal{I}_j]$ where w_j is at distance at least $1/M$ from either one of the edges of $\mathcal{I}_j$. Conditioned on E, the random variables r_i and X_i^M are independent for all $i \in [s]$. By the above, x and z are independent conditioned on E. Hence, we can invoke Fact 19 w.r.t. y. By this fact we have:

$$D_M(V_S^X|E) \leq D_M(V_S^{X,M})$$

and so by Fact 26

$$\mathsf{P}_v\left(D_M(V_S^X|E) \leq \log^s(n) \cdot n^{-0.1B}\right) \geq 1 - 3n^{-0.1B}, \tag{11}$$

By Equation 7 the probability of E is at least:

$$\mathsf{P}_v(E) \geq 1 - |\mathcal{I}_j|/(2M) \geq 1 - M^{0.2}/(2M) \geq 1 - n^{-B}.$$

Thus: $\mathsf{P}_v\left(D_M(V_S^X) \leq \log^s(n) \cdot n^{-0.1B}\right) \geq 1 - 3n^{-0.1B} - \mathsf{P}(E) \geq 1 - 4n^{-0.1B}$. ◀

Conclusion of proof: By Corollary 27 we have

$$\mathsf{P}_v\left(D_M(V_S^X) \leq 2\log^s(n) \cdot n^{-0.1B}\right) \geq 1 - 4n^{-0.1B}$$

and by Fact 25 we have

$$D_M(V_S) \leq D_M(V_S^X) + s \cdot n^{-0.2B}$$

Thus by the union bound:

$$\mathsf{P}_v\left(D_M(V_S) \leq 2\log^s(n) \cdot n^{-0.1B} + s \cdot n^{-0.2B}\right) \geq 1 - 4n^{-0.1B}$$

thus:

$$\mathsf{P}_v\left(D_M(V_S) \leq 3\log^s(n) \cdot n^{-0.1B}\right) \geq 1 - 4n^{-0.1B}$$

Hence for all but a measure $4n^{-0.1B}$ of sampled vectors v, the resulting sequence has discrepancy at most $3\log^s(n)n^{-0.1B}$. Since the discrepancy measures the worst-case additive error for any set this implies that:

$$D_M(V_S) \leq 3\log^s(n)n^{-0.1B} + 4n^{-0.1B} \leq 4\log^s(n)n^{-0.1B}$$ ◀

5 A Filtering Algorithm

In this section we provide the specification of the filtering algorithm, which is the main computational black box of our algorithm. This algorithm accepts an integer m that separates the i-th eigenvalue of a Hermitian matrix A and computes an approximation for the i-th eigenvector, with high probability:

Algorithm 2 Filter(A, m, δ)

1. Compute parameters: $p = 2n^2\lceil \ln(1/\delta)\rceil, \zeta = \delta^2/(2pm)$.
2. **Sample random unit vector:**
   Sample a standard complex Gaussian vector v, set $w_0 = v/\|v\|$.
3. **Approximate matrix exponent:**
   Compute a ζ Taylor-series approximation of $e^{2\pi iA}$, denoted by $\tilde{U}$.
4. **Raise to power:**
   Compute $\tilde{U}^m$ by repeated squaring.
5. **Generate matrix polynomial:**
   Compute $B = \left(\frac{I+\tilde{U}^m}{2}\right)^p$ by repeated squaring.
6. **Filter:**
   Compute $w = \frac{B\cdot w_0}{\|B\cdot w_0\|}$.
7. **Decide:**
   Set $z = A\cdot w$, $i_0 = \arg\max_{i\in[n]} |w_i|$ and compute $c = z_{i_0}/w_{i_0}$. If

   $$\|A\cdot w - c\cdot w\| \le 3\delta\sqrt{n}$$

   return w, and otherwise reject.

We now show that if the algorithm is provided with an integer m that separates the k-th eigenvalue of A in the sense defined in Definition 6, then the output is close to the k-th eigenvector of A.

▶ **Theorem 28.** *Let n be some integer, $\delta \le n^{-10}$ and $\alpha = 3\sqrt{\ln(1/\delta)}$. We are given an $n\times n$ Hermitian matrix A with eigenvalues $\{\lambda_i\}_{i\in[n]}$. Additionally, we are provided an integer m that separates k in A, w.r.t. $B_{in}(\alpha), B_{out}$, in the sense of Definition 5. Let $w = Filter(A, m, \delta)$. Then*

$$\mathsf{P}\left(\|w - v_k\| \le \delta\right) \ge 1 - 3n^{-3},$$

for some unit eigenvector v_k of λ_k, and sufficiently large n. The algorithm has boolean complexity $O(n^\omega \cdot \log(2p^2m^2/\delta^2))$, and runs in parallel time $O(\log^2(n))$.

Proof. Let $\{\tau_\ell\}_{\ell\in[n]}$ denote the set of eigenvalues of $\tilde{U}$. Since $\tilde{U}$ is a polynomial in A (truncated Taylor series) then $\{v_\ell\}_{\ell\in[n]}$ is also an orthonormal basis for $\tilde{U}$. Since in addition $\|\tilde{U} - e^{2\pi iA}\| \le \zeta$ then

$$\forall \ell\in[n] \quad \left|\tau_\ell - e^{2\pi i\lambda_\ell}\right| \le \zeta. \tag{12}$$

Let $w' = B\cdot w_0$ and denote $w_0 = \sum_{\ell\in[n]}\beta_\ell v_\ell$, and $w' = \sum_{\ell\in[n]}\alpha_\ell v_\ell$. Since $A, \tilde{U}$ share the same basis of eigenvectors, then by the definition of the matrix B the coefficients α_ℓ, β_ℓ are related by:

$$|\alpha_\ell|^2 = |\beta_\ell|^2 \cdot \left|\frac{1+{\tau_\ell}^m}{2}\right|^{2p}.$$

So by Equation 12

$$\frac{|\alpha_\ell|^2}{|\beta_\ell|^2} \ge \left|\frac{1+e^{2\pi im\lambda_\ell}}{2}\right|^{2p} - 2pm\zeta = |\cos(2\pi m\lambda_\ell/2)|^{2p} - 2pm\zeta$$

Since m separates k then $\{m\lambda_k\} \in B_{in}$, and for all $\ell \neq k$ we have $\{m\lambda_\ell\} \notin B_{out}$. Thus, for $\ell = k$:

$$\frac{|\alpha_k|^2}{|\beta_k|^2} \geq \left|\cos(2\pi/6n\sqrt{\ln(1/\delta)})\right|^{2p} - 2pm\zeta$$

Using Claim 32

$$\geq \left(1 - \frac{1}{n^2 \ln(1/\delta)}\right)^{2p} - 2pm\zeta \geq \frac{1}{2e^4}. \tag{13}$$

On the other hand, for all $\ell \neq k$ we have:

$$\frac{|\alpha_\ell|^2}{|\beta_\ell|^2} \leq \left|\frac{1 + e^{2\pi i m\lambda_\ell}}{2}\right|^{2p} + 2pm\zeta.$$

so since m separates k then the above is at most:

$$\leq |\cos(2\pi/2n)|^{2p} + 2pm\zeta$$

which by Claim 32 is at most:

$$\leq (1 - \pi^2/(3n^2))^{2n^2 \ln(1/\delta)} + 2pm\zeta \leq e^{-2\ln(1/\delta)} + 2pm\zeta \leq 2\delta^2. \tag{14}$$

By Fact 31 for any $\varepsilon = 1/\mathrm{poly}(n)$ there exists a constant $c > 0$ such that

$$\mathsf{P}(\forall i,j \quad |\beta_j| \leq c|\beta_i|\sqrt{\ln(1/\varepsilon)}/\varepsilon) \geq 1 - 3n\varepsilon.$$

Choose $\varepsilon = n^{-4}$. Then by Equations 13 and 14:

$$\mathsf{P}\left(\forall \ell \neq k \quad \frac{|\alpha_\ell|^2}{|\alpha_k|^2} \leq c^2(2\delta^2) \cdot (4e^8) \cdot 4\ln n \cdot n^8\right) \geq 1 - 3n^{-3}.$$

and so for $\delta \leq n^{-10}$ there exists $\eta \in \mathbb{C}$, $|\eta| = 1$ such that

$$\left\| \frac{w'}{\|w'\|} - \eta \cdot v_k \right\|^2 \leq \frac{1}{|\alpha_k|^2} \sum_{j \neq k} |\alpha_j|^2 \leq 32c^2 n^9 \ln n \delta^2 e^8 < \delta.$$

for sufficiently large n. Using Claim 30 we conclude that w.p. at least $1 - 3n^{-3}$ over choices w_0, the criterion is met and the algorithm returns a vector $w = w'/\|w'\|$ satisfying the equation above.

Arithmetic run-time: The approximation of $e^{2\pi i A}$ by $\tilde{U}$ requires, using Fact 33 a time at most

$$O(n^\omega \log(1/\zeta)) = O(n^\omega \cdot \log(2pm/\delta^2)).$$

Next, the repeated powering of $\tilde{U}$ to a power m requires time at most: $O(n^\omega \lceil \log(m) \rceil)$ and the repeated powering of B to the power p requires time at most: $O(n^\omega \lceil \log(p) \rceil)$ Hence the total complexity is : $O(n^\omega \cdot \log(pm/\delta^2))$.

Depth complexity: Each matrix product can be carried out in depth $\log(n)$. Each of steps 3 to 6 involves at most $\log(m)+\log(p)$ sequential matrix multiplications. Hence the depth complexity of the entire circuit is at most $\log(n)\cdot(\log(m)+\log(p))+O(\log(n))=O(\log^2(n))$.

We conclude the proof of the theorem by showing stability:

▶ **Claim 29.** *Under the assumption of Theorem 28 the algorithm is log-stable.*

Proof. Consider the arithmetic operations involved in computing the filtering algorithm:

1. Generating an approximation $\tilde{U}$ of $e^{2\pi iAm}$ as a truncated Taylor series.
2. Raising $\tilde{U}$ to a power $m\in[M]$.
3. Computing $((I+\tilde{U})/2)^p$.
4. Normalizing $Bw_0/\|Bw_0\|$.

Consider an arithmetic circuit C implementing the above, and the circuit $D=D(C,t)$ - the discretization of C to t bits of precision modeled as follows: after each arithmetic step, the result is rounded to the nearest value of 2^{-t}. Consider all steps except division. A is δ-separated so in particular $\|A\|\le 1$. Thus, whenever we multiply two matrices at any of the steps above both have norm at most 1. Hence, at each rounding step the error is increased by at most $\sqrt{n}2^{-t}$. Finally, considering the final division step, we observe that since m separates k, then by Equation 13 we have $\|Bw_0\|\ge 1/(2e^6)$. This implies that the total error is at most $\sqrt{n}(p+M)\cdot 2^{-t}\cdot 2e^6$. Since M,p are both polynomial in n then for any $\delta=1/\mathrm{poly}(n)$ the error is at most δ for some $t=O(\log(1/\delta))$. ◀

◀

5.1 Supporting Claims

We now state the important supporting claims. Their proofs appear in the full version of the paper.

▶ **Claim 30.** *Let A be some $n\times n$ Hermitian matrix, $\|A\|\le 1$. Suppose that $\|w-v\|\le\delta$ for some unit eigenvector v of A, and $\delta\le 1/4$. Let $z=A\cdot w$, and i_0 denote $i_0=\arg\max_{i\in[n]}|w_i|$. Let $c=z_{i_0}/w_{i_0}$. Then*

$$\|A\cdot w-c\cdot w\|\le 3\delta\sqrt{n}.$$

▶ **Fact 31** (Random unit vectors have well-balanced entries)**.** *Let $\{v_i\}_{i\in[n]}$ be some orthonormal basis of $\mathbb{C}^n$, $0<\varepsilon=1/\mathrm{poly}(n)$, and $v\in\mathbb{C}^n$ a uniformly random complex unit vector. For any $i\in[n]$ let $\alpha_i=|\langle v,v_i\rangle|$. For any $\varepsilon=1/\mathrm{poly}(n)$ there exists a number $c_1>0$ independent of n, such that*

$$\mathsf{P}\left(\forall i,j\quad |\alpha_i|/|\alpha_j|\le c_1\sqrt{\ln(1/\varepsilon)}/\varepsilon\right)\ge 1-3n\varepsilon.$$

▶ **Claim 32.** $\forall\theta\in[-0.01,0.01]\quad 1-\frac{\theta^2}{2}\le\frac{1+\cos(\theta)}{2}\le 1-\frac{\theta^2}{3}$.

▶ **Fact 33** (Efficient approximation of exponentiated matrix)**.** *Given a Hermitian $n\times n$ matrix A, $\|A\|\le 1/(2\pi)$, and error parameter $\varepsilon>0$, a Taylor approximation of $e^{2\pi iA}$, denoted by $\tilde{U}_A$ can be computed in time $O(n^\omega\log(1/\varepsilon))$ and satisfies $\left\|e^{2\pi iA}-\tilde{U}_A\right\|\le\varepsilon$.*

6 Sampling Separating Integers

In this section we show our main technical tool: which is that perturbing a δ-separated Hermitian matrix A by a Gaussian matrix of a carefully calibrated variance, results in a corresponding sequence of residuals $S(A)$ having low-discrepancy, at least for 2-dimensional sequences - i.e. pairs of variables. This, in turn, implies that we can separate each eigenvalue of A almost uniformly:

▶ **Theorem 34.** *Let A be a δ-separated $n \times n$ PSD matrix, $\mathcal{E}$ GUE, $\zeta \leq \min\{\delta^{13}, n^{-50}\}$, and $\alpha > 4$. For any $M \geq \zeta^{-1.6}$ we have:*

$$\forall k \in [n] \quad \mathsf{P}_{\mathcal{E}, m\sim U[M]}\left(\text{ } m \text{ separates } k \text{ in } A + \zeta \cdot \mathcal{E} \text{ w.r.t. } B_{in}(\alpha), B_{out} \text{ } \right) \geq 1/(5\alpha n)$$

6.1 Additive Perturbation

By our definitions above, Gaussian perturbation of a matrix with well-separated eigenvalues results in a (σ, ε)-normal vector as follows:

▶ **Fact 35** (Perturbation of well-separated matrices)**.** *Let A be an $n \times n$ ε-separated Hermitian matrix with eigenvalues $\lambda_1 \geq \lambda_2 \ldots \geq \lambda_n$. Let $\mathcal{E}$ be GUE, and $A' = A + \varepsilon^L \cdot \mathcal{E}$, where $L \geq 2$. Then w.p. at least $1 - n \cdot 2^{-n}$ the vector of eigenvalues of A' $(\lambda'_1, \ldots, \lambda'_n)$ is $(\varepsilon^L, cn\varepsilon^{L-1})$-normal, for some constant $c > 0$.*

Proof. Invoke Corollary 14 choosing ε as ε^L and δ as ε, and take the union bound over all $i \in [n]$. ◀

6.2 Approximate Pairwise Independence

▶ **Lemma 36.** *Let $\bar{\lambda} = (\lambda_1, \ldots, \lambda_n) \in [0,1)^n$ and M a positive integer that satisfy:*

$$\forall i \neq j \quad D_M\left(\{(m\lambda_i, m\lambda_j)\}_{m\in[M]}\right) \leq \zeta, \quad \zeta \leq n^{-4}$$

Let $\alpha > 4$. For each $k \in [n]$ w.p. at least $1/(5\alpha n)$ over choices of $m \sim U[M]$ the sampled sequence m separates k w.r.t. $B_{in}(\alpha), B_{out}$.

Proof. Fix $x_i = \{m\lambda_i\}$ and let E_i denote the following event:

$$E_i := (x_i \in B_{in}) \wedge (\forall j \neq i \;\; x_j \notin B_{out})$$

We want to show that

$$\forall i \in [n] \;\; \mathsf{P}(E_i) \geq \frac{1}{5\alpha n}.$$

Let t denote the number of x_j's in B_{out}:

$$t = |\{j \;\; j \neq i \;\; x_j \in B_{out}\}|$$

Then under this notation we have:

$$\mathsf{P}(E_i) = \mathsf{P}(t = 0 \wedge x_i \in B_{in}). \tag{15}$$

Consider the conditional expectation: $\mathbf{E}\left[t | x_i \in B_{in}\right]$ By linearity of expectation:

$$\mathbf{E}\left[t | x_i \in B_{in}\right] = \sum_{j\neq i} \mathsf{P}\left[x_j \in B_{out} | x_i \in B_{in}\right]. \tag{16}$$

Considering each summand separately:

$$\mathsf{P}(x_j \in B_{out} | x_i \in B_{in}) = \frac{\mathsf{P}(x_j \in B_{out} \wedge x_i \in B_{in})}{\mathsf{P}(x_i \in B_{in})}$$

Using the pairwise discrepancy assumption, the above is at most:

$$\frac{|B_{out}| \cdot |B_{in}| + \zeta}{|B_{in}| - \zeta} \leq |B_{out}| + 2\zeta\alpha n = \frac{1}{2n} + 2\alpha\zeta n \leq \frac{0.51}{n}$$

and so by Equation 16

$$\mathbf{E}\left[t | x_i \in B_{in}\right] = (n-1) \cdot \mathsf{P}\left[x_j \in B_{out} | x_i \in B_{in}\right] \leq 0.51.$$

The variable $t | x_i \in B_{in}$ accepts only integral values, and by Markov's inequality:

$$\mathsf{P}(t \geq 1 | x_i \in B_{in}) \leq 0.51$$

Therefore $\mathsf{P}(t = 0 | x_i \in B_{in}) \geq 0.49$. Using again the 1-dimensional discrepancy we have

$$\mathsf{P}(x_i \in B_{in}) \geq \frac{1}{\alpha n} - \zeta \geq \frac{1}{2\alpha n}.$$

Substituting the last two inequalities into Equation 15 yields: $\mathsf{P}(E_i) \geq 0.49 \cdot \frac{1}{2\alpha n} \geq \frac{1}{5\alpha n}$. ◀

6.3 Proof of Theorem 34

Proof. By assumption A is δ-separated and $\zeta \leq \min\{n^{-50}, \delta^{13}\}$. Consider the perturbed matrix $A' = A + \zeta\mathcal{E}$. Choose $L = 13$ and $\varepsilon = \zeta^{1/13}$. By Fact 35 there exists some constant $c > 0$ such that w.p. at least $1 - n2^{-n}$ the vector $\mathcal{L}(A')$ is (ζ, ε)-normal with parameters

$$\zeta \leq n^{-50}, \varepsilon \leq cn\zeta^{12/13} \leq \zeta^{0.9}$$

where the last inequality follows because $\zeta \leq n^{-50}$. We assume that this is the case and account for the negligible error at the end. Set now $B = \log_n(1/\zeta)$. Then the eigenvalues of A' are (σ, ε)-normal for $\sigma = n^{-B}$ an $\varepsilon \leq n^{-0.9B}$. Since in addition $\alpha > 4$ then by Lemma 24 there exists an integer $M \leq n^{1.6B}$ satisfying:

$$\forall S \subseteq [n],\ |S| = s \quad D_M(\{m\lambda_S\}) \leq 4\log^s(n)n^{-5} \leq n^{-4}, \tag{17}$$

for sufficiently large n. Hence, by Lemma 36 a random $m \sim U[M]$ separates the k-th eigenvalue of $A + \mathcal{E}$ w.r.t. $B_{in}(\alpha), B_{out}$ w.p at least $1/(5\alpha n)$. ◀

7 Parallel Algorithm for ASD

The algorithm Filter(A, m, δ) described in Section 5 is given an integer m that separates the i-th eigenvalue, and returns an approximation for the i-th eigenvector. In this section, we use this algorithm in a black-box fashion and design a Las-Vegas algorithm for computing the full ASD of a matrix. Essentially, it amounts to running sufficiently many copies of the filtering algorithm in parallel so that all eigenvectors are collected as "coupons" with high probability.

Algorithm 3

Input: $n \times n$ Hermitian matrix A, parameter δ.

1. **Initialize:**
   a. Recall the definition of B^* in Definition 16 and compute parameters: $\delta = \min\{\delta, n^{-10}\}$, $B = \min\{\delta, B^*(\delta/(3\sqrt{n}))\}$, $\delta' = (\min\{\delta, B\})^{13}/4$, $\alpha = \sqrt{\ln(1/\delta')}$, $M = (\max\{B^{-12}, n^{-50}\})^{1.6}$, $\mathcal{T} = 60n\alpha\log(n)$.
   b. Perturb A with GUE matrix $\mathcal{E}_1$: $A_0 := A + \mathcal{E}_1 \cdot \delta/(3n)$
2. **Collect all eigenvectors:**
   Run $\mathcal{T}$ parallel processes of the following procedure
   a. Perturb A_0: $A_1 = A_0 + \delta' \cdot \mathcal{E}_2$, for GUE matrix $\mathcal{E}_2$.
   b. Sample $m \sim U[M]$
   c. Run Filter (A_1, m, δ') and store output w.
3. **Generate database:**
   a. For vector $w = w_k$ sampled at process $i \in [\mathcal{T}]$, compute $z = A \cdot w$, $i_0 = \arg\max_{i \in [n]} |w_i|$ and $\tilde{\lambda}_k = z_{i_0}/w_{i_0}$.
   b. Sort the values $\tilde{\lambda}_i$: assume $\tilde{\lambda}_1 \leq \ldots \leq \tilde{\lambda}_{\mathcal{T}}$. Initialize: $\gamma = \tilde{\lambda}_1$, $\mathcal{D} = \emptyset$. Iterate over all $i = 1, \ldots, \mathcal{T}$. At each step i: if $|\gamma - \tilde{\lambda}_i| \geq B/4$ then add $\mathcal{D} \to \mathcal{D} \cup \{w_i\}$, and set $\gamma = \tilde{\lambda}_i$.

Overview:

The first step of the algorithm adds a "coarse" perturbation to A to make sure that it has n unique eigenvalues that are well-separated. The second step is essentially a parallel execution of the Filter() procedure where each call to this sub-routine uses independent random bits to add a "fine" perturbation to A. This implies that each process samples independently and uniformly an approximation of the k-th eigenvector of A, for each $k \in [n]$. The final step merely builds up a database of approximate eigenvectors so that all eigenvectors are represented exactly once.

We now state our main theorem the proof of which appears in the full version of the paper:

▶ **Theorem 37.** *For any $n \times n$ Hermitian matrix $0 \preceq A \preceq 0.9I$, and $\delta = 1/\mathrm{poly}(n)$ there exists an $\mathrm{RNC}^{(2)}$ algorithm computing $\mathrm{ASD}(A, \delta)$, in boolean complexity $\tilde{O}(n^{\omega+1})$. The algorithm is log-stable.*

Acknowledgements. The authors thank Naomi Kirshner, Robin Kothari, Yosi Atia, and anonymous reviewers for their helpful comments and suggestions.

References

1 Selim G. Akl. *Parallel Sorting Algorithms*. Academic Press, Inc., Orlando, FL, USA, 1990.
2 F. L. Bauer and C. T. Fike. Norms and exclusion theorems. *Numerische Mathematik*, 2(1):137–141, Dec 1960. doi:10.1007/BF01386217.
3 Dario Bini and Victor Y. Pan. Practical improvement of the divide-and-conquer eigenvalue algorithms. *Computing*, 48(1):109–123, 1992. doi:10.1007/BF02241709.
4 James Demmel, Ioana Dumitriu, and Olga Holtz. Fast linear algebra is stable. *Numerische Mathematik*, 108(1):59–91, 2007. doi:10.1007/s00211-007-0114-x.

5 Gene H. Golub and Charles F. Van Loan. *Matrix Computations (3rd Ed.)*. Johns Hopkins University Press, Baltimore, MD, USA, 1996.

6 Aram W. Harrow, Avinatan Hassidim, and Seth Lloyd. Quantum algorithm for linear systems of equations. *Phys. Rev. Lett.*, 103:150502, Oct 2009. doi:10.1103/PhysRevLett.103.150502.

7 Dexter C. Kozen. *The Design and Analysis of Algorithms*. Springer-Verlag New York, Inc., New York, NY, USA, 1992.

8 Mauro Leoncini, Giovanni Manzini, and Luciano Margara. Parallel complexity of numerically accurate linear system solvers. *SIAM J. Comput.*, 28(6):2030–2058, 1999. doi:10.1137/S0097539797327118.

9 Hoi Nguyen, Terence Tao, and Van Vu. Random matrices: tail bounds for gaps between eigenvalues. *Probability Theory and Related Fields*, pages 1–40, 2016. doi:10.1007/s00440-016-0693-5.

10 H. Niederreiter. *Random Number Generation and Quasi-Monte Carlo Methods*. Society for Industrial and Applied Mathematics, 1992. doi:10.1137/1.9781611970081.

11 John H. Reif. Efficient parallel factorization and solution of structured and unstructured linear systems. *Journal of Computer and System Sciences*, 71(1):86–143, 2005.

12 Gilbert W. Stewart and Jiguang Sun. *Matrix perturbation theory*. Computer science and scientific computing. Academic Press, Boston, 1990.

13 G.W. Stewart. Pertubation bounds for the definite generalized eigenvalue problem. *Linear Algebra and its Applications*, 23:69–85, 1979.

14 Amnon Ta-Shma. Inverting well conditioned matrices in quantum logspace. 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 881–890. ACM, 2013. doi:10.1145/2488608.2488720.

15 Lloyd N. Trefethen and David Bau. *Numerical linear algebra*. Society for Industrial and Applied Mathematics, Philadelphia, 1997.

16 Virginia Vassilevska Williams. Multiplying matrices faster than coppersmith-winograd. 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 887–898. ACM, 2012. doi:10.1145/2213977.2214056.

Non-Negative Sparse Regression and Column Subset Selection with L_1 Error

Aditya Bhaskara[*1] **and Silvio Lattanzi**[2]

1 **School of Computing, University of Utah, Salt Lake City, UT, USA**
bhaskara@cs.utah.edu
2 **Google Research, Zurich, Switzerland**
silviol@google.com

Abstract

We consider the problems of sparse regression and column subset selection under ℓ_1 error. For both problems, we show that in the non-negative setting it is possible to obtain tight and efficient approximations, without any additional structural assumptions (such as restricted isometry, incoherence, expansion, etc.). For sparse regression, given A, b with non-negative entries, we give an efficient algorithm to output a vector x of sparsity $O(k)$, for which $\|Ax - b\|_1$ is comparable to the smallest error possible using non-negative k-sparse x. We then use this technique to obtain our main result: an efficient algorithm for column subset selection under ℓ_1 error for non-negative matrices.

1998 ACM Subject Classification F.2.0 Analysis of Algorithms and Problem Complexity

Keywords and phrases Sparse regression, L1 error optimization, Column subset selection

Digital Object Identifier 10.4230/LIPIcs.ITCS.2018.7

1 Introduction

Sparsity plays a crucial role in learning and signal processing. Representing a signal as a sparse combination of "elementary" signals (sparse recovery) and finding bases in which a collection of signals have sparse representation (sparse coding) are fundamental problems, with applications ranging from genetics, to speech processing, to computer vision [12, 33, 31, 32, 38, 40, 41].

In most of these applications, we require recovery algorithms that are tolerant to noise. Different "types" of noise lead to different optimization formulations. For instance, if signals are corrupted under independent Gaussian noise, recovery algorithms with an ℓ_2 objective perform well. On the other hand, when there are a few yet large deviations, recovery algorithms with an ℓ_1 objective perform much better (cf. [13, 26, 42]). This is also the case for problems involving matrices, such as low rank approximation and column subset selection, in which we could either have uniform noise in all the columns, or have a few "outlier" columns with large error. Also, from a theoretical perspective, it is interesting to ask the approximation question for general ℓ_p norms. Interestingly, for the matrix problems above, minimizing the ℓ_1 error (and more generally ℓ_p error for $p \neq 2$) turns out to be significantly harder than minimizing the ℓ_2 error. The beautiful theory of singular value decompositions that lets us minimize ℓ_2 error does not have a counter-part for ℓ_1. Indeed, even finding approximate solutions has recently been shown to be NP hard [36]. However, given its effectiveness in applications ([31, 32, 38, 41]), many heuristics [27, 37, 43, 11, 20] and, more recently, approximation algorithms [15, 36] have been proposed.

* Partially supported by a Google Research Faculty Award.

Our main focus in this paper is the well-studied problem of *column selection*, with the goal of minimizing the ℓ_1 reconstruction error. The column selection problem is the following: we are given a matrix A ($m \times n$), and the goal is to find a subset S of its columns of a prescribed size, so as to minimize the "reconstruction error" $\min_{X \in \mathbb{R}^{|S| \times n}} \|A - A_S X\|_1$, where A_S is the submatrix of A restricted to the columns S. Besides its direct applications [12, 31, 32, 38, 40, 41] column selection has found several applications as a tool for building efficient machine learning algorithms, including feature selection (as a pre-processing step for learning), coreset computation, and more broadly, as *interpretable* dimension reduction [1, 16, 21].

While column selection has been quite well understood with ℓ_2 error (in a sequence of works, including [10, 23, 25, 34]), it is computationally much harder under ℓ_1 error. Recently, Song et al. [36] provided the first CUR and low rank approximation algorithm for general matrices and for $p \in [1, 2]$. Their main result is an $(O(\log m)\,\mathrm{poly}(k))$-approximation to the error in $\mathrm{nnz}(A) + (n + m)\,\mathrm{poly}(k)$ time, for every k, where $\mathrm{nnz}(A)$ is the number of non-zero entries in A. They also prove that if one compares the error obtained by the column approximation with the best rank-k approximation, a factor of $\sqrt{k}$ is inevitable, even if one uses significantly more than k columns. Our result is incomparable in two respects: first, it gets past the lower bound by comparing the solution obtained with the error of the best k-column approximation. Second, our focus is on the non-negative setting (defined below), for which we obtain a significantly better approximation. A more detailed comparision with the works of Song et al. [36] and Chierichetti et al. [15] is provided in Section 1.2.

It is important to note that the non-negativity assumption is natural, in fact in many applications where column selection is used to "explain" a collection of points in terms of a smaller subset, we have that all the points are expressible as a *non-negative* (often even convex) combination of the selected points. Further, in applications such as recommender systems and image reconstruction, the points themselves have coordinates that are all non-negative. Motivated by such applications, we define the *non-negative column subset selection* problem, as follows.

▶ **Problem 1** (Non-negative CSS(A, B))**.** *Given two matrices A, B with non-negative entries and parameter k, find a subset S of the columns of A of size k, so as to minimize the best reconstruction error for B, i.e., the quantity* $\min_{X \geq 0} \|B - A_S X\|_1$. *($X \geq 0$ refers to entry-wise non-negativity.)*

Note that this is a slight generalization of column selection as explained earlier (which can be viewed as the case $B = A$). Our main result in this paper is an efficient algorithm for this problem; enroute to this we also design an algorithm for the ℓ_1 sparse regression problem.

Sparse regression

The key ingredient in our result is a new algorithm for another fundamental problem – sparse regression. Given a matrix A, and a target vector b, the goal here is to find a sparse x such that $\|Ax - b\|_1$ is as small as possible. The classic paper of Donoho [19], and the beautiful line of work [5, 14, 24, 30] have resulted in algorithms for approximate recovery, under special assumptions on A (for instance, the so-called restricted isometry property (RIP), or expansion properties of an appropriate graph associated with A). Here we solve the general version of the non-negative problem.

It is also known that sparse recovery is hard if we do not make assumptions on A [2, 22] (incidentally, even checking if a matrix satisfies the RIP condition is hard, in many interesting parameter ranges [6, 28, 39]). So our focus here is on a non-negative variant, defined as follows.

▶ **Problem 2** (Non-negative sparse regression). *Given a non-negative matrix A, a non-negative* target *vector b, and a parameter k, find a vector $x \geq 0$ (entry-wise) of sparsity k that minimizes $\|Ax - b\|_1$.*

Finding sparse solutions to optimization problems is a classic theme in approximation theory. A classic approach to solve those problems for a large class of convex functions $f(x)$ over the domain $\Delta_n := \{(x_1, \ldots x_n) \ : \ x_i \geq 0, \sum_i x_i = 1\}$ is the Frank-Wolfe procedure that produces a solution that is $O(1/T)$ away from the optimum, after T iterations. This method can be used to obtain a trade-off between the approximation of the objective and the sparsity of the solution obtained. [17, 18, 35] are three excellent sources for this line of work. For ℓ_2 sparse regression, this implies that for *any* A, we can obtain a solution sparsity roughly $O(1/\epsilon^2)$, that has a loss ϵ away from the optimum (Maurey's lemma, see also [7]).

However, such a result is impossible with ℓ_1 error. It is easy to see that if A is the identity, and $b = (\frac{1}{n}, \frac{1}{n}, \ldots, \frac{1}{n})$, then the minimum over $x \in \Delta_n$ of $f(x) = \|Ax - b\|_1$ is zero, while for any k-sparse x, the error is $\geq 1 - \frac{k}{n}$. Thus, we ask the question: can we perform such an approximation, when the goal is simply to *compete* with the best k-sparse solution? Our contribution in Theorem 3 is to show that this is possible via a Frank-Wolfe type update using a novel potential function.

A setting very similar to ours was considered in the work [8], where it is shown that if there is a k-sparse vector x^* such that $Ax^* = b$ *exactly*, then an algorithm based on an exponential potential function finds an $O(k/\epsilon^3)$-sparse vector y such that $\|Ay - b\|_1 \leq \epsilon$.[1] The paper [8] uses sparse regression for learning low-dimensional Gaussian mixtures (i.e., express the p.d.f. of the mixture —obtained empirically— as a sparse convex combination of the p.d.f.'s of Gaussians). Our ability to handle error implies that our algorithm can learn mixtures even in the presence of noisy points.

Further related work. Problem 1 above is closely related to many well-studied questions. Blum et al. [9] consider the problem of finding a small subset Q of a given point set P, such that the convex hulls of P and Q are "close". This is equivalent to approximately representing the points in $P \setminus Q$ using points in Q – a goal similar to ours. However, they consider ℓ_2 error, and also require a good approximation for *every* point in P. Another closely related question is that of finding a non-negative matrix factorization (NMF), under the "anchor word" assumption [4, 3]. In NMF, we are given a non-negative matrix M, and the goal is to write $M = XY$, where X, Y are non-negative matrices. The anchor word assumption states that X can be chosen to be a subset of the columns of M, which reduces the problem to non-negative CSS. Our methods can thus be directly applied; however in [3] and related works, the measures of error are different from ours. Also in contrast to our result, much of the work in this area focuses on finding precisely k columns.

Notations. We review some of the notation we will use throughout the paper. For a matrix A, A_i refers to its i'th column, and $A^{(i)}$ refers to its ith row. By $\|A\|_1$, we refer to the sum of the absolute values of the entries in A. Also, $\text{nnz}(A)$ refers to the number of non-zero entries in the matrix. We will refer to Δ_n the "probability simplex" in n dimensions, namely $\{(x_1, \ldots, x_n) \ : \ x_i \geq 0, \sum_i x_i = 1\}$.

[1] Their *proof* can handle a small amount of noise, albeit under the restriction that in *every coordinate*, Ax^* and b are within a $(1 \pm \epsilon)$ factor. Note that is a lot more restrictive than $\|Ax^* - b\|_1$ being small.

1.1 Our results

Let us start by stating our result for sparse regression. The non-negative sparse regression problem (stated above) has as input a matrix A and a target vector b (both non-negative).

▶ **Theorem 3.** *Suppose there exists a non-negative k-sparse vector x^* such that $\|Ax^* - b\|_1 \leq \epsilon \|b\|_1$. Then, there is an efficient algorithm that, for any $\delta > 0$, outputs a y of sparsity $O(k \log(1/\delta)/\delta^2)$, with the guarantee that $\|Ay - b\|_1 \leq \left(4\sqrt{2(\epsilon + 2\delta)}\right) \|b\|_1$.*

The running time of the algorithm is $O(k \log(1/\delta)/\delta^2 \cdot \text{nnz}(A))$. Note also that if we set $\delta = \epsilon$, we obtain an error roughly $O(\sqrt{\epsilon})$ factor of the optimum. It is an interesting open problem to understand if this ϵ versus $\sqrt{\epsilon}$ guarantee is *necessary*. In our analysis, it arises due to a move from KL divergence to ℓ_1 error (via Pinsker's inequality). Indeed, our algorithm produces a much better approximation in KL divergence, as we will see. We then use the theorem above to show our main result, on the non-negative CSS problem.

▶ **Theorem 4.** *Let A, B be non-negative matrices, and suppose there exists a subset S of k columns of A, such that $\min_{X \geq 0} \|B - A_S X\|_1 \leq \epsilon \|B\|_1$. Then for any $\delta > 0$, there is an efficient algorithm that finds a set S' of $O(k \log(1/\delta)/\delta^2)$ columns of A that give an approximation error $O(\sqrt{\delta + \epsilon}) \|B\|_1$.*

Theorems 3 and 4 are proved in sections 2 and 3 respectively. Given the conceptual simplicity of the algorithms, we also implement them effectively, and show some preliminary experimental results in Appendix A.

1.2 Interpreting error bounds and comparisons to prior work

We now focus on the low-rank approximation result (Theorem 4) and discuss the tradeoff between error and the number of columns output. We then compare our result with prior work on ℓ_1 low rank approximation.

First, note that our error bound is *additive*, in a sense. Specifically, the approximation factor can be arbitrarily bad if ϵ is sufficiently small. Indeed, if $\epsilon = o(1)/k$, then prior work gives better approximations. While it is common in theory to assume that low-rank and k-column approximations have error that is *tiny* compared to the norm of the matrix, in practice it is quite common to have situations in which only (say) 90% of the "mass" can be "explained" via a low rank approximation, while the rest is noise. These are the settings in which our methods do significantly better (indeed, none of the earlier results we are aware of give any non-trivial guarantees in such settings). In the case of ℓ_2 error (when we can actually compute the error efficiently), we often observe a drop in the singular *values* (i.e., $\sigma_1, \ldots, \sigma_k$ are larger than the rest) in practice, but the total Frobenius mass on the *tail* is still non-trivial.

Second, we have a tradeoff between the number of columns output and the error we can obtain. We do not know if such a dependence is optimal, and this is an interesting open question. However, in [8], it was shown that even in the zero error case, a trade-off of this nature is essential, assuming that a random planted version of the set cover problem does not have polynomial time algorithms (which seems consistent with current algorithmic techniques).

Next, we compare with the previous work on ℓ_1 low rank approximation. Recently, Song et al. [36] and Chierichetti et al. [15] have made significant progress on the problem on both the algorithm and the hardness fronts. First, our methods are quite different in many respects, even when we restrict to non-negative matrices and we compare the error

of the algorithm with the best k-column approximation, as opposed to the best rank-k approximation. Second, our analysis also applies to the *generalized CSS* problem, where we have two matrices, and we approximate one using the columns of the other. Third, as mentioned above, our algorithm has weaker guarantees than the prior work when the matrix B has a *very low error* ($\ll (1/k)\|B\|_1$) ℓ_1 approximation.

2 Sparse recovery under noise

The aim of this section is to outline the proof of Theorem 3. We start with some simple observations about re-scaling. The first is that we may assume that $\|b\|_1 = 1$ without loss of generality, because otherwise, we can run the entire procedure with $b/\|b\|_1$, and re-scale the coefficients of the obtained y by a factor $\|b\|_1$. The second observation is that we may assume that all the columns of A (which we may assume to be non-zero, as zero columns can be ignored) have unit ℓ_1 norm, w.l.o.g. This is again simple: if not, we can solve the problem with a matrix whose columns are $A_i/\|A_i\|_1$, and then divide the obtained x_i by the corresponding $\|A_i\|_1$ to obtain a solution with the original matrix.

Thus, by way of simplifying notation, assume that the columns of A are denoted by the set V of vectors on the "probability simplex" Δ_n in $\mathbb{R}^n$, and we denote by $p \in \Delta_n$ be the target vector. Suppose there exist $v_1, \ldots, v_k \in V$, and non-negative α_i, such that $\|p - \sum_i \alpha_i v_i\|_1 \leq \epsilon$.

Our goal is to design an iterative algorithm that maintains a vector $q \in \Delta_n$ (the *current approximation* to p), and adds one vector from $v \in V$ to q (with appropriate step size) in each iteration so as to improve a potential. The most natural potential (which works for ℓ_p norms for all $p > 1$), is simply the ℓ_1 distance $\|p - q\|_1$. Unfortunately, for this potential, depending on the current q, there may not exist "local improvements". This can be observed in a simple example:

$$v_1 = (1,0,1,0),\ v_2 = (1,0,0,1),\ v_3 = (0,1,1,0),\ v_4 = (0,1,0,1).$$

Let $p = (\frac{1}{2}, \frac{1}{2}, \frac{1}{2}, \frac{1}{2})$. Clearly, it is in the convex hull of the vectors. Now, consider $q = (1,1,0,0)$. It can easily be verified that there is no single vector that can be added in order to improve $\|p - q\|$. (Indeed, such examples are well-known [18].)

Another natural potential is the relative entropy $D_{\mathrm{KL}}(p \parallel q)$. The problem with this is that it can be extremely sensitive to changes in q when q is close to the *boundary* of the simplex. While we may hope to control the distance to the boundary via a "warm start", it turns out to be tricky to implement.

Instead, we maintain a potential that automatically controls the distance to the boundary, while at the same time, allows us to reason about proximity to the optimum at the end:

$$\Phi(q) := D_{\mathrm{KL}}\left(p \parallel \frac{p+q}{2}\right) = \sum_i p_i \log\left(\frac{2p_i}{p_i + q_i}\right). \tag{1}$$

The potential is, roughly speaking, "one part" of the Jensen-Shannon divergence. The updates we consider are analogous to Frank-Wolfe iterations. In particular, in every round of the algorithm we greedily select the column that minimizes the potential and using it we recompute the vector q as shown in Algorithm 1.

To prove our main results we analyze the drop of the potential in every round of our algorithm. Interestingly we can show that the potential decrease geometrically during the execution of the algorithm. More formally, we show the following lemma.

Algorithm 1 Warm-KL

Initialize $q^{(0)} = v$ for an arbitrary $v \in V$, $S = \emptyset$, $\eta = \delta^2/2k$, and $T = \lceil \log\left(\frac{1}{4\sqrt{\epsilon+2\delta}}\right) / \log\left(\frac{1}{1-\frac{\eta}{2}}\right) \rceil$
for $t = 1 \dots T$ **do**
  Find the column $u \in V$ that minimizes $\Phi((1-\eta)q^{(t-1)} + \eta u)$, and add it to S.
  Set $q^{(t)} \leftarrow (1-\eta)q^{(t-1)} + \eta u$.
end for

▶ **Lemma 5** (Potential drop). *Consider the execution of the algorithm, and suppose* $\Phi(q^{(t-1)}) \geq 4(\epsilon + 2\delta)$. *Then we have*

$$\Phi(q^{(t)}) \leq \left(1 - \frac{\eta}{2}\right) \Phi(q^{(t-1)}).$$

We first show how to prove Theorem 3 using Lemma 5, then in the next section we focus on the proof of the lemma.

Proof of Theorem 3. Note that $\Phi(q^{(t)}) = \sum_i p_i \log\left(\frac{2p_i}{p_i+q_i}\right) \leq \sum_i p_i \ln 2 \leq 1$. So after $T = \lceil \log\left(\frac{1}{4\sqrt{\epsilon+2\delta}}\right) / \log\left(\frac{1}{1-\frac{\eta}{2}}\right) \rceil$ steps, we have $\Phi(q^{(T)}) \leq 4(\epsilon + 2\delta)$. Now, using Pinsker's inequality, we have that

$$\left\| p - \frac{p + q^{(T)}}{2} \right\|_1 \leq \sqrt{2\Phi(q^{(T)})} \leq 2\sqrt{2(\epsilon + 2\delta)}.$$

This then implies that $\left\|p - q^{(T)}\right\|_1 \leq 4\sqrt{2(\epsilon + 2\delta)}$, completing the proof of the theorem. ◀

2.1 Analyzing the potential drop

As shown in the previous subsection the key is to analyze the drop in potential. Let us fix some t, and for convenience, write $q = q^{(t-1)}$, and $q' = q^{(t)}$. We have $q' = (1-\eta)q + \eta u$, for some $u \in V$. Then, we have

$$\Phi(q) - \Phi(q') = \sum_i p_i \log\left(\frac{p_i + q'_i}{p_i + q_i}\right) = \sum_i p_i \log\left(1 + \eta \cdot \frac{u_i - q_i}{p_i + q_i}\right). \tag{2}$$

Note that we wish to lower bound this potential drop. In other words, we need to prove that there exists a $u \in V$ such that the difference above is "large". Since we know that there is a linear combination $\sum_i \alpha_i v_i$ that is ϵ-close to p (in ℓ_1), the natural goal is to prove that one of the v_i achieves the necessary potential drop, by an averaging argument. This is typically done by approximating the change in Φ by a linear function of the u_i's. If $\eta \cdot \frac{u_i - q_i}{p_i + q_i} < 1$, this can be done using $\log(1+x) \approx x$. But unfortunately in our case, the term $\frac{u_i - q_i}{p_i + q_i}$ can be arbitrarily large, making such an approximation impossible.

To deal with this, we take advantage of additional structure. The first observation is the following.

▶ **Lemma 6.** *Let* v_j, $1 \leq j \leq k$, *be vectors in* Δ_n *such that* $\left\|p - \sum_j \alpha_j v_j\right\|_1 \leq \epsilon$. *Then for all* $\delta > 0$, *there exists a subset* S^* *of* $[k]$, *such that* $\left\|p - \sum_{j \in S^*} \alpha_j v_j\right\|_1 \leq \delta + \epsilon$, *and additionally,* $\alpha_j \geq \delta/k$ *for all* $j \in S^*$. *(I.e., the coefficients used are all "non-negligible".)*

▶ Remark. Even though the lemma is straightforward, it is the only place in the proof we use the "promise" that there exists a k-sparse approximation to p.

Proof. The proof is simple: we consider $\sum_j \alpha_j v_j$, and remove all the terms whose coefficients are $< \delta/k$. As there are at most k terms, the total ℓ_1 norm of the removed terms is $\leq \delta$. This implies that considering only the terms that remain has an error at most $\epsilon + \delta$. ◀

The lemma shows that we can obtain a good approximation only using large coefficients. As a consequence, we now show that we can restrict our attention to a truncated version of the vectors V, which enables our analysis. Formally, define "truncated" vectors w_j by setting

$$w_{ji} = \min\left\{v_{ji}, \frac{kp_i}{\delta}\right\}.$$

I.e., the w_j are the vectors v_j, truncated so that no entry is more than k/δ times the corresponding entry in p. We start with a simple observation.

▶ **Lemma 7.** *For the vectors v_j, w_j as above, we have*

$$\left\|p - \sum_{j\in S^*} \alpha_j w_j\right\|_1 \leq \left\|p - \sum_{j\in S^*} \alpha_j v_j\right\|_1.$$

Proof. Let us look at the ith coordinate of the vectors on the LHS and RHS. The only way we can have $v_{ji} \neq w_{ji}$ (for some j) is when $v_{ji} > kp_i/\delta$, in which case $\alpha_j v_{ji} > p_i$. Thus in this coordinate, $\sum_{j\in S^*} \alpha_j v_j$ has a value larger than p_i. Now, by moving to $\sum_{j\in S^*} \alpha_j w_j$, we decrease the value, but remain larger or equal to p_i. Thus, the norm of the difference only improves. ◀

Let us now go back to the drop in potential we wished to analyze (eq. (2)). Our aim is to prove that setting $u = v_j$ for some $j \in S^*$ in the algorithm leads to a significant drop in the potential. We instead prove that setting $u = w_j$ (as opposed to v_j) leads to a significant drop in the potential. Then, we can use the fact that the RHS of (2) is monotone in u (noting that $v_{ji} \geq w_{ji}$) to complete the proof.

Let us analyze the potential drop when $u = w_j$ for some $j \in S^*$. Let $\eta = \delta^2/2k$. Write $\gamma_i := \frac{u_i - q_i}{p_i + q_i}$. Then, we have

$$\gamma_i = \frac{p_i + u_i}{p_i + q_i} - 1 \in [-1, k/\delta], \quad \text{as } w_{ji}/p_i \leq k/\delta \text{ for } j \in S^*.$$

Now, using the value of η, we have that $\eta\gamma_i \in (-\eta, \delta/2)$. This allows us to use a linear approximation for the logarithmic term in Φ. Once we move to a linear approximation, we can use an averaging argument to show that there exists a choice of u that improves the potential substantially.

More formally, let I_+ denote the set of indices with $u_i \geq q_i$ (i.e., $\gamma_i \geq 0$), and I_- denote the other indices. Then, using that, for $x > -1$, $\log(1+x) \geq x(1-x)$ we have that for any $i \in I_+$, we have $\log(1 + \eta\gamma_i) \geq \eta\gamma_i(1 - \frac{\delta}{2})$. Further, for $i \in I_-$, we have $\log(1 + \eta\gamma_i) \geq \eta\gamma_i(1 + \eta)$. Thus, in both the cases, we have that

$$\log(1 + \eta\gamma_i) \geq \eta\gamma_i - \frac{\delta\eta|\gamma_i|}{2}. \tag{3}$$

Now before showing how to use to the inequality to conclude the proof, we use an averaging argument to prove the existence of a good column j.

▶ **Lemma 8.** *There exists an index $j \in S^*$ such that setting $u = w_j$ in the algorithm gives*

$$\sum_i p_i \gamma_i \geq D_{KL}\left(p \parallel \frac{p+q}{2}\right) - (2\epsilon + \delta).$$

Proof. We start by recalling (via Lemmas 7 and 6), that

$$\left\| p - \sum_{j \in S^*} \alpha_j w_j \right\|_1 \leq \left\| p - \sum_{j \in S^*} \alpha_j v_j \right\|_1 \leq (\epsilon + \delta).$$

For convenience, let us write $r = \sum_{j \in S^*} \alpha_j w_j$, and so $\|p - r\|_1 \leq \epsilon + \delta$. Now, we wish to analyze the behavior of γ_i when we set $u = w_j$ for different j.

We start by defining $\tau_i^{(j)}$ as

$$\tau_i^{(j)} := \frac{w_{ji} - q_i}{p_i + q_i},$$

i.e., the value of γ_i obtained by setting $u = w_j$. For convenience, write $Z = \sum_{j \in S^*} \alpha_j$. Now by linearity, we have

$$\sum_{j \in S^*} \alpha_j \left(\sum_i p_i \tau_i^{(j)} \right) = \sum_i p_i \frac{(\sum_{j \in S^*} \alpha_j w_{ji}) - Z q_i}{p_i + q_i} = \sum_i p_i \frac{r_i - Z q_i}{p_i + q_i}.$$

Thus, by averaging (specifically, the inequality that if $\sum_j \alpha_j X_j \geq Y$ for $\alpha_j \geq 0$, then there exists a j such that $X_j \geq Y/(\sum_j \alpha_j)$), we have that there exists a $j \in S^*$ such that

$$\sum_i p_i \tau_i^{(j)} \geq \frac{1}{Z} \cdot \sum_i p_i \frac{r_i - Z q_i}{p_i + q_i} = \sum_i p_i \frac{r_i - q_i}{p_i + q_i} + \left(\frac{1}{Z} - 1\right) \sum_i p_i \frac{r_i}{p_i + q_i}. \tag{4}$$

The first term on the RHS can now be lower bounded as:

$$\sum_i p_i \frac{r_i - q_i}{p_i + q_i} = \sum_i p_i \left(\frac{2p_i}{p_i + q_i} - 1 - \frac{p_i - r_i}{p_i + q_i} \right) \geq \sum_i p_i \left(\frac{2p_i}{p_i + q_i} - 1 \right) - \sum_i |p_i - r_i|,$$

where we use the fact that $|p_i/(p_i + q_i)| \leq 1$. Thus, appealing to the inequality $(x - 1) \geq \log x$, if we write $D := D_{\mathrm{KL}}\left(p \parallel \frac{p+q}{2}\right)$, then we have

$$\sum_i p_i \frac{r_i - q_i}{p_i + q_i} \geq D - \epsilon - \delta, \quad \text{since } \|p - r\|_1 \leq \epsilon + \delta.$$

To conclude the proof, let us consider the second term in the RHS of (4). If $Z \leq 1$, the term is non-negative, and there is nothing to show. We next argue that $Z \leq 1 + \epsilon$, by showing that the sum of α_j over *all* $j \in S$ can be bounded by $1 + \epsilon$. In fact, note that $\left\| \sum_{j \in S} \alpha_j v_j \right\|_1 - 1 \leq \left\| p - \sum_{j \in S} \alpha_j v_j \right\|_1 \leq \epsilon$ (triangle inequality). Furthermore for $v_j \in \Delta_n$, we have $\left\| \sum_j \alpha_j v_j \right\|_1 = \sum_j \alpha_j$. Thus $\sum_{j \in S} \alpha_j \leq 1 + \epsilon$, and thus

$$\left(1 - \frac{1}{Z}\right) \sum_i \frac{p_i r_i}{p_i + q_i} \leq \frac{\epsilon}{1 + \epsilon} \sum_i r_i \leq \epsilon.$$

Plugging this into (4) we can conclude the proof of the lemma. ◀

We are now ready to complete the proof of the main lemma of this section – Lemma 5.

Proof of Lemma 5. Let $D := D_{\mathrm{KL}}\left(p \parallel \frac{p+q}{2}\right)$. We can use Lemma 8 in conjunction with Eq. (3) to obtain that there exists a j such that setting $u = w_j$ gives us

$$\sum_i p_i \log(1+\eta\gamma_i) \geq \sum_i \eta p_i \gamma_i - \frac{\delta\eta}{2}\sum_i p_i|\gamma_i| \geq \eta(D - 2\epsilon - \delta) - \frac{\delta\eta}{2}\sum_i p_i|\gamma_i|.$$

The last term on the RHS can be bounded, noting that

$$\sum_i p_i|\gamma_i| \leq \sum_i p_i \cdot \frac{|u_i - q_i|}{p_i + q_i} \leq \sum_i |u_i - q_i| \leq 2,$$

as u_i, p_i, q_i are all probability distributions, and $\frac{p_i}{p_i+q_i} \leq 1$. This implies that there exists a choice of $u = w_j$ (and thus setting $u = v_j$ also works, as discussed earlier), such that the potential drop is at least $\eta(D - 2\epsilon - \delta) - \eta\delta$. If $D \geq 4(\epsilon + \delta)$, this is at least $\eta D/2$. This completes the proof of the lemma. ◀

3 Low rank approximation

We now come to our main result – Theorem 4. Let the dimensions of B be $n \times m$ (thus A also has n rows for the problem to be well-defined). The main difference between this setting and the one in Section 2 is that we have a collection of m columns, and we wish to find a subset S that can "simultaneously" approximate all of them. Another technical difference is that the columns of B can all have different lengths, thus we can only re-scale all of them by the same amount.

Let us start with some simple assumptions we can make w.l.o.g. First, we may assume that $\|B\|_1 = 1$, as we can scale the entire matrix by $1/\|B\|_1$, solve the problem, and then multiply the obtained X (entry-wise) by $\|B\|_1$. Second, we may assume that every column of A has unit ℓ_1 norm, as otherwise, we can solve the problem using a matrix with columns $A_i/\|A_i\|_1$, and then re-scale the ith *row* of the obtained X by $1/\|A_i\|$ to find a solution to the original problem.

Under these assumptions (i.e., $\|B\|_1 = 1$ and $A_i \in \Delta_n$), we now show how Section 2 gives a "framework" that can be used here. The main idea is as follows.

Outline of the approach

Let us flatten the matrix B into an nm dimensional vector p (thus the (i,j)th entry of B now appears in the $[(j-1)n + i]$th position of p). Now, the CSS problem can be re-stated as expressing p as a linear combination of vectors of the form $A_i \otimes y_i$, for some non-negative vectors y_i (these will form the *rows* of X in the problem definition). Thus, let V denote the set of all vectors of the form

$$\{A_i \otimes z \ : \ A_i \text{ is a column of } A,\ z \in \Delta_m\}.$$

The goal is to use the framework of Section 2 to find a small subset of V. Unfortunately, the set V above is infinite! So we cannot apply Algorithm 1 directly. However, note that as long as we can find a $u \in V$ that satisfies the potential drop condition of Lemma 5, the analysis still applies. In the remainder of the section, we show how to design an efficient "oracle" to find such a u, thus proving the theorem.

Before we begin, let us observe that our normalizations indeed reduce our matrix problem to the problem analyzed in the previous section. For any $u \in V$, we have $\|u\|_1 = \|A_i\|_1 \|z\|_1 = 1$. We have also normalized so that $\|p\|_1 = 1$ (recall p is the flattened form of B). Now, suppose B has a good ℓ_1 approximation using a submatrix A_S, i.e., suppose $\|B - A_S X\| \leq \epsilon$,

for some matrix X. Then, denoting by $X^{(j)}$ the jth row of X, we can re-write the above as $\left\|B - \sum_{j\in S} A_j X^{(j)}\right\|_1 \leq \epsilon$. Now setting $y_j = X^{(j)} / \left\|X^{(j)}\right\|_1$ (so $y_j \in \Delta_m$), and $\alpha_j = \left\|X^{(j)}\right\|_1$, we can re-write the above as

$$\left\| p - \sum_{j\in S} \alpha_j (A_j \otimes y_j) \right\|_1 \leq \epsilon, \tag{5}$$

which is precisely the form we need. Let us thus see how to efficiently find a $u \in V$ that reduces the potential significantly.

3.1 Oracle for each iteration

Consider the t-th iteration, and let $q^{(t-1)}$ be the previous iterate (which we denote by q, for convenience). Our goal is to find a u such that $\Phi((1-\eta)q + \eta u)$ is small. The technical difficulty here is the logarithmic term in the definition of $\Phi()$.

As this was also the challenge in Section 2, we begin by recalling one key aspect of the analysis: the definition of the truncated vectors w. Following the notation earlier, and from Eq. (5), define $v_j = A_j \otimes y_j$, and let w_j be defined using $w_{ji} := \min\{v_{ji}, \frac{kp_i}{\delta}\}$ (i now ranges from 1 to nm). The main component of the argument was the existence of an index j such that $\sum_i p_i \frac{w_{ji}-q_i}{p_i+q_i}$ is large.

Thus, if we can find a $u \in V$ such that for its truncation h (defined by $h_i := \min\{u_i, \frac{kp_i}{\delta}\}$), the value $\sum_i p_i \frac{h_i-q_i}{p_i+q_i}$ is large, we would be done. When V is finite, we simply did this by iterating over all $u \in V$, constructing the h, and computing the values. In the current setting, we have infinitely many V. Fortunately we can handle this complication by analyzing the columns separately. First partition V into V_j, one for each column A_j. I.e., $V_j := \{A_j \otimes z \ : \ z \in \Delta_n\}$. We then design an efficient method to maximize the quantity described above over a single V_j. Since there are only finitely many columns in A, we can finally just take the maximum.

▶ **Lemma 9.** *Let us fix some j. There is an efficient algorithm to find $u \in V_j$ (defined above) to maximize $\sum_i p_i \frac{h_i-q_i}{p_i+q_i}$, where $h_i = \min\{u_i, \frac{kp_i}{\delta}\}$.*

Proof. Observe that as every u is of the form $A_j \otimes z$, we need to find a $z \in \Delta_m$ that maximizes the objective value. The key observation is that this can be written as a linear program! To define the LP, it helps to view i as being defined by (i_1, i_2), where $i = n(i_2 - 1) + i_1$, where $i_1 \in [n]$ and $i_2 \in [m]$. The variables are $h \in \mathbb{R}^{nm}$, $z \in \mathbb{R}^m$; p, q are given (p is the target and q is the current iterate). The LP is as follows:

$$\text{maximize} \sum_{i_1,i_2} \frac{p_{(i_1,i_2)}}{p_{(i_1,i_2)} + q_{(i_1,i_2)}} h_{(i_1,i_2)}, \quad \text{subject to} \tag{6}$$

$$\forall\ i_1 \in [n], i_2 \in [m],\ \ 0 \leq h_{(i_1,i_2)} \leq A_{j,i_1} \cdot z_{i_2} \tag{7}$$

$$\forall\ i_1 \in [n], i_2 \in [m],\ \ h_{(i_1,i_2)} \leq \frac{kp_{(i_1,i_2)}}{\delta} \tag{8}$$

$$\forall\ i_2 \in [m],\ \ z_{i_2} \geq 0 \tag{9}$$

$$\sum_{i_2 \in [m]} z_{i_2} = 1. \tag{10}$$

In the above, A_{j,i_1} is simply the i_1'th entry of the vector A_j. The optimum solution will satisfy $h_{(i_1,i_2)} = \min\{A_{j,i_1} \cdot z_{i_2}, \frac{kp_{(i_1,i_2)}}{\delta}\}$, as for any given z, this setting will maximize the objective (which is the goal). From the definition of V_j, it follows that the LP indeed captures the problem of maximizing over V_j. This proves the lemma. ◀

As the LP can be solved in polynomial time, we can follow this procedure for every column j of A. This completes the proof of Theorem 4 if we only require a polynomial time algorithm. As LP solvers can be inefficient, in the next subsection we developed a fast "greedy" solution to our specific LP.

4 Solving the linear program efficiently

While linear programs can be solved in polynomial time, they can often be a bottleneck in learning algorithms in practice. We thus give a simple "greedy" algorithm to solve the LP in (6)-(10)[2].

▶ **Theorem 10.** *Consider the LP* (6)-(10)*, written for a given column A_j of A. There is a greedy algorithm that find the optimal z in time $O(mn \log n)$. (For sparse matrices this can be further reduced to $O(\text{nnz}(A) \log n)$).*

Proof. Let us rewrite the objective function, splitting it according to i_2:

$$\sum_{i_2 \in [m]} \sum_{i_1 \in [n]} \frac{p_{(i_1,i_2)}}{p_{(i_1,i_2)} + q_{(i_1,i_2)}} h_{(i_1,i_2)}.$$

Let us now focus on one index i_2, and consider increasing z_{i_2} from 0 to 1. The value of $h_{(i_1,i_2)}$ is equal to $A_{j,i_1} \cdot z_{i_2}$ until z_{i_2} hits $\min\{1, kp_{(i_1,i_2)}/\delta A_{j,i_1}\}$, and then remains constant. (If $A_{j,i_1} = 0$, then it stays 0 throughout.) This happens for each i_1, and thus the inner summation, as a function of z_{i_2}, is piecewise linear and non-decreasing. Let us denote this function by $f_{i_2}(x)$.

Finding the optimal z can now be cast as the following problem: we have monotone, concave, piecewise-linear functions $f_1, \ldots, f_m$ defined on $[0, 1]$, and we wish to find $z_j \geq 0$, summing to 1, such that $\sum_j f_j(z_j)$ is maximized.

This can be greedily done as follows: suppose we have a sorted list of the "break-points" for each f_j (a break-point is the value of x at which a piece-wise linear function changes slope). Note that for every j, the number of break-points (in the interval $[0, 1]$) is no more than n. Now, we start with $z_j = 0$ for all j. In every iteration, we decide to increment one z_j. The choice will depend on the j that has the largest slope to the right of z_j. Once we pick a j, we increment the z_j until its next break-point (or until the sum of z_j's becomes 1).

The correctness of this procedure follows from the fact that we always increment the z_j with the largest slope so we maximize the value of $\sum_j f_j(z_j)$ (and because the functions are concave, we can never encounter a higher slope later). As for the run time, we note that the total number of break points is mn, and in the worst case, we could encounter each one. At each step, as the slopes are monotone, the choice of the j can be found using a priority queue, and this leads to a run time of $O(mn \log n)$. If the A is sparse, then the number of "slopes" we need to consider is at most the number of non-zero entries, which gives the desired bound. ◀

5 Conclusion

We have presented new algorithmic results for two fundamental problems, sparse regression and column subset selection, under ℓ_1 error. The key assumption necessary for us is the non-negativity of the associated matrices and the decomposition. Under this assumption,

[2] We note that our argument is similar to the fractional knapsack argument.

our algorithms provably achieve approximation guarantees with respect to the corresponding optimal solutions. Our sparse regression analysis gives a simple framework for obtaining ℓ_1 error guarantees. We have used it for our column selection result, but it may be applicable to other contexts as well, such as non-negative matrix factorization (for which our result applies under the anchor word assumption), matrix and tensor variants of sparse regression/recovery, and also in obtaining recovery algorithms for broader classes of mixture models under noise. We leave these as interesting avenues for future work.

References

1 Jason Altschuler, Aditya Bhaskara, Gang Fu, Vahab S. Mirrokni, Afshin Rostamizadeh, and Morteza Zadimoghaddam. Greedy column subset selection: New bounds and distributed algorithms. In Maria-Florina Balcan and Kilian Q. Weinberger, editors, *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*, volume 48 of *JMLR Workshop and Conference Proceedings*, pages 2539–2548. JMLR.org, 2016. URL: `http://jmlr.org/proceedings/papers/v48/altschuler16.html`.

2 Edoardo Amaldi and Viggo Kann. On the approximability of minimizing nonzero variables or unsatisfied relations in linear systems. *Theoretical Computer Science*, 209(1Ð2):237–260, 1998.

3 Sanjeev Arora, Rong Ge, Yonatan Halpern, David M. Mimno, Ankur Moitra, David Sontag, Yichen Wu, and Michael Zhu. A practical algorithm for topic modeling with provable guarantees. In *Proceedings of the 30th International Conference on Machine Learning, ICML 2013, Atlanta, GA, USA, 16-21 June 2013*, volume 28 of *JMLR Workshop and Conference Proceedings*, pages 280–288. JMLR.org, 2013. URL: `http://jmlr.org/proceedings/papers/v28/arora13.html`.

4 Sanjeev Arora, Rong Ge, Ravindran Kannan, and Ankur Moitra. Computing a nonnegative matrix factorization - provably. 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 145–162. ACM, 2012. `doi:10.1145/2213977.2213994`.

5 Arturs Backurs, Piotr Indyk, Ilya P. Razenshteyn, and David P. Woodruff. Nearly-optimal bounds for sparse recovery in generic norms, with applications to k-median sketching. 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 318–337. SIAM, 2016. `doi:10.1137/1.9781611974331.ch24`.

6 Afonso S. Bandeira, Edgar Dobriban, Dustin G. Mixon, and William F. Sawin. Certifying the restricted isometry property is hard, 2012. `arXiv:arXiv:1204.1580`.

7 Siddharth Barman. Approximating nash equilibria and dense bipartite subgraphs via an approximate version of caratheodory's theorem. 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 361–369. ACM, 2015. `doi:10.1145/2746539.2746566`.

8 Aditya Bhaskara, Ananda Theertha Suresh, and Morteza Zadimoghaddam. Sparse solutions to nonnegative linear systems and applications. In Guy Lebanon and S. V. N. Vishwanathan, editors, *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Statistics, AISTATS 2015, San Diego, California, USA, May 9-12, 2015*, volume 38 of *JMLR Workshop and Conference Proceedings*. JMLR.org, 2015. URL: `http://jmlr.org/proceedings/papers/v38/bhaskara15.html`.

9 Avrim Blum, Sariel Har-Peled, and Benjamin Raichel. Sparse approximation via generating point sets. 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 548–557. SIAM, 2016. doi:10.1137/1.9781611974331.ch40.

10 Christos Boutsidis and David P. Woodruff. Optimal CUR matrix decompositions. *SIAM J. Comput.*, 46(2):543–589, 2017. doi:10.1137/140977898.

11 J.P. Brooks, J.H. Dulá, and E.L. Boone. A pure ℓ_1-norm principal component analysis. *Computational Statistics & Data Analysis*, 61:83–98, 2013.

12 Joe M Butler, D Timothy Bishop, and Jennifer H Barrett. Strategies for selecting subsets of single-nucleotide polymorphisms to genotype in association studies. *BMC genetics*, 6(1):S72, 2005.

13 Emmanuel J. Candès, Xiaodong Li, Yi Ma, and John Wright. Robust principal component analysis? *JACM*, 58(3):11:1–11:37, 2011.

14 Emmanuel J Candes, Justin K Romberg, and Terence Tao. Stable signal recovery from incomplete and inaccurate measurements. *Communications on pure and applied mathematics*, 59(8):1207–1223, 2006.

15 Flavio Chierichetti, Sreenivas Gollapudi, Ravi Kumar, Silvio Lattanzi, Rina Panigrahy, and David P. Woodruff. Algorithms for ℓ_p low-rank approximation. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pages 806–814. PMLR, 2017. URL: http://proceedings.mlr.press/v70/chierichetti17a.html.

16 Hugh A Chipman and Hong Gu. Interpretable dimension reduction. *Journal of applied statistics*, 32(9):969–987, 2005.

17 Kenneth L. Clarkson. Coresets, sparse greedy approximation, and the frank-wolfe algorithm. *ACM Trans. Algorithms*, 6(4):63:1–63:30, 2010. doi:10.1145/1824777.1824783.

18 M. J. Donahue, C. Darken, L. Gurvits, and E. Sontag. Rates of convex approximation in non-hilbert spaces. *Constructive Approximation*, 13(2):187–220, 1997. doi:10.1007/BF02678464.

19 David L Donoho and Michael Elad. Optimally sparse representation in general (nonorthogonal) dictionaries via ?1 minimization. *Proceedings of the National Academy of Sciences*, 100(5):2197–2202, 2003.

20 A. Eriksson and A. van den Hengel. Efficient computation of robust low-rank matrix approximations using the L_1 norm. *PAMI*, 34(9):1681–1690, 2012.

21 Dan Feldman, Mikhail Volkov, and Daniela Rus. Dimensionality reduction of massive sparse datasets using coresets. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 2766–2774. Curran Associates, Inc., 2016. URL: http://papers.nips.cc/paper/6596-dimensionality-reduction-of-massive-sparse-datasets-using-coresets.pdf.

22 Dean P. Foster, Howard J. Karloff, and Justin Thaler. Variable selection is hard. In Peter Grünwald, Elad Hazan, and Satyen Kale, editors, *Proceedings of The 28th Conference on Learning Theory, COLT 2015, Paris, France, July 3-6, 2015*, volume 40 of *JMLR Workshop and Conference Proceedings*, pages 696–709. JMLR.org, 2015. URL: http://jmlr.org/proceedings/papers/v40/Foster15.html.

23 Alan M. Frieze, Ravi Kannan, and Santosh Vempala. Fast monte-carlo algorithms for finding low-rank approximations. *J. ACM*, 51(6):1025–1041, 2004. doi:10.1145/1039488.1039494.

24 Rahul Garg and Rohit Khandekar. Gradient descent with sparsification: an iterative algorithm for sparse recovery with restricted isometry property. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 337–344. ACM, 2009.

25 Venkatesan Guruswami and Ali Kemal Sinop. Optimal column-based low-rank matrix reconstruction. In Yuval Rabani, editor, *Proceedings of the Twenty-Third Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2012, Kyoto, Japan, January 17-19, 2012*, pages 1207–1214. SIAM, 2012. URL: `http://portal.acm.org/citation.cfm?id=2095211&CFID=63838676&CFTOKEN=79617016`.

26 Peter J. Huber. *Robust Statistics*. John Wiley & Sons, New York„ 1981.

27 Qifa Ke and Takeo Kanade. Robust L_1 norm factorization in the presence of outliers and missing data by alternative convex programming. In *CVPR*, pages 739–746, 2005.

28 Pascal Koiran and Anastasios Zouzias. Hidden cliques and the certification of the restricted isometry property. *IEEE Trans. Information Theory*, 60(8):4999–5006, 2014. `doi:10.1109/TIT.2014.2331341`.

29 Robert Krauthgamer, editor. *Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2016, Arlington, VA, USA, January 10-12, 2016*. SIAM, 2016. `doi:10.1137/1.9781611974331`.

30 Anastasios Kyrillidis, Stephen Becker, Volkan Cevher, and Christoph Koch. Sparse projections onto the simplex. In *Proceedings of the 30th International Conference on Machine Learning, ICML 2013, Atlanta, GA, USA, 16-21 June 2013*, volume 28 of *JMLR Workshop and Conference Proceedings*, pages 235–243. JMLR.org, 2013. URL: `http://jmlr.org/proceedings/papers/v28/kyrillidis13.html`.

31 Cewu Lu, Jiaping Shi, and Jiaya Jia. Scalable adaptive robust dictionary learning. *TIP*, 23(2):837–847, 2014.

32 Deyu Meng and Fernando. D. L. Torre. Robust matrix factorization with unknown noise. In *ICCV*, pages 1337–1344, 2013.

33 Bruno A. Olshausen and David J. Field. Sparse coding with an overcomplete basis set: A strategy employed by v1? *Vision Research*, 37(23):3311–3325, 1997. `doi:10.1016/S0042-6989(97)00169-7`.

34 Saurabh Paul, Malik Magdon-Ismail, and Petros Drineas. Column selection via adaptive sampling. In Corinna Cortes, Neil D. Lawrence, Daniel D. Lee, Masashi Sugiyama, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pages 406–414, 2015. URL: `http://papers.nips.cc/paper/6011-column-selection-via-adaptive-sampling`.

35 Shai Shalev-Shwartz, Nathan Srebro, and Tong Zhang. Trading accuracy for sparsity in optimization problems with sparsity constraints. *SIAM J. on Optimization*, 20(6):2807–2832, 2010.

36 Zhao Song, David P. Woodruff, and Pelin Zhong. Low rank approximation with entrywise ℓ_1-norm error. In *STOC*, 2017.

37 Naiyan Wang, Tiansheng Yao, Jingdong Wang, and Dit-Yan Yeung. A probabilistic approach to robust matrix factorization. In *ECCV*, pages 126–139, 2012.

38 Naiyan Wang and Dit-Yan Yeung. Bayesian robust matrix factorization for image and video processing. In *ICCV*, pages 1785–1792, 2013.

39 Tengyao Wang, Quentin Berthet, and Yaniv Plan. Average-case hardness of RIP certification. *CoRR*, abs/1605.09646, 2016. `arXiv:1605.09646`.

40 Kai Wei, Yuzong Liu, Katrin Kirchhoff, Chris Bartels, and Jeff Bilmes. Submodular subset selection for large-scale speech training data. In *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*, pages 3311–3315. IEEE, 2014.

41 L. Xiong, X. Chen, and J. Schneider. Direct robust matrix factorization for anomaly detection. In *ICDM*, pages 844–853, 2011.

42 L. Xu and A. L. Yuille. Robust principal component analysis by self-organizing rules based on statistical physics approach. *IEEE Transactions on Neural Networks*, 6(1):131–143, 1995.

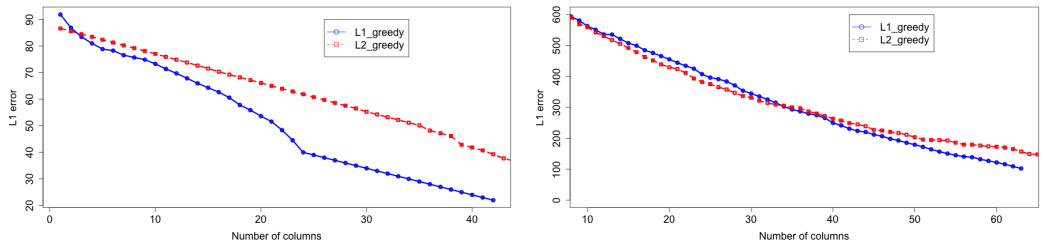
43 Y. Zheng, G. Liu, S. Sugimoto, S. Yan, and M. Okutomi. Practical low-rank matrix approximation under robust L_1-norm. In *CVPR*, pages 1410–1417, 2012.

A Experiments

We now outline some preliminary experimental results that show the effectiveness of our ℓ_1 column subset selection algorithm. In particular we compare our algorithm with the greedy algorithm of [1], which optimizes a potential function similar to ours, but tailored towards ℓ_2 approximation. We compare on synthetic as well as a real-life dataset.

The synthetic dataset is constructed as follows. The matrix A has all its columns on the unit simplex Δ_n, where $n = 150$. The points are divided into two categories – inliers and outliers. The inliers are all in the convex hull of a special set of $k = 15$ points, and there are 115 of them (including the k special points). Along with these, there are $M = 40$ outliers, which are chosen to be sparser than the average point in the special set. (Note that the special points are the "anchor points" in the non-negative factorization connection pointed to in Section 1.) Thus the dataset has a large number of outliers (25%). The figure shows the decay of ℓ_1 error as we pick more columns. Note that even in the first few iterations, our potential allows quickly zeroing in on the right columns.

We then consider a real world matrix obtained from a region economic model, known as the WM1 matrix. The matrix is asymmetric and it is 207 x 277 and contains 2909 real-valued entries. The results on these matrices (showing rank vs the approximation error) for the two algorithms we consider can be found in the figure below. It is interesting to observe that the our ℓ_1 based algorithms has better performances when the error becomes smaller and it converges earlier to a better solution.



(a) (b)

Figure 1 L1-recovery error of the two column selection algorithms on (a) a synthetic dataset, (b) WM1 dataset.

Spectrum Approximation Beyond Fast Matrix Multiplication: Algorithms and Hardness*

Cameron Musco[1], Praneeth Netrapalli[2], Aaron Sidford[3], Shashanka Ubaru[4], and David P. Woodruff[5]

1 **Massachusetts Institute of Technology, Cambridge, MA, USA**
cnmusco@mit.edu
2 **Microsoft Research, Bangalore, India**
praneeth@microsoft.com
3 **Stanford University, Stanford, CA, USA**
sidford@stanford.edu
4 **University of Minnesota, Minneapolis, MN, USA**
ubaru001@umn.edu
5 **Carnegie Mellon University, Pittsburgh, PA, USA**
dwoodruf@cs.cmu.edu

Abstract

Understanding the singular value spectrum of a matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$ is a fundamental task in countless numerical computation and data analysis applications. In matrix multiplication time, it is possible to perform a full SVD of $\mathbf{A}$ and directly compute the singular values $\sigma_1, \ldots, \sigma_n$. However, little is known about algorithms that break this runtime barrier.

Using tools from stochastic trace estimation, polynomial approximation, and fast linear system solvers, we show how to efficiently isolate different ranges of $\mathbf{A}$'s spectrum and approximate the number of singular values in these ranges. We thus effectively compute an *approximate histogram* of the spectrum, which can stand in for the true singular values in many applications.

We use our histogram primitive to give the first algorithms for approximating a wide class of symmetric matrix norms and spectral sums *faster than the best known runtime for matrix multiplication*. For example, we show how to obtain a $(1+\epsilon)$ approximation to the Schatten 1-norm (i.e. the nuclear or trace norm) in just $\tilde{O}((\text{nnz}(\mathbf{A})n^{1/3} + n^2)\epsilon^{-3})$ time for $\mathbf{A}$ with uniform row sparsity or $\tilde{O}(n^{2.18}\epsilon^{-3})$ time for dense matrices. The runtime scales smoothly for general Schatten-p norms, notably becoming $\tilde{O}(p\,\text{nnz}(\mathbf{A})\epsilon^{-3})$ for any real $p \geq 2$.

At the same time, we show that the complexity of spectrum approximation is inherently tied to fast matrix multiplication in the small ϵ regime. We use fine-grained complexity to give conditional lower bounds for spectrum approximation, showing that achieving milder ϵ dependencies in our algorithms would imply triangle detection algorithms for general graphs running in faster than state of the art matrix multiplication time. This further implies, through a reduction of [72], that highly accurate spectrum approximation algorithms running in subcubic time can be used to give subcubic time matrix multiplication. As an application of our bounds, we show that precisely computing all effective resistances in a graph in less than matrix multiplication time is likely difficult, barring a major algorithmic breakthrough.

1998 ACM Subject Classification F.2.1 Numerical Algorithms and Problems

Keywords and phrases spectrum approximation, matrix norm computation, fine-grained complexity, linear algebra

Digital Object Identifier 10.4230/LIPIcs.ITCS.2018.8

* A full version of the paper is available at https://arxiv.org/abs/1704.04163

1 Introduction

Given $\mathbf{A} \in \mathbb{R}^{n\times d}$, a central primitive in numerical computation and data analysis is to compute $\mathbf{A}$'s spectrum: the singular values $\sigma_1(\mathbf{A}) \geq \ldots \geq \sigma_d(\mathbf{A}) \geq 0$. These values can reveal matrix structure and low effective dimensionality, which can be exploited in a wide range of spectral data analysis methods [33, 67]. The singular values are also used as tuning parameters in many numerical algorithms performed on $\mathbf{A}$ [22], and in general, to determine some of the most well-studied matrix functions [29]. For example, for any $f : \mathbb{R}^+ \to \mathbb{R}^+$, we can define the *spectral sum*:

$$\mathcal{S}_f(\mathbf{A}) \stackrel{\text{def}}{=} \sum_{i=1}^{d} f(\sigma_i(\mathbf{A})).$$

Spectral sums often serve as snapshots of $\mathbf{A}$'s spectrum and are important in many applications. They encompass, for example, the log-determinant, the trace inverse, the Schatten-p norms, including the nuclear norm, and general Orlicz norms (see Section 1.2 for details).

While the problem of computing a few of the largest or smallest singular values of $\mathbf{A}$ has been exhaustively studied [56, 60], much less is known about algorithms that approximate the full spectrum, and in particular, allow for the computation of summary statistics such as spectral sums. In n^ω time, it is possible to perform a full SVD and compute the singular values exactly.[1] Here, and throughout, $\omega \approx 2.3729$ denotes the *current* best exponent of fast matrix multiplication [70]. However, even if one simply desires, for example, a constant factor approximation to the nuclear norm $\|\mathbf{A}\|_1$, no $o(n^\omega)$ time algorithm is known. We study the question of spectrum approximation, asking whether obtaining an accurate picture of $\mathbf{A}$'s spectrum is truly as hard as matrix multiplication, or if it is possible to break this barrier. We focus on spectral sums as a motivating application.

1.1 Our Contributions

1.1.1 Upper Bounds

On the upper bound side, we show that significant information about $\mathbf{A}$'s spectrum can be determined in $o(n^\omega)$ time, for the current value of ω. We show how to compute a histogram of the spectrum, which gives approximate counts of the number of squared singular values in the ranges $[(1-\alpha)^t\sigma_1^2(\mathbf{A}), (1-\alpha)^{t-1}\sigma_1^2(\mathbf{A})]$ for some width parameter α and for t ranging from 0 to some maximum T. Specifically our algorithm satisfies the following:

▶ **Theorem 1** (Histogram Approximation – Informal). *Given $\mathbf{A} \in \mathbb{R}^{n\times d}$, let b_t be the number of squared singular values of $\mathbf{A}$ on the range $[(1-\alpha)^t\sigma_1^2(\mathbf{A}), (1-\alpha)^{t-1}\sigma_1^2(\mathbf{A})]$. Then given error parameter $\epsilon > 0$, with probability 99/100, Algorithm 1 outputs for all $t \in \{0, ..., T\}$, $\tilde{b}_t$ satisfying:*

$$(1-\epsilon)b_t \leq \tilde{b}_t \leq (1+\epsilon)b_t + \epsilon(b_{t-1} + b_{t+1}).$$

For input parameter $k \in \{1, ..., d\}$, let $\bar{\kappa} \stackrel{\text{def}}{=} \frac{k\sigma_k^2(\mathbf{A}) + \sum_{i=k+1}^{d}\sigma_i^2(\mathbf{A})}{d\cdot(1-\alpha)^T}$ and $\hat{\kappa} \stackrel{\text{def}}{=} \frac{\sigma_{k+1}^2(\mathbf{A})}{(1-\alpha)^T}$. Let $d_s(\mathbf{A})$

[1] Note that an exact SVD is incomputable even with exact arithmetic [65]. Nevertheless, direct methods for the SVD obtain superlinear convergence rates and hence are often considered to be 'exact'.

be the maximum number of nonzeros in a row of $\mathbf{A}$*. The algorithm's runtime is bounded by:*

$$\tilde{O}\left(\frac{\text{nnz}(\mathbf{A})k + dk^{\omega-1} + \sqrt{\text{nnz}(\mathbf{A})[d \cdot d_s(\mathbf{A}) + dk]\bar{\kappa}}}{\text{poly}(\epsilon, \alpha)}\right) \text{ or } \tilde{O}\left(\frac{\text{nnz}(\mathbf{A})k + dk^{\omega-1} + (\text{nnz}(\mathbf{A}) + dk)\lceil\sqrt{\bar{\kappa}}\rceil}{\text{poly}(\epsilon, \alpha)}\right)$$

for sparse $\mathbf{A}$ *or* $\tilde{O}\left(\frac{nd^{\gamma-1}+n^{1/2}d^{3/2}\sqrt{\bar{\kappa}})}{\text{poly}(\epsilon,\alpha)}\right)$ *for dense* $\mathbf{A}$*, where* d^γ *is the time it takes to multiply a* $d \times d$ *matrix by a* $d \times k$ *matrix using fast matrix multiplication.*

This primitive is useful on its own – the summary of $\mathbf{A}$'s spectrum which can be used in many downstream applications. Setting the parameter k appropriately to balance costs (see overview in Section 1.3), we use it to give the first $o(n^\omega)$ algorithms for computing $(1 \pm \epsilon)$ relative error approximations to a broad class of spectral sums for functions f, which are a) smooth and b) quickly growing, so that very small singular values cannot make a significant contribution to $\mathcal{S}_f(\mathbf{A})$. This class includes for example the Schatten p-norms for all $p > 0$, the SVD entropy, the Ky Fan norms, and many general Orlicz norms.

For a summary of our p-norm results see Table 1. Focusing for simplicity on square matrices, with uniformly sparse rows, and assuming ϵ, p are constants, our algorithms approximate $\|\mathbf{A}\|_p^p$ in $\tilde{O}(\text{nnz}(\mathbf{A}))$ time for any real $p \geq 2$.[2] For $p \leq 2$, we achieve $\tilde{O}\left(\text{nnz}(\mathbf{A})n^{\frac{1/p-1/2}{1/p+1/2}} + n^{\frac{4/p-1}{2/p+1}}\sqrt{\text{nnz}(\mathbf{A})}\right)$ runtime. In the important case of $p = 1$, this becomes $\tilde{O}\left(\text{nnz}(\mathbf{A})n^{1/3} + n\sqrt{\text{nnz}(\mathbf{A})}\right)$. Note that $n\sqrt{\text{nnz}(\mathbf{A})} \leq n^2$, and for sparse enough $\mathbf{A}$, this bound is subquadratic. For dense $\mathbf{A}$, we use fast matrix multiplication, achieving time $\tilde{O}\left(n^{\frac{2.3729-.0994p}{1+.0435p}}\right)$ for all $p < 2$. For $p = 1$, this gives $\tilde{O}(n^{2.18})$. Even without fast matrix multiplication, the runtime is $\tilde{O}(n^{2.33})$, and so $o(n^\omega)$ for $\omega \approx 2.3729$.

1.1.2 Lower Bounds

On the lower bound side, we show that obtaining $o(n^\omega)$ time spectrum approximation algorithms with very high accuracy may be difficult. Our runtimes all depend polynomially on the error ϵ, and we show that improving this, e.g., to $\log(1/\epsilon)$, or even to a better polynomial, would give faster algorithms for the well studied Triangle Detection problem.

Specifically, for a broad class of spectral sums, including all Schatten p-norms with $p \neq 2$, SVD entropy, $\log\det(\mathbf{A})$, $\text{tr}(\mathbf{A}^{-1})$, and $\text{tr}(\exp(\mathbf{A}))$, we show that any $(1 \pm \epsilon)$ approximation algorithm running in $O(n^\gamma \epsilon^{-c})$ time yields an algorithm for triangle detection running in $O(n^{\gamma+O(c)})$ time. For $\gamma < \omega$ and sufficiently small c, such an algorithm would improve the state of the art in triangle detection, which currently requires $\Theta(n^\omega)$ time on dense graphs. Furthermore, through a reduction of [72], any subcubic time triangle detection algorithm yields a subcubic time algorithm for Boolean Matrix Multiplication (BMM). Thus, any spectral sum algorithm achieving subcubic runtime and $\frac{1}{\epsilon^c}$ accuracy for small enough constant c, must (implicitly) implement fast matrix multiplication. This is in stark contrast to the fact that, for $c = 3$, for many spectral sums, including all Schatten-p with $p \geq 1/2$, we are able to obtain subcubic, and in fact $o(n^\omega)$ for $\omega = 2.3729$, runtimes without using fast matrix multiplication (see Table 1 for precise ϵ dependencies).

Our lower bounds hold even for well-conditioned matrices and structured matrices like symmetric diagonally dominant (SDD) systems, both of which admit nearly linear time algorithms for system solving [63]. This illustrates a dichotomy between linear algebraic

[2] For any $\mathbf{A} \in \mathbb{R}^{n \times d}$, $\text{nnz}(\mathbf{A})$ denotes the number of nonzero entries in $\mathbf{A}$.

Table 1 Summary of our results for approximating the Schatten-p norms. We define $f(p,\epsilon) = \min\{1, p^3\} \cdot \epsilon^{\max\{3,1+1/p\}}$, which appears as a factor in many of the bounds. The uniform sparsity assumption is that the maximum row sparsity $d_s(\mathbf{A}) \leq \frac{\xi}{n}\operatorname{nnz}(\mathbf{A})$ for some constant ξ. In our theorems, we give general runtimes, parameterized by ξ. When we do not have the uniform sparsity assumption, we are still able to give a $(1+\epsilon)$ approximation in $\tilde{O}(\epsilon^{-3}\operatorname{nnz}(\mathbf{A})\sqrt{n} + n^2)$ time for example for $\|\mathbf{A}\|_1$. We can also give $1/\gamma$ approximation for any constant $\gamma < 1$ by paying an $n^{\gamma/2}$ factor in our runtime. Note that for dense matrices, for all p we obtain $o(n^\omega)$ runtime, or $o(n^3)$ runtime if we do not use fast matrix multiplication. Theorems numbers reference the full paper [53].

p	Sparsity	Appx.	Runtime	Theorem
$p > 2$	uniform	$(1+\epsilon)$	$\tilde{O}\left(\operatorname{nnz}(\mathbf{A}) \cdot p/\epsilon^3\right)$	Thm 32
$p \leq 2$	uniform	$(1+\epsilon)$	$\tilde{O}\left(\frac{1}{f(p,\epsilon)}\left[\operatorname{nnz}(\mathbf{A})n^{\frac{1/p-1/2}{1/p+1/2}} + n^{\frac{4/p-1}{2/p+1}}\sqrt{\operatorname{nnz}(\mathbf{A})}\right]\right)$	Thm 33
$p \leq 2$	dense	$(1+\epsilon)$	$\tilde{O}\left(\frac{1}{f(p,\epsilon)}n^{\frac{2.3729-.0994p}{1+.0435p}}\right)$, $\tilde{O}\left(\frac{1}{f(p,\epsilon)}n^{\frac{3+p/2}{1+p/2}}\right)$ w/o FMM	Thm 31
$p > 0$	general	$(1+\epsilon)$	$\tilde{O}\left(\frac{1}{f(p,\epsilon)}\left[\operatorname{nnz}(\mathbf{A})n^{\frac{1}{1+p}} + n^{1+\frac{2}{1+p}}\right]\right)$	Thms 32, 33
$p > 2$	general	$1/\gamma$	$\tilde{O}(p\operatorname{nnz}(\mathbf{A}) \cdot n^\gamma)$	Thm 34
$p < 2$	general	$1/\gamma$	$\tilde{O}\left(\frac{1}{p^3}\left[\operatorname{nnz}(\mathbf{A})n^{\frac{1/p-1/2}{1/p+1/2}+\gamma/2} + \sqrt{\operatorname{nnz}(\mathbf{A})} \cdot n^{\frac{4/p-1}{2/p-1}}\right]\right)$	Thm 34

primitives like applying $\mathbf{A}^{-1}$ to a vector and spectral summarization tasks like precisely computing $\operatorname{tr}(\mathbf{A}^{-1})$. Our analysis has ramifications regarding natural open problems in graph theory and numerical computation. For example, for graph Laplacians, we show that accurately computing all *effective resistances* yields an accurate algorithm for computing $\operatorname{tr}(\mathbf{A}^{-1})$ of certain matrices, which is enough to give triangle detection.

1.2 Related Work on Spectral Sums

The applications of approximate spectral sum computation are broad. When $\mathbf{A}$ is positive semidefinite (PSD) and $f(x) = \log(x)$, $\mathcal{S}_f(\mathbf{A})$ is the log-determinant, which is important in machine learning and inference applications [57, 13, 19]. For $f(x) = 1/x$, $\mathcal{S}_f(\mathbf{A})$ is the trace of the inverse, used in uncertainty quantification [7] and quantum chromodynamics [64].

When $f(x) = x^p$, $S_f(\mathbf{A}) = \|\mathbf{A}\|_p^p$ where $\|\mathbf{A}\|_p$ is the Schatten p-norm of $\mathbf{A}$. Computation of the Schatten 1-norm, also known as the nuclear or trace norm, is required in a wide variety of applications. It is often used in place of the matrix rank in matrix completion algorithms and other convex relaxations of rank-constrained optimization problems [10, 15, 31, 54]. It appears as the 'graph energy' in theoretical chemistry [25, 26], the 'singular value bound' in differential privacy [28, 42], and in rank aggregation and collaborative ranking [48].

Similar to the nuclear norm, general Schatten p-norms are used in convex relaxations for rank-constrained optimization [55]. They also appear in image processing applications [76], classification [49], restoration [75], and feature extraction [17].

When $f(x) = -x\log x$ (after $\mathbf{A}$ is normalized by $\|\mathbf{A}\|_1$), $\mathcal{S}_f(\mathbf{A})$ is the SVD entropy [2], used in feature selection [69, 5], financial data analysis [11, 24], and genomic applications [2].

Despite their importance, prior to our work, few algorithms for fast computation of spectral sums existed. Only a few special cases of the Schatten p-norms were known to be computable in $o(n^\omega)$ time. The Frobenius norm ($p = 2$) is trivially computed in $O(\operatorname{nnz}(\mathbf{A}))$ time. The spectral norm ($p = \infty$) which can be estimated via the Lanczos method in $\tilde{O}(\operatorname{nnz}(\mathbf{A})\epsilon^{-\frac{1}{2}})$ time [38]. Finally, the Schatten-p norms for *even integers* $p > 2$, or general integers with PSD $\mathbf{A}$. These norms can be approximated in $O(\operatorname{nnz}(\mathbf{A})\epsilon^{-2})$ time via trace estimation [74, 9], since when p is even or $\mathbf{A}$ is PSD, $\mathbf{A}^p$ is PSD and so its trace equals $\|\mathbf{A}\|_p^p$.

There are a number of works which consider estimating matrix norms in sublinear space and with a small number of passes over $\mathbf{A}$ [44, 3, 45, 9, 46]. However, in these works, the main focus is on space complexity, and no non-trivial runtime bounds are given. We seem to be the first to tackle the arguably more fundamental problem of obtaining the best time complexity for simple norms like the Schatten-p norms.

Another interesting line of works tries to estimate the Schatten-p norms of an underlying covariance matrix from a small number of samples from the distribution [37], or from entrywise sampling under various incoherence assumptions [34]. This model is different from ours, as we do not assume an underlying distribution or any incoherence properties. Moreover, even with such assumptions, these algorithms also only give non-trivial sample complexity when either $\mathbf{A}$ is PSD and p is an integer, or $\mathbf{A}$ is a general matrix but p is an even integer, which as mentioned above are easy to handle from the perspective of time complexity alone.

A number of works have focused on computing spectral sums when $\mathbf{A}$ has bounded condition number, and relative error results exist for the log-determinant, $\mathrm{tr}(\exp(\mathbf{A}))$, $\mathrm{tr}(\mathbf{A}^{-1})$, and the Schatten p-norms [8, 27, 66]. We are the first to give relative error results in $o(\omega)$ time for general matrices, without the condition number dependence. Our histogram approach resembles spectral filtering and spectral density estimation techniques that have been considered in the numerical computation literature [77, 16, 47, 67, 68]. However, this literature typically requires assuming gaps between the singular values and existing work is not enough to give relative error spectral sum approximation for general matrices

1.3 Algorithmic Approach

We now give a high level overview of the techniques used in our algorithms.

1.3.1 Spectral Sums via Trace Estimation

A common approach to spectral sum approximation is to reduce to a trace estimation problem involving the PSD matrix $\mathbf{A}^T\mathbf{A}$, using the fact that the trace of this matrix equals the sum of its singular values. In fact, this has largely been the only known technique, other than the full SVD, for obtaining aggregate information about $\mathbf{A}$'s singular values [30, 64, 59, 18, 8, 27]. The idea is, letting $g(x) = f(x^{1/2})$, we have $\mathcal{S}_f(\mathbf{A}) = \mathcal{S}_g(\mathbf{A}^T\mathbf{A})$. Writing the SVD $\mathbf{A}^T\mathbf{A} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^T$, and defining the matrix function $g(\mathbf{A}^T\mathbf{A}) \stackrel{\text{def}}{=} \mathbf{U}g(\mathbf{\Lambda})\mathbf{U}^T$ where $[g(\mathbf{\Lambda})]_{i,i} = g([\mathbf{\Lambda}]_{i,i})$, we have $\mathcal{S}_g(\mathbf{A}^T\mathbf{A}) = \mathrm{tr}(g(\mathbf{A}^T\mathbf{A}))$ since, if $g(\cdot)$ is nonnegative, $g(\mathbf{A}^T\mathbf{A})$ is PSD and its trace equals the sum of its singular values.

It is well known that this trace can be approximated up to $(1 \pm \epsilon)$ accuracy by averaging $\tilde{O}(\epsilon^{-2})$ samples of the form $\mathbf{x}^T g(\mathbf{A}^T\mathbf{A})\mathbf{x}$ where $\mathbf{x}$ is a random Gaussian or sign vector [30, 4]. While $g(\mathbf{A}^T\mathbf{A})$ cannot be explicitly computed without a full SVD, a common approach is to approximate g with a low-degree polynomial ϕ [8, 27]. If ϕ has degree q, one can apply $\phi(\mathbf{A}^T\mathbf{A})$ to any vector $\mathbf{x}$ in $O(\mathrm{nnz}(\mathbf{A}) \cdot q)$ time, and so estimate its trace in just $O(\mathrm{nnz}(\mathbf{A}) \cdot \frac{q}{\epsilon^2})$ time. Unfortunately, for many of the functions most important in applications, e.g., $f(x) = x^p$ for odd p, $f(x) = x \log x$, $f(x) = x^{-1}$, $g(x)$ has a discontinuity at $x = 0$ and *cannot* be approximated well by a low-degree polynomial near zero. While the approximation only needs to be good in the range $[\sigma_n^2(\mathbf{A}), \sigma_1^2(\mathbf{A})]$, the required degree q will still typically depend on $\sqrt{\kappa}$ where $\kappa \stackrel{\text{def}}{=} \frac{\sigma_1^2(\mathbf{A})}{\sigma_n^2(\mathbf{A})}$ is the condition number, which can be unbounded in general.

1.3.2 Singular Value Deflation for Improved Conditioning

Our first observation is that, for many functions, it is not necessary to approximate $g(x)$ on the full spectral range. For example, for $g(x) = x^{p/2}$ (i.e., when $\mathcal{S}_g(\mathbf{A}^T\mathbf{A}) = \|\mathbf{A}\|_p^p$), setting $\lambda = (\frac{\epsilon}{n}\|\mathbf{A}\|_p^p)^{1/p}$:

$$\sum_{\{i|\sigma_i(\mathbf{A})\leq\lambda\}} \sigma_i(\mathbf{A})^p \leq n \cdot \frac{\epsilon}{n}\|\mathbf{A}\|_p^p \leq \epsilon\|\mathbf{A}\|_p^p.$$

Hence we can safely 'ignore' any $\sigma_i(\mathbf{A}) \leq \lambda$ and still obtain a relative error approximation to $\mathcal{S}_g(\mathbf{A}^T\mathbf{A}) = \|\mathbf{A}\|_p^p$. The larger p is, the larger we can set λ (corresponding to $(1-\alpha)^T$ in Theorem 1) to be, since, after powering, the singular values below this threshold do not contribute significantly to $\|\mathbf{A}\|_p^p$. For $\|\mathbf{A}\|_p^p$, our 'effective condition number' for approximating $g(x)$ becomes $\hat{\kappa} = \frac{\sigma_1^2(\mathbf{A})}{\lambda^2} = (\frac{n}{\epsilon})^{2/p} \cdot \frac{\sigma_1^2(\mathbf{A})}{\|\mathbf{A}\|_p^2}$. Unfortunately, in the worst case, we may have $\sigma_1(\mathbf{A}) \approx \|\mathbf{A}\|_p$ and hence $\sqrt{\hat{\kappa}} = (\frac{n}{\epsilon})^{1/p}$. Hiding ϵ dependences, this gives runtime $\tilde{O}(\text{nnz}(\mathbf{A}) \cdot n)$ when $p = 1$.

To improve the effective condition number, we can apply *singular vector deflation*. Our above bound on $\hat{\kappa}$ is only tight when $\sigma_1(\mathbf{A})$ is very large and so dominates $\|\mathbf{A}\|_p$. We can remedy this by flattening $\mathbf{A}$'s spectrum by deflating off the top k singular vectors (corresponding to k in Theorem 1), and including their values in the spectral sum directly.

Specifically, let $\mathbf{P}_k$ be the projection onto the top k singular vectors of $\mathbf{A}$ and consider the deflated matrix $\bar{\mathbf{A}} \stackrel{\text{def}}{=} \mathbf{A}(\mathbf{I} - \mathbf{P}_k)$, which has $\sigma_1(\bar{\mathbf{A}}) = \sigma_{k+1}(\mathbf{A})$. Importantly, $\sigma_{k+1}^p(\mathbf{A}) \leq \frac{1}{k}\|\mathbf{A}\|_p^p$, and so this singular value cannot dominate the p-norm. For example, considering $p = 1$ and ignoring ϵ dependencies, our effective condition number after deflation is

$$\hat{\kappa} = \frac{n^2 \cdot \sigma_{k+1}^2(\mathbf{A})}{\|\mathbf{A}\|_1^2} \leq \frac{n^2}{k^2} \tag{1}$$

The runtime required to approximate $\mathbf{P}_k$ via an iterative method (ignoring possible gains from fast matrix multiplication) is roughly $O(\text{nnz}(\mathbf{A})k + nk^2)$. We then require $\tilde{O}(\text{nnz}(\mathbf{A})\sqrt{\hat{\kappa}} + nk\sqrt{\hat{\kappa}})$ time to approximate the polynomial trace of $\bar{\mathbf{A}}^T\bar{\mathbf{A}}$. The $nk\sqrt{\hat{\kappa}}$ term comes from projecting off the top singular directions with each application of $\bar{\mathbf{A}}^T\bar{\mathbf{A}}$. Setting $k = \sqrt{n}$ to balance the costs, we obtain runtime $\tilde{O}(\text{nnz}(\mathbf{A})\sqrt{n} + n^2)$.

For $p \neq 1$ a similar argument gives runtime $\tilde{O}(\text{nnz}(\mathbf{A})n^{\frac{1}{p+1}} + n^{2+\frac{1}{p+1}})$. This is already a significant improvement over a full SVD. As p grows larger, the runtime approaches $\tilde{O}(\text{nnz}(\mathbf{A}))$ reflecting the fact that for larger p we can ignore a larger and larger portion of the small singular values in $\mathbf{A}$ and correspondingly deflate off fewer and fewer top values.

Unfortunately, we get stuck here. Considering the important Schatten-1 norm, for a matrix with $\sqrt{n}$ singular values each equal to $\sqrt{n}$ and $\Theta(n)$ singular values each equal to 1, the tail of small singular values contributes a constant fraction of $\|\mathbf{A}\|_1 = \Theta(n)$. However, there is no good polynomial approximation to $g(x) = x^{1/2}$ on the range $[1, n]$ with degree $o(\sqrt{n})$ (recall that we pick this function since $\mathcal{S}_g(\mathbf{A}^T\mathbf{A}) = \|\mathbf{A}\|_1$). So to accurately approximate $g(\mathbf{A}^T\mathbf{A})$, we either must deflate off all $\sqrt{n}$ top singular values, requiring $\Theta(\text{nnz}(\mathbf{A})\sqrt{n})$ time, or apply a $\Theta(\sqrt{n})$ degree polynomial approximation, requiring the same amount of time.

1.3.3 Further Improvements with Stochastic Gradient Descent

To push beyond this barrier, we look to *stochastic gradient* methods for linear systems. When using polynomial approximation, our bounds depend on the condition number of the interval over which we must approximate $g(\mathbf{A}^T\mathbf{A})$, after ignoring the smallest singular

values and deflating off the largest. This is analogous to the condition number dependence of iterative linear system solvers like conjugate gradient or accelerated gradient descent, which approximate $f(\mathbf{A}^T\mathbf{A})$ for $f = 1/x$ using a polynomial of $\mathbf{A}^T\mathbf{A}$.

However, recent advances in convex optimization offer an alternative. Stochastic gradient methods [32, 61] sample one row, $\mathbf{a}_i$, of $\mathbf{A}$ at a time, updating the current iterate by adding a multiple of $\mathbf{a}_i$. They trade a larger number of iterations for updates that take $O(\text{nnz}(\mathbf{a}_i))$ time, rather than $O(\text{nnz}(\mathbf{A}))$ time to multiply $\mathbf{A}$ by a vector. These methods give much finer dependencies on the singular value spectrum. Specifically, it is possible to approximately apply $(\mathbf{A}^T\mathbf{A})^{-1}$ to a vector with the number of iterations dependent on the *average condition number:* $\bar{\kappa} = \frac{\frac{1}{n}\sum_{i=1}^{n}\sigma_i^2(\mathbf{A})}{\sigma_n^2(\mathbf{A})}$. $\bar{\kappa}$ is always at most the standard condition number, $\kappa = \frac{\sigma_1^2(\mathbf{A})}{\sigma_n^2(\mathbf{A})}$. It can be significantly smaller when $\mathbf{A}$ has a quickly decaying spectrum, and hence $\frac{1}{n}\sum_{i=1}^{n}\sigma_i^2(\mathbf{A}) \ll \sigma_1^2(\mathbf{A})$. Further, the case of a quickly decaying spectrum with a few large and many small singular values is *exactly the hard case for our earlier approach.* If we can understand how to translate improvements on linear system solvers to spectral sum approximation, we can handle this hard case.

1.3.4 From Linear System Solvers to Histogram Approximation

The key idea to translating the improved average condition number bounds for linear systems to our problem of approximating $\mathcal{S}_f(\mathbf{A})$ is to note that linear system solvers can be used to apply threshold functions to $\mathbf{A}^T\mathbf{A}$.

Specifically, given any vector $\mathbf{y}$, we can first compute $\mathbf{A}^T\mathbf{A}\mathbf{y}$. We can then apply a fast system solver to approximate $(\mathbf{A}^T\mathbf{A}+\lambda\mathbf{I})^{-1}\mathbf{A}^T\mathbf{A}\mathbf{y}$. The matrix function $r_\lambda(\mathbf{A}^T\mathbf{A}) \stackrel{\text{def}}{=} (\mathbf{A}^T\mathbf{A}+\lambda\mathbf{I})^{-1}\mathbf{A}^T\mathbf{A}$ has a number of important properties. All its singular values are between 0 and 1. Further, any singular value in $\mathbf{A}^T\mathbf{A}$ with value $\geq \lambda$ is mapped to a singular value in $r_\lambda(\mathbf{A}^T\mathbf{A})$ which is $\geq 1/2$. Correspondingly, any singular value $< \lambda$ is mapped to $< 1/2$.

Thus, we can apply a low degree polynomial approximation to a step function at $1/2$ to $r_\lambda(\mathbf{A}^T\mathbf{A})$ to obtain $s_\lambda(\mathbf{A}^T\mathbf{A})$, which approximates a step function at λ [20]. For some steepness parameter γ which affects the degree of the polynomial approximation, for $x \geq (1+\gamma)\lambda$ we have $s_\lambda(x) \approx 1$ and for $x < (1-\gamma)\lambda$, $s_\lambda(x) \approx 0$. On the intermediate range $x \in [(1-\gamma)\lambda, (1+\gamma)\lambda]$, $s_\lambda(x)$ falls somewhere between 0 and 1.

Composing approximate threshold functions lets us 'split' our spectrum into a number of small spectral windows. For example, $s_a(\mathbf{A}^T\mathbf{A}) \cdot (\mathbf{I} - s_b(\mathbf{A}^T\mathbf{A}))$ is ≈ 1 on the range $[a, b]$ and ≈ 0 outside this range, with some ambiguity near a and b.

Splitting our spectrum into windows of the form $[(1-\alpha)^{t-1}, (1-\alpha)^t]$ for a width parameter α, and applying trace estimation on each window lets us produce an approximate spectral histogram. Of course, this histogram is not exact and in particular, the 'blurring' of our windows at their boundaries can introduce significant error. However, by applying a random shifting technique and setting the steepness parameter γ small enough (i.e., $1/\text{poly}(\alpha, \epsilon)$), we can ensure that most of the spectral weight falls outside these boundary regions with good probability, giving Theorem 1.

1.3.5 From Histogram Approximation to Spectral Sums

If α is small enough, and $f(\cdot)$ and correspondingly $g(\cdot)$ (where $g(x) = f(x^{1/2})$) are smooth enough, we can approximate $\mathcal{S}_f(\mathbf{A}) = \mathcal{S}_g(\mathbf{A}^T\mathbf{A})$ by simply summing over each window in the histogram, approximating $g(x)$ by its value at one end of the window.

This technique can be applied for any spectral sum. The number of windows required (controlled by α) and the histogram accuracy ϵ scale with the smoothness of $f(\cdot)$ and the desired accuracy in computing the sum, giving runtime dependencies on these parameters.

However, the most important factor determining the final runtime is the smallest value λ (corresponding to $(1-\alpha)^T$ in Theorem 1) which we must include in our histogram in order to approximate $S_f(\mathbf{A})$. The cost of computing the last window of the histogram is proportional to the cost of applying $s_\lambda(\mathbf{A}^T\mathbf{A})$, and hence of approximately computing $(\mathbf{A}^T\mathbf{A}+\lambda\mathbf{I})^{-1}\mathbf{A}^T\mathbf{A}\mathbf{y}$. Using stochastic gradient descent this depends on the average condition number of $(\mathbf{A}^T\mathbf{A}+\lambda\mathbf{I})$.

Again considering the Schatten 1-norm for illustration, we can ignore any singular values with $\sigma_i(\mathbf{A}) \leq \frac{\epsilon}{n}\|\mathbf{A}\|_1$. Hiding ϵ dependence, this means that in our histogram, we must include any singular values of $\mathbf{A}^T\mathbf{A}$ with value $\sigma_i(\mathbf{A}^T\mathbf{A}) = \sigma_i^2(\mathbf{A}) \geq \frac{1}{n^2}\|\mathbf{A}\|_1^2$. This gives us effective average condition number after deflating off the top k singular values:

$$\bar{\kappa} = \frac{n^2\sum_{i=k+1}^n \sigma_i^2(\mathbf{A})}{n\|\mathbf{A}\|_1^2} \leq \frac{n^2\sigma_{k+1}(\mathbf{A})\cdot\sum_{i=k+1}^n \sigma_i(\mathbf{A})}{n\|\mathbf{A}\|_1^2} \leq \frac{n}{k} \tag{2}$$

where the last inequality follows since $\sigma_{k+1}(\mathbf{A}) \leq \frac{1}{k}\|\mathbf{A}\|_1$ and $\sum_{i=k+1}^n \sigma_i(\mathbf{A}) \leq \|\mathbf{A}\|_1$. Comparing to (1), this bound is better by an n/k factor.

Ignoring details and using a simplification of the runtimes in Theorem 1, we obtain an algorithm running in $\tilde{O}(\text{nnz}(\mathbf{A})k + nk^2)$ time to deflate k singular vectors, along with $\tilde{O}\left(\text{nnz}(\mathbf{A})\sqrt{\bar{\kappa}} + \sqrt{\text{nnz}(\mathbf{A})nk\bar{\kappa}}\right)$ time to approximate the spectral sum over the deflated matrix. Choosing k to balance these costs, gives our final runtimes. For the nuclear norm, using the bound on $\bar{\kappa}$ from (2), we set $k = n^{1/3}$ which gives $\bar{\kappa} = n^{2/3}$ and runtime $\tilde{O}(\text{nnz}(\mathbf{A})n^{1/3} + n^{3/2}\sqrt{d_s})$ where $d_s \leq n$ is the maximum row sparsity. For dense $\mathbf{A}$ this is $\tilde{O}(n^{2.33})$, which is faster than state of the art matrix multiplication time. It can be further accelerated using fast matrix multiplication. See details in Section 7 of the full paper [53].

Returning to our original hard example for intuition, we have $\mathbf{A}$ with $\sqrt{n}$ singular values at $\sqrt{n}$ and $\Theta(n)$ singular values at 1. Even without deflation, we have (again ignoring ϵ dependencies) $\bar{\kappa} = \frac{\sum_{i=1}^n \sigma_i^2(\mathbf{A})}{n\lambda} = \frac{n\|\mathbf{A}\|_F^2}{\|\mathbf{A}\|_1^2}$. Since $\|\mathbf{A}\|_F^2 = \Theta(n^{3/2})$ and $\|\mathbf{A}\|_1^2 = \Theta(n^2)$, this gives $\bar{\kappa} = \Theta(\sqrt{n})$. Thus, we can actually approximate $\|\mathbf{A}\|_1$ in just $\tilde{O}(\text{nnz}(\mathbf{A})n^{1/4})$ time.

With average condition number dependence, our performance is limited by a new hard case. Consider $\mathbf{A}$ with $n^{1/3}$ singular values at $n^{2/3}$ and $\Theta(n)$ at 1. $\mathbf{A}$'s average condition number is $\frac{n\|\mathbf{A}\|_F^2}{\|\mathbf{A}\|_1^2} = \Theta\left(\frac{n^{5/3}}{n}\right) = \Theta(n^{2/3})$ giving $\sqrt{\bar{\kappa}} = \Theta(n^{1/3})$. Further, unless we deflate off nearly all $n^{1/3}$ top singular vectors, we do not improve this bound significantly.

1.4 Lower Bound Approach

We now shift focus to our lower bounds, which explore the fundamental limits of spectrum approximation using fine-grained complexity. Fine-grained complexity has had much success for graph problems, string problems, and problems in other areas (see, e.g., [71] for a survey), and is closely tied to understanding the complexity of matrix multiplication. However, to the best of our knowledge it has not been applied broadly to problems in linear algebra.

Existing hardness results for linear algebraic problems tend to apply to restricted computational models such as arithmetic circuits [6], bilinear circuits or circuits with bounded coefficients and number of divisions [51, 58], algorithms for dense linear systems that can only add multiples of rows to each other [35, 36], and algorithms with restrictions on the dimension of certain manifolds defined in terms of the input [73, 14]. In contrast, we obtain conditional lower bounds for arbitrary polynomial time algorithms by showing that faster algorithms for them imply faster algorithms for canonical hard problems.

1.4.1 From Schatten 3-norm to Triangle Detection

We start with the fact that the number of triangles in any unweighted graph G is equal to $\text{tr}(\mathbf{A}^3)/6$, where $\mathbf{A}$ is the adjacency matrix. Any algorithm for approximating $\text{tr}(\mathbf{A}^3)$ to high enough accuracy therefore gives an algorithm for detecting if a graph has a triangle.

$\mathbf{A}$ is not PSD, so $\text{tr}(\mathbf{A}^3)$ is actually not a function of $\mathbf{A}$'s singular values – it depends on the signs of $\mathbf{A}$'s eigenvalues. However, the graph Laplacian given by $\mathbf{L} = \mathbf{D} - \mathbf{A}$ where $\mathbf{D}$ is the diagonal degree matrix, is PSD and we have:

$$\|\mathbf{L}\|_3^3 = \text{tr}(\mathbf{L}^3) = \text{tr}(\mathbf{D}^3) - 3\,\text{tr}(\mathbf{D}^2\mathbf{A}) + 3\,\text{tr}(\mathbf{D}\mathbf{A}^2) - \text{tr}(\mathbf{A}^3).$$

$\text{tr}(\mathbf{D}^2\mathbf{A}) = 0$ since $\mathbf{A}$ has an all 0 diagonal. Further, it is not hard to see that $\text{tr}(\mathbf{D}\mathbf{A}^2) = \text{tr}(\mathbf{D}^2)$. So this term and $\text{tr}(\mathbf{D}^3)$ are easy to compute exactly. Thus, if we approximate $\|\mathbf{L}\|_3^3$ up to additive error 6, we can determine if $\text{tr}(\mathbf{A}^3) = 0$ or $\text{tr}(\mathbf{A}^3) \geq 6$ and so detect if G contains a triangle. $\|\mathbf{L}\|_3^3 \leq 8n^4$ for any unweighted graph on n nodes, and hence computing this norm up to $(1 \pm \epsilon)$ relative error for $\epsilon = 3/(6n^4)$ suffices to detect a triangle. If we have an $O(n^\gamma \epsilon^{-c})$ time $(1 \pm \epsilon)$ approximation algorithm for the Schatten 3-norm, we can thus perform triangle detection in $O(n^{\gamma+4c})$ time.

Our strongest algorithmic result for the Schatten 3-norm requires just $\tilde{O}(n^2/\epsilon^3)$ time for dense matrices. Improving the ϵ dependence to $o(1/\epsilon^{(\omega-2)/4}) = O(1/\epsilon^{.09})$ for the current value of ω, would yield an algorithm for triangle detection running in $o(n^\omega)$ time for general graphs, breaking a longstanding runtime barrier for this problem. Even a $1/\epsilon^{1/3}$ dependence would give a sub-cubic time triangle detection algorithm, and hence could be used to give a subcubic time matrix multiplication algorithm via the reduction of [72].

1.4.2 Generalizing to Other Spectral Sums

We can generalize the above approach to the Schatten 4-norm by adding λ self-loops to each node of G, which corresponds to replacing $\mathbf{A}$ with $\lambda\mathbf{I} + \mathbf{A}$. We then consider $\text{tr}((\lambda\mathbf{I} + \mathbf{A})^4) = \|\lambda\mathbf{I} + \mathbf{A}\|_4^4$. This is the sum over all vertices of the number of paths that start at v_i and return to v_i in four steps. All of these paths are either (1) legitimate four cycles, (2) triangles combined with self loops, or (3) combinations of self-loops and two-step paths from a vertex v_i to one of its neighbors and back. The number of type (3) paths is exactly computable using the node degrees and number of self loops. Additionally, if the number of self loops λ is large enough, the number of type (2) paths will dominate the number of type (1) paths, even if there is just a single triangle in the graph. Hence, an accurate approximation to $\|\lambda\mathbf{I} + \mathbf{A}\|_4^4$ will give us the number of type (2) paths, from which we can easily compute the number of triangles.

This argument extends to a very broad class of spectral sums by considering a power series expansion of $f(x)$ and showing that for large enough λ, $\text{tr}\left(f(\lambda\mathbf{I} + \mathbf{A})\right)$ is dominated by $\text{tr}(\mathbf{A}^3)$ along with some exactly computable terms. Thus, an accurate approximation to this spectral sum allows us to determine the number of triangles in G. This approach works for any $f(x)$ that can be represented as a power series, with reasonably well-behaved coefficients on some interval of $\mathbb{R}^+$, giving bounds for all $\|\mathbf{A}\|_p$ with $p \neq 2$, the SVD entropy, $\log\det(\mathbf{A})$, $\text{tr}(\mathbf{A}^{-1})$, and $\text{tr}(\exp(\mathbf{A}))$.

We further show that approximating $\text{tr}(\mathbf{A}^{-1})$ for the $\mathbf{A}$ used in our lower bound can be reduced to computing all effective resistances of a certain graph Laplacian up to $(1 \pm \epsilon)$ error. Thus, we rule out highly accurate (with $1/\epsilon^c$ dependence for small c) approximation algorithms for all effective resistances, despite the existence of linear time system solvers (with $\log(1/\epsilon)$ error dependence) for Laplacians [63]. Effective resistances and leverage scores

are quantities that have recently been crucial to achieving algorithmic improvements to fundamental problems like graph sparsification [62] and regression [43, 12]. While crude multiplicative approximations to the quantities suffice for these problems, more recently computing these quantities has been used to achieve breakthroughs in solving maximum flow and linear programming [40], cutting plane methods [41], and sampling random spanning trees [50]. In each of these settings having more accurate estimates would be a natural route to either simplify or possibly improve existing results; we show that this is unlikely to be successful if the precision requirements are two high.

1.5 Paper Outline

Section 2: Preliminaries. We review notations that will be used throughout.

Section 3: Spectral Windows. We show how to approximately restrict the spectrum of a matrix to a small window. This is our main primitive for accessing the spectrum.

Section 4: Spectral Histogram. We show how our spectral window algorithms can be used to compute an approximate spectral histogram. We give applications to approximating general spectral sums, including the Schatten-p norms, Orlicz norms, and Ky Fan norms.

The last three sections are including in our full paper [53].

Section 5: Lower Bounds. We prove lower bounds showing that highly accurate spectral sum algorithms can be used to give algorithms for triangle detection and matrix multiplication.

Section 6: Improved Algorithms via Polynomial Approximation. We demonstrate how to tighten ϵ dependencies in our runtimes using a polynomial approximation approach.

Section 7: Optimized Runtime Bounds. We instantiate the techniques of Section 6 give our best runtimes for the Schatten p-norms and SVD entropy.

2 Preliminaries

Here we outline notation and conventions used throughout the paper.

Matrix Properties: For $\mathbf{A} \in \mathbb{R}^{n \times d}$ we assume w.l.o.g. that $d \leq n$. We let $\sigma_1(\mathbf{A}) \geq \ldots \geq \sigma_d(\mathbf{A}) \geq 0$ denote the matrix's singular values, $\text{nnz}(\mathbf{A})$ denote the number of non-zero entries, and $d_s(\mathbf{A}) \in [\text{nnz}(\mathbf{A})/n, d]$ denote the maximum number of non-zero entries in a row.

Fast Matrix Multiplication: Let $\omega \approx 2.3729$ denote the current best exponent of fast matrix multiplication [70, 21]. Additionally, let $\omega(\gamma)$ denote the exponent such that it takes $O\left(d^{\omega(\gamma)}\right)$ time to multiply a $d \times d$ matrix by a $d \times d^\gamma$ matrix for any $\gamma \leq 1$. $\omega(\gamma) = 2$ for $\gamma < \alpha$ where $\alpha > 0.31389$ and $\omega(\gamma) = 2 + (\omega - 2)\frac{\gamma - \alpha}{1 - \alpha}$ for $\gamma \geq \alpha$ [39, 21]. For $\gamma = 1$, $\omega(\gamma) = \omega$.

Asymptotic Notation: We use $\tilde{O}(\cdot)$ notation to hide poly-logarithmic factors in the input parameters, including dimension, failure probability, and error ϵ. We use 'with high probability' or 'w.h.p.' to refer to events happening with probability at least $1 - 1/d^c$ for some constant c, where d is our smaller input dimension.

Other: We denote $[d] \stackrel{\text{def}}{=} \{0, \ldots, d\}$. For any $\mathbf{y} \in \mathbb{R}^d$ and PSD $\mathbf{N} \in \mathbb{R}^{d \times d}$, we denote $\|\mathbf{y}\|_{\mathbf{N}} \stackrel{\text{def}}{=} \sqrt{\mathbf{y}^T \mathbf{N} \mathbf{y}}$.

3 Approximate Spectral Windows via Ridge Regression

In this section, we give state-of-the-art results for approximating spectral windows over $\mathbf{A}$. As discussed, our algorithms will split $\mathbf{A}$'s spectrum into small slices using these window functions, performing trace estimation to estimate the number of singular values on each window and producing an approximate spectral histogram.

In Section 3.1 we show how to efficiently apply smooth approximations to threshold functions of the spectrum given access to an algorithm for solving regularized regression problems with the matrix. In Section 3.2 we then provide the fastest known algorithms for the regression problems in the given parameter regimes using both stochastic gradient methods and traditional solvers. Departing from our high level description in Section 1.3, we incorporate singular vector deflation directly into our system solvers to reduce condition number. This simplifies our final algorithms but has the same effect as the techniques discussed in Section 1.3. In Section 3.3 we give runtimes for applying smooth approximations to window functions of the spectrum, which is the main export of this section.

3.1 Step Function Approximation

To compute a window over $\mathbf{A}$'s spectrum, we will combine two threshold functions at the boundaries of the window. We begin by discussing how to compute these threshold functions.

Let $s_\lambda : [0,1] \to [0,1]$ be the threshold function at λ. $s_\lambda(x) = 1$ for $x \in [\lambda, 1]$ and 0 for $x \in [0, \lambda)$. For some gap γ we define a soft step function by:

▶ **Definition 2** (Soft Step Function). *$s_\lambda^\gamma : [0,1] \to [0,1]$ is a γ-soft step at $\lambda > 0$ if:*

$$s_\lambda^\gamma(x) = \begin{cases} 0 \text{ for } x \in [0, (1-\gamma)\lambda] \\ 1 \text{ for } x \in [\lambda, 1] \end{cases} \quad \text{and } s_\lambda^\gamma(x) \in [0,1] \text{ for } x \in [(1-\gamma)\lambda, \lambda]. \tag{3}$$

We use the strategy from [20], which, for $\mathbf{A}$ with $\|\mathbf{A}\|_2 \leq 1$ shows how to efficiently multiply a γ-soft step $s_\lambda^\gamma(\mathbf{A}^T\mathbf{A})$ by any $\mathbf{y} \in \mathbb{R}^d$ using ridge regression. The trick is to first approximately compute $\mathbf{A}^T\mathbf{A}(\mathbf{A}^T\mathbf{A} + \lambda\mathbf{I})^{-1}\mathbf{y} = r_\lambda(\mathbf{A}^T\mathbf{A})\mathbf{y}$ where $r_\lambda(x) \stackrel{\text{def}}{=} \frac{x}{x+\lambda}$. Then, note that $s_{1/2}(r_\lambda(x)) = s_\lambda(x)$. Additionally, the symmetric step function $s_{1/2}$ can be well approximated with a low degree polynomial. Specifically, there exists a polynomial of degree $O(\gamma^{-1}\log(1/(\gamma\epsilon)))$ that is within additive ϵ of a true γ-soft step at $1/2$ and can be applied stably such that any error in computing $r_\lambda(\mathbf{A}^T\mathbf{A})$ remains bounded. The upshot, following from Theorem 7.4 of [1] is:

▶ **Lemma 3** (Step Function via Ridge Regression). *Let $\mathcal{A}(\mathbf{A}, \mathbf{y}, \lambda, \epsilon)$ be an algorithm that on input $\mathbf{A} \in \mathbb{R}^{n\times d}$, $\mathbf{y} \in \mathbb{R}^d$, $\lambda, \epsilon > 0$ returns $\mathbf{x} \in \mathbb{R}^d$ such that $\left\|\mathbf{x} - (\mathbf{A}^T\mathbf{A} + \lambda\mathbf{I})^{-1}\mathbf{y}\right\|_2 \leq \epsilon \|\mathbf{y}\|_2$ with high probability. Then there is an algorithm $\mathcal{B}(\mathbf{A}, \mathbf{y}, \lambda, \gamma, \epsilon)$ which on input $\mathbf{A} \in \mathbb{R}^{n\times d}$ with $\|\mathbf{A}\|_2 \leq 1$, $\mathbf{y} \in \mathbb{R}^d$, $\lambda \in (0,1)$, and $\gamma, \epsilon > 0$, returns $\mathbf{x} \in \mathbb{R}^d$ with*

$$\left\|\mathbf{x} - s_\lambda^\gamma(\mathbf{A}^T\mathbf{A})\mathbf{y}\right\|_2 \leq \epsilon \|\mathbf{y}\|_2$$

where s_λ^γ is a γ-soft step at λ (i.e. satisfies Defn. 2). $\mathcal{B}(\mathbf{A}, \mathbf{y}, \lambda, \gamma, \epsilon)$ requires $O(\gamma^{-1}\log(1/\epsilon\gamma))$ calls to $\mathcal{A}(\mathbf{A}, \mathbf{y}, \lambda, \epsilon')$ along with $O(\text{nnz}(\mathbf{A}))$ additional runtime, where $\epsilon' = \text{poly}(1/(\gamma\epsilon))$.

3.2 Ridge Regression

Given Lemma 3, to efficiently compute $s_\lambda^\gamma(\mathbf{A}^T\mathbf{A})\mathbf{y}$ for $s_\lambda^\gamma(\cdot)$ satisfying Definition 2, it suffices to quickly approximate $(\mathbf{A}^T\mathbf{A} + \lambda\mathbf{I})^{-1}\mathbf{y}$ (i.e. to provide the algorithm $\mathcal{A}(\mathbf{A}, \mathbf{y}, \lambda, \epsilon)$ used in

the lemma). In this section we provide two theorems which give the state-of-the-art ridge regression running times achievable in our parameter regime, using sampling, acceleration, and singular value deflation.

Naively, computing $(\mathbf{A}^T\mathbf{A}+\lambda\mathbf{I})^{-1}\mathbf{y}$ using an iterative system solver involves a dependence on the condition number $\sigma_1^2(\mathbf{A})/\lambda$. In our theorems, this condition number is replaced by a deflated condition number depending on $\sigma_k^2(\mathbf{A})$ for some input parameter $k \in [d]$. We achieve this improved dependence following the techniques presented in [23]. We first approximate the top k singular vectors of $\mathbf{A}$ and then construct a preconditioner based on this approximation, which significantly flattens the spectrum of the matrix. By using this preconditioner in conjunction with a stochastic gradient based linear system solver, we further enjoy an average condition number dependence. The following theorem summarizes the results.

▶ **Theorem 4** (Ridge Regression – Accelerated Preconditioned SVRG). *For any $\mathbf{A} \in \mathbb{R}^{n\times d}$ and $\lambda > 0$, let $\mathbf{M}_\lambda \stackrel{\text{def}}{=} \mathbf{A}^T\mathbf{A} + \lambda\mathbf{I}$. Let $\bar{\kappa} \stackrel{\text{def}}{=} \frac{k\sigma_k^2(\mathbf{A})+\sum_{i=k+1}^{d}\sigma_i^2(\mathbf{A})}{d\lambda}$ where $k \in [d]$ is an input parameter. There is an algorithm that builds a preconditioner for $\mathbf{M}_\lambda$ using precomputation time $\tilde{O}(\text{nnz}(\mathbf{A})k + dk^{\omega-1})$ for sparse $\mathbf{A}$ or $\tilde{O}(nd^{\omega(\log_d k)-1})$ time for dense $\mathbf{A}$, and for any input $\mathbf{y} \in \mathbb{R}^d$, returns $\mathbf{x}$ such that with high probability $\left\|\mathbf{x} - \mathbf{M}_\lambda^{-1}\mathbf{y}\right\|_{\mathbf{M}_\lambda} \leq \epsilon \left\|\mathbf{y}\right\|_{\mathbf{M}_\lambda^{-1}}$ in*

$$\tilde{O}\left(\text{nnz}(\mathbf{A}) + \sqrt{\text{nnz}(\mathbf{A})[d \cdot d_s(\mathbf{A}) + dk]\bar{\kappa}}\right)$$

time for sparse $\mathbf{A}$ or $\tilde{O}\left(nd + n^{1/2}d^{3/2}\sqrt{\bar{\kappa}})\right)$ time for dense $\mathbf{A}$.

We give a proof in Appendix A of the full paper [53]. Note that the ϵ dependence in the runtime is $\log(1/\epsilon)$ and so is hidden by the $\tilde{O}(\cdot)$ notation.

When $\mathbf{A}$ is dense, the runtime of Theorem 4 is essentially the best known. Due to its average condition number dependence, the method always outperforms traditional iterative methods, like conjugate gradient, up to log factors. However, in the sparse case, traditional approaches can give faster runtimes if the rows of $\mathbf{A}$ are not uniformly sparse and $d_s(\mathbf{A})$ is large. We have the following, also proved in Appendix A of the full paper using the same deflation-based preconditioner as in Theorem 4:

▶ **Theorem 5** (Ridge Regression – Preconditioned Iterative Method). *For any $\mathbf{A} \in \mathbb{R}^{n\times d}$ and $\lambda > 0$, let $\mathbf{M}_\lambda \stackrel{\text{def}}{=} \mathbf{A}^T\mathbf{A}+\lambda\mathbf{I}$ and $\hat{\kappa} \stackrel{\text{def}}{=} \frac{\sigma_{k+1}^2(\mathbf{A})}{\lambda}$ where $k \in [d]$ is an input parameter. There is an algorithm that builds a preconditioner for $\mathbf{M}_\lambda$ using precomputation time $\tilde{O}(\text{nnz}(\mathbf{A})k+dk^{\omega-1})$, and for any input $\mathbf{y} \in \mathbb{R}^d$, returns $\mathbf{x}$ such that with high probability $\left\|\mathbf{x} - \mathbf{M}_\lambda^{-1}\mathbf{y}\right\|_{\mathbf{M}_\lambda} \leq \epsilon \left\|\mathbf{y}\right\|_{\mathbf{M}_\lambda^{-1}}$ in $\tilde{O}\left((\text{nnz}(\mathbf{A}) + dk)\lceil\sqrt{\hat{\kappa}}\rceil\right)$ time.*

3.3 Overall Runtimes For Spectral Windows

Combined with Lemma 3, the ridge regression routines above let us efficiently compute soft step functions of $\mathbf{A}$'s spectrum. Composing step functions then gives our key computational primitive: the ability to approximate soft window functions that restrict $\mathbf{A}$'s spectrum to a specified range. We first define our notion of soft window functions and then discuss runtimes. The corresponding Theorem 7 is our main tool for spectrum approximation.

▶ **Definition 6** (Soft Window Function). Given $0 < a < b$, and $\gamma \in [0, 1]$, $h_{[a,b]}^\gamma : [0, 1] \to [0, 1]$ is a γ-soft window for $[a, b]$ if $h_{[a,b]}^\gamma(x) \in [0, 1]$ for $x \in [(1 - \gamma)a, a] \cup [b, (1 + \gamma)b]$ and:

$$h_{[a,b]}^\gamma(x) = \begin{cases} 1 \text{ for } x \in [a, b] \\ 0 \text{ for } x \in [0, (1 - \gamma)a] \cup [(1 + \gamma)b, 1] \end{cases}$$

▶ **Theorem 7** (Spectral Windowing). *For $\mathbf{A} \in \mathbb{R}^{n\times d}$ with $\|\mathbf{A}\|_2 \leq 1$, $\mathbf{y} \in \mathbb{R}^d$, and $a, b, \gamma, \epsilon \in (0, 1]$, with $a < b$, there is an algorithm $\mathcal{W}(\mathbf{A}, \mathbf{y}, a, b, \gamma, \epsilon)$ that returns $\mathbf{x}$ satisfying w.h.p.:*

$$\left\|\mathbf{x} - h^{\gamma}_{[a,b]}(\mathbf{A}^T\mathbf{A})\mathbf{y}\right\|_2 \leq \epsilon \|\mathbf{y}\|_2$$

where $h^{\gamma}_{[a,b]}$ is a soft window function satisfying Def. 6. Let $\bar{\kappa} \stackrel{\text{def}}{=} \frac{k\sigma_k^2(\mathbf{A}) + \sum_{i=k+1}^{d}\sigma_i^2(\mathbf{A})}{d\cdot a}$ and $\hat{\kappa} \stackrel{\text{def}}{=} \frac{\sigma_{k+1}^2(\mathbf{A})}{a}$ where $k \in [d]$ is an input parameter. The algorithm uses precomputation time $\tilde{O}(\text{nnz}(\mathbf{A})k + dk^{\omega-1})$ for sparse $\mathbf{A}$ or $\tilde{O}(nd^{\omega(\log_d k)-1})$ for dense $\mathbf{A}$ after which given any $\mathbf{y}$ it returns $\mathbf{x}$ in time:

$$\tilde{O}\left(\frac{\text{nnz}(\mathbf{A}) + \sqrt{\text{nnz}(\mathbf{A})[d\cdot d_s(\mathbf{A}) + dk]\bar{\kappa}}}{\gamma}\right) \quad \text{or} \quad \tilde{O}\left(\frac{(\text{nnz}(\mathbf{A}) + dk)\lceil\sqrt{\hat{\kappa}}\rceil}{\gamma}\right)$$

for sparse $\mathbf{A}$ or $\tilde{O}\left(\frac{nd + n^{1/2}d^{3/2}\sqrt{\bar{\kappa}})}{\gamma}\right)$ for dense $\mathbf{A}$.

Proof. If $b \geq 1/(1+\gamma)$ then we can simply define $h^{\gamma}_{[a,b]}(x) = s^{\gamma}_a(x)$ for any s^{γ}_a satisfying Definition 2. Otherwise, given soft steps s^{γ}_a and $s^{\gamma/2}_{(1+\gamma)b}$ satisfying Definition 2, we can define $h^{\gamma}_{[a,b]}(x) = s^{\gamma}_a(x)\cdot(1 - s^{\gamma/2}_{(1+\gamma)b}(x))$. Since $\frac{\gamma}{2} \leq \frac{\gamma}{1+\gamma}$ we can verify that this will be a valid soft window function for $[a, b]$ (i.e. satisfy Definition 6). Further, we have for any $\mathbf{y} \in \mathbb{R}^d$:

$$h^{\gamma}_{[a,b]}(\mathbf{A}^T\mathbf{A})\mathbf{y} = s^{\gamma}_a(\mathbf{A}^T\mathbf{A})(\mathbf{I} - s^{\gamma/2}_{(1+\gamma)b}(\mathbf{A}^T\mathbf{A}))\mathbf{y}. \tag{4}$$

We can compute $s^{\gamma}_a(\mathbf{A}^T\mathbf{A})\mathbf{y}$ and $s^{\gamma/2}_{(1+\gamma)b}(\mathbf{A}^T\mathbf{A})\mathbf{y}$ each up to error $\epsilon\|\mathbf{y}\|_2$ via Lemma 3. This gives the error bound in the theorem, since we have both $\left\|s^{\gamma}_a(\mathbf{A}^T\mathbf{A})\right\|_2 \leq 1$ and $\left\|\mathbf{I} - s^{\gamma/2}_{(1+\gamma)b}(\mathbf{A}^T\mathbf{A})\right\|_2 \leq 1$ so the computation in (4) does not amplify error. Our runtime follows from combining Theorems 4 and 5 with $\lambda = a, b$ with Lemma 3. The errors in these theorems are measured with respect to $\|\cdot\|_{\mathbf{M}_\lambda}$. To obtain the error in $\|\cdot\|_2$ as used by Lemma 3, we simply apply the theorems with $\epsilon' = \epsilon\kappa(\mathbf{M}_\lambda)$ which incurs an additional $\log(\kappa(\mathbf{M}_\lambda))$ cost. Since $a < b$ the runtime is dominated by the computation of $s^{\gamma}_a(\mathbf{A}^T\mathbf{A})\mathbf{y}$, which depends on the condition number $\bar{\kappa} \stackrel{\text{def}}{=} \frac{k\sigma_k^2(\mathbf{A}) + \sum_{i=k+1}^{d}\sigma_i^2(\mathbf{A})}{d\cdot a}$ when using SVRG (Theorem 4) or $\hat{\kappa} \stackrel{\text{def}}{=} \frac{\sigma_{k+1}^2(\mathbf{A})}{a}$ for a traditional iterative solver (Theorem 5). ◀

4 Approximating Spectral Sums via Spectral Windows

We now use the window functions discussed in Section 3 to compute an approximate spectral histogram of $\mathbf{A}$. We give our main histogram algorithm and approximation guarantee in Section 4.1. In Section 4.2 we show how this guarantee translates to accurate spectral sum approximation for any smooth and sufficiently quickly growing function $f(x)$. In Section 4.3 we apply this general result to approximating the Schatten p-norms for all real $p > 0$. We give applications to bounded Orlicz norms and the Ky Fan norms in our full paper [53].

4.1 Approximate Spectral Histogram

Our main histogram approximation method is given as Algorithm 1. The algorithm is reasonably simple. Assuming $\|\mathbf{A}\|_2 \leq 1$ (this is w.l.o.g. as we can just scale the matrix), and given cutoff λ, below which we will not evaluate $\mathbf{A}'s$ spectrum, we split the range $[\lambda, 1]$ into successive windows $R_0, ..., R_T$ where $R_t = [a_1(1-\alpha)^t, a_1(1-\alpha)^{t-1}]$. Here α determines the

Algorithm 1 Approximate Spectral Histogram

Input: $\mathbf{A} \in \mathbb{R}^{n \times d}$ with $\|\mathbf{A}\|_2 \leq 1$, accuracy parameters $\epsilon_1, \epsilon_2 \in (0,1)$, width parameter $\alpha \in (0,1)$, and minimum singular value parameter $\lambda \in (0,1)$.

Output: Set of range boundaries $a_{T+1} < a_T < ... < a_1 < a_0$ and counts $\{\tilde{b}_0, ..., \tilde{b}_T\}$ where $\tilde{b}_t$ approximates the number of squared singular values of $\mathbf{A}$ on $[a_{t+1}, a_t]$.

Set $\gamma = c_1\epsilon_2\alpha$, $T = \lceil \log_{(1-\alpha)} \lambda \rceil$, and $S = \frac{\log n}{c_2\epsilon_1^2}$.
Set $a_0 = 1$ and choose a_1 uniformly at random in $[1 - \alpha/4, 1]$.
Set $a_t = a_1(1-\alpha)^{t-1}$ for $2 \leq t \leq T+1$.
for $t = 0 : T$ **do** ▷ Iterate over histogram buckets.
  Set $\tilde{b}_t = 0$. ▷ Initialize bucket size estimate.
  for $s = 1 : S$ **do** ▷ Estimate bucket size via trace estimation.
    Choose $\mathbf{y} \in \{-1, 1\}^d$ uniformly at random.
    Set $\tilde{b}_t = \tilde{b}_t + \frac{1}{S} \cdot \mathbf{y}^T \mathcal{W}(\mathbf{A}^T\mathbf{A}, \mathbf{y}, a_{t+1}, a_t, \gamma, c_3\epsilon_1^2/n)$. ▷ Apply soft window via Thm 7.
    If $\tilde{b}_t \leq 1/2$ set $\tilde{b}_t = 0$. ▷ Round small estimates to ensure relative error.
  end for
end for
return a_1 and $\tilde{b}_t$ for $t = 0 : T$. ▷ Output histogram representation.

width of our windows. In our final spectral approximation algorithms, we will set $\alpha = \Theta(\epsilon)$. a_1 is a random shift, which insures that, in expectation, the boundaries of our soft windows do not overlap too many singular values. This argument requires that most of the range $[\lambda, 1]$ is not covered by boundary regions. Thus, we set the steepness parameter $\gamma = \Theta(\epsilon_2\alpha)$ where ϵ_2 will control the error introduced by the boundaries. Finally, we iterate over each window, applying trace estimation to approximate the singular value count in each window.

In our final algorithms, the number of windows and samples required for trace estimation will be $\tilde{O}(\text{poly}(1/\epsilon))$. The dominant runtime cost comes from the lowest range R_T, which incurs a dependence on the condition number of $\mathbf{A}^T\mathbf{A} + a_T\mathbf{I}$ with $a_T = \Theta(\lambda)$.

▶ **Theorem 8** (Histogram Approximation). *Let a_1, $\tilde{b}_0, ..., \tilde{b}_T$ be output by Algorithm 1. Let $R_0 = [a_1, 1]$, $R_t = [a_1(1-\alpha)^t, a_1(1-\alpha)^{t-1}]$ for $t \geq 1$, and $b_t = \left|\{i : \sigma_i^2(\mathbf{A}) \in R_t\}\right|$ be the number of squared singular values of $\mathbf{A}$ on the range R_t. Then, for sufficiently small c_1, c_2, c_3, with probability 99/100, for all $t \in \{0, ..., \lceil \log_{(1-\alpha)} \lambda \rceil\}$, $\tilde{b}_t$ output by Algorithm 1 satisfies:*

$$(1-\epsilon_1)b_t \leq \tilde{b}_t \leq (1+\epsilon_1)b_t + \lceil \log_{(1-\alpha)} \lambda \rceil \cdot \epsilon_2(b_{t-1} + b_{t+1}).$$

That is, $\tilde{b}_t$ approximates the number of singular values of the range R_t up to multiplicative $(1 \pm \epsilon_1)$ error and additive error $\lceil \log_{(1-\alpha)} \lambda \rceil \cdot \epsilon_2(b_{t-1} + b_{t+1})$. Note that by setting $\epsilon_2 \leq \frac{\epsilon_1}{\lceil \log_{(1-\alpha)} \lambda \rceil}$, the error on each bucket is just multiplicative on its size plus the size of the two adjacent buckets, which contain singular values in nearby ranges. For simplicity we assume $\mathbf{A}$ passed to the algorithm has $\|\mathbf{A}\|_2 \leq 1$. This is w.l.o.g.: we can estimate $\|\mathbf{A}\|_2$ in $\tilde{O}(\text{nnz}(\mathbf{A}))$ time via the power or Lanczos methods [38, 52], and scale down the matrix appropriately.

The runtime of Algorithm 1 is dominated by the calls to $\mathcal{W}$ for the bucket corresponding to the smallest singular values, with $a_T = \Theta(\lambda)$. This runtime is given by Theorem 7. Since balancing the deflation parameter k with the minimum squared singular value λ considered can be complex, we wait to instantiate full runtimes until employing Algorithm 1 for specific spectral sum computations. See full paper [53] for a proof of Theorem 8.

4.2 Application to General Spectral Sums

While Theorem 8 is useful on its own, we now apply it to approximate a broad class of spectral sums. We need two assumptions. First, for the histogram discretization to be relatively accurate, we need the sum function to be smooth. Second, it is expensive to compute the histogram over very small singular values of $\mathbf{A}$ (i.e. with λ very small in Algorithm 1) as this makes the condition number in Theorem 7 large. So it is important that small singular values cannot contribute significantly to our sum. We start with the following definition:

▶ **Definition 9** (Multiplicative Smoothness). *$f : \mathbb{R}^+ \to \mathbb{R}^+$ is δ_f-multiplicatively smooth if for some $\delta_f \geq 1$, for all x, $|f'(x)| \leq \delta_f \frac{f(x)}{x}$.*

For the Schatten-p norm, $f(x) = x^p$, $f'(x) = px^{p-1}$ and so f is p-multiplicatively smooth. We have the following claim, proven in Appendix D of the full paper [53]:

▶ **Claim 10.** *Let f be a δ_f-multiplicatively smooth function. For all $x, y \in \mathbb{R}^+$ and $c \in (0, \frac{1}{3\delta_f})$*

$$y \in [(1-c)x, (1+c)x] \Rightarrow f(y) \in [(1-3\delta_f c)f(x), (1+3\delta_f c)f(x)].$$

We now give our general approximation theorem, showing that any spectral sum depending on sufficiently smooth and rapidly growing f can be computed using Algorithm 1:

▶ **Theorem 11** (Spectral Sums Via Approximate Histogram). *Consider any $\mathbf{A} \in \mathbb{R}^{n \times d}$ and any function $f : \mathbb{R}^+ \to \mathbb{R}^+$ satisfying:*

- *Multiplicative Smoothness: For some $\delta_f \geq 1$, f is δ_f-multiplicatively smooth (Defn. 9).*
- *Small Tail: For any $\epsilon > 0$ there exists $\lambda_f(\epsilon)$ such that for $x \in [0, \lambda_f(\epsilon)]$, $f(x) \leq \frac{\epsilon}{n}\mathcal{S}_f(\mathbf{A})$*

Given error parameter $\epsilon \in (0,1)$ and spectral norm estimate $M \in [\|\mathbf{A}\|_2, 2\|\mathbf{A}\|_2]$, for sufficiently small constant c, if we run Algorithm 1 on $\frac{1}{M}\mathbf{A}$ with input parameters $\epsilon_1, \epsilon_2 = c\epsilon$, $\alpha = c\epsilon/\delta_f$ and $\lambda = \lambda_f(c\epsilon)^2/M^2$ then with probability $99/100$, letting a_1, $\tilde{b}_0, ..., \tilde{b}_T$ be the outputs of the algorithm and $g(x) = f(x^{1/2})$:

$$(1-\epsilon)\mathcal{S}_f(\mathbf{A}) \leq \sum_{t=0}^{T} g(M^2 \cdot a_1(1-\alpha)^t) \cdot \tilde{b}_t \leq (1+\epsilon)\mathcal{S}_f(\mathbf{A}).$$

For parameter $k \in [d]$, letting $\bar{\kappa} \stackrel{\text{def}}{=} \frac{k\sigma_k^2(\mathbf{A}) + \sum_{i=k+1}^{d}\sigma_i^2(\mathbf{A})}{d \cdot \lambda}$ and $\hat{\kappa} \stackrel{\text{def}}{=} \frac{\sigma_{k+1}^2(\mathbf{A})}{\lambda}$, the algorithm runs in

$$\tilde{O}\left(\text{nnz}(\mathbf{A})k + dk^{\omega-1} + \frac{\text{nnz}(\mathbf{A}) + \sqrt{\text{nnz}(\mathbf{A})[d \cdot d_s(\mathbf{A}) + dk]\bar{\kappa}}}{\epsilon^5/(\delta_f^2 \log(1/\lambda))}\right)$$

$$\text{or } \tilde{O}\left(\text{nnz}(\mathbf{A})k + dk^{\omega-1} + \frac{(\text{nnz}(\mathbf{A}) + dk)\lceil\sqrt{\hat{\kappa}}\rceil}{\epsilon^5/(\delta_f^2 \log(1/\lambda))}\right)$$

time for sparse $\mathbf{A}$ or $\tilde{O}\left(nd^{\omega(\log_d k)-1} + \frac{nd + n^{1/2}d^{3/2}\sqrt{\bar{\kappa}})}{\epsilon^5/(\delta_f^2 \log(1/\lambda))}\right)$ for dense $\mathbf{A}$.

That is, we accurately approximate $\mathcal{S}_f(\mathbf{A})$ by discretizing over the histogram of Algorithm 1. We can boost our probability of success to $1-\delta$ by repeating the algorithm $\Theta(\log(1/\delta))$ times and taking the median of the outputs. Theorem 11 is proven in our full paper [53].

4.3 Application to Schatten-p Norms

Theorem 11 is very general, allowing us to approximate any function satisfying a simple smoothness condition as long as the smaller singular values of $\mathbf{A}$ cannot contribute significantly to $\mathcal{S}_f(\mathbf{A})$. As an example, we show how it gives the fastest known algorithms for Schatten-p norm estimation. We will not go into all runtime tradeoffs now as our best runtimes will be worked out in detail in Sections 6 and 7 of the full paper [53].

▶ **Corollary 12** (Schatten-p norms via Histogram Approximation). *For any $\mathbf{A} \in \mathbb{R}^{n\times n}$ with uniformly sparse rows (i.e. $d_s(\mathbf{A}) = O(\text{nnz}(\mathbf{A})/n)$), given error parameter $\epsilon \in (0,1)$ and $M \in [\|\mathbf{A}\|_2, 2\|\mathbf{A}\|_2]$, if we run Algorithm 1 on $\frac{1}{M}\mathbf{A}$ with $\epsilon_1, \epsilon_2 = c\epsilon$, $\alpha = c\epsilon/\max\{1,p\}$ and $\lambda = \frac{1}{M^2}\left(\frac{c\epsilon}{n}\|\mathbf{A}\|_p^p\right)^{2/p}$ for sufficiently small constant c then with probability $99/100$, $(1-\epsilon)\|\mathbf{A}\|_p^p \leq \sum_{t=0}^{T}\left[M^2 a_1(1-\alpha)^t\right]^{p/2}\cdot\tilde{b}_t \leq (1+\epsilon)\|\mathbf{A}\|_p^p$. The algorithm's runtime is:*

$$\tilde{O}\left(\frac{\text{nnz}(\mathbf{A})p^2}{\epsilon^{5+1/p}}\right) \text{ for } p \geq 2 \quad \text{and} \quad \tilde{O}\left(\frac{\text{nnz}(\mathbf{A})n^{\frac{1/p-1/2}{1/p+1/2}} + n^{\frac{5/p-1/2}{2/p+1}}\sqrt{d_s(\mathbf{A})}}{p\cdot\epsilon^{5+1/p}}\right) \text{ for } p \leq 2.$$

For dense inputs this can be sped up to $\tilde{O}\left(\frac{n^{\frac{2.3729-.1171p}{1+.0346p}}}{p\cdot\epsilon^{5+1/p}}\right)$ using fast matrix multiplication.

For constant $\epsilon, p > 2$ the first runtime is $\tilde{O}(\text{nnz}(\mathbf{A}))$, and for the nuclear norm ($p = 1$), for constant ϵ the second runtime gives $\tilde{O}(\text{nnz}(\mathbf{A})n^{1/3} + n^{3/2}\sqrt{d_s(\mathbf{A})})$ which is at worst $\tilde{O}(\text{nnz}(\mathbf{A})n^{1/3} + n^2)$. For dense matrices, the nuclear norm estimation time is $\tilde{O}(n^{2.18})$ using fast matrix multiplication. It is already $\tilde{O}(n^{2.33})$, without using fast matrix multiplication.

Note that we can compute the spectral norm approximation used to scale $\mathbf{A}$ via the Lanczos or power method in $\tilde{O}(\text{nnz}(\mathbf{A}))$ time. λ depends on $\|\mathbf{A}\|_p$ which we are estimating. However, as we will discuss in the proof, we can use a rough estimate for $\|\mathbf{A}\|_p$ which suffices. λ could also be identified via binary search. We can start with $\lambda = \sigma_1(\mathbf{A})^2/M^2$ and successively decrease λ running Algorithm 1 up to the stated runtime bounds. If it does not finish in the allotted time, we know that we have set λ too small. Thus, we can output the result with the smallest λ such that the algorithm completes within in the stated bounds.

Proof. We invoke Theorem 11 with $f(x) = x^p$. We have $f'(x) = p\frac{f(x)}{x}$ so $\delta_f = \max\{1,p\}$ and our setting of $\alpha = c\epsilon/\max\{1,p\}$ suffices. Additionally, for any c, we can set $\lambda_f(c\epsilon) = \left(\frac{c\epsilon}{n}\|\mathbf{A}\|_p^p\right)^{1/p} = \frac{c^{1/p}\epsilon^{1/p}}{n^{1/p}}\|\mathbf{A}\|_p$ and so our setting of λ suffices. Thus the accuracy bound follows from Theorem 11. We now consider runtime. For $p \geq 2$:

$$\bar{\kappa} = \frac{k\sigma_k^2(\mathbf{A}) + \sum_{i=k+1}^{n}\sigma_i^2(\mathbf{A})}{n\lambda} \leq \frac{n^{2/p-1}}{\epsilon^{2/p}}\cdot\frac{\|\mathbf{A}\|_F^2}{\|\mathbf{A}\|_p^2}.$$

We can bound $\|\mathbf{A}\|_F \leq n^{1/2-1/p}\|\mathbf{A}\|_p$ and so have $\bar{\kappa} \leq \frac{1}{\epsilon^{2/p}}$. For $p < 2$ we have:

$$\bar{\kappa} = \frac{n^{2/p-1}}{\epsilon^{2/p}}\cdot\frac{k\sigma_k^2(\mathbf{A}) + \sum_{i=k+1}^{n}\sigma_i^2(\mathbf{A})}{\|\mathbf{A}\|_p^2} \leq \frac{n^{2/p-1}}{\epsilon^{2/p}}\cdot\frac{\sigma_k^{2-p}(\mathbf{A})\sum_{i=1}^{n}\sigma_i^p(\mathbf{A})}{\|\mathbf{A}\|_p^2} = \frac{n^{2/p-1}}{\epsilon^{2/p}}\cdot\frac{\sigma_k^{2-p}(\mathbf{A})}{\|\mathbf{A}\|_p^{2-p}}.$$

Using the fact that $\sigma_k^p(\mathbf{A}) \leq \frac{1}{k}\|\mathbf{A}\|_p^p$ we have the tradeoff between k and $\bar{\kappa}$:

$$\bar{\kappa} \leq \frac{1}{\epsilon^{2/p}}\left(\frac{n}{k}\right)^{2/p-1}. \tag{5}$$

As mentioned, λ depends on the value of $\|\mathbf{A}\|_p$. We can simply lower bound $\|\mathbf{A}\|_p^p$ by $k\sigma_k^p(\mathbf{A})$, which we estimate up to multiplicative error when performing deflation. We can use

this lower bound to set λ. Our estimated λ will only be smaller than the true value, giving a better approximation guarantee and the above condition number bound will still hold.

Recall that for $f(x) = x^p$, $\delta_f = \max\{1, p\}$. Correspondingly, $\log(1/\lambda) = \tilde{O}(\max\{1, 1/p\})$ and so $\delta_f^2 \log(1/\lambda) = \max\{p^2, 1/p\}$. Plugging into the first runtime of Theorem 11, using the uniform sparsity assumption and the fact that $\sqrt{x+y} \leq \sqrt{x} + \sqrt{y}$ we have:

$$\tilde{O}\left(\text{nnz}(\mathbf{A})k + nk^{\omega-1} + \frac{\text{nnz}(\mathbf{A})\sqrt{\bar{\kappa}} + \sqrt{\text{nnz}(\mathbf{A})nk\bar{\kappa}}}{\epsilon^5/(\max\{p^2, 1/p\}}\right).$$

For $p \geq 2$ we just set $k = 0$ and have $\tilde{O}(\text{nnz}(\mathbf{A})p^2/\epsilon^{5+1/p})$ runtime by our bound $\bar{\kappa} \leq \frac{1}{\epsilon^{2/p}}$. For $p \leq 2$, not trying to optimize $\text{poly}(1/\epsilon)$ terms, we write the runtime as

$$\tilde{O}\left(nd_s(\mathbf{A})k + nk^{\omega-1} + \frac{nd_s(\mathbf{A})\sqrt{\bar{\kappa}} + n\sqrt{d_s(\mathbf{A})k\bar{\kappa}}}{\epsilon^5 p}\right).$$

Balancing the first two coefficients on n, set $k = n^{\frac{1/p-1/2}{1/p+1/2}}$ which gives $\sqrt{\bar{\kappa}} = n^{\frac{1/p-1/2}{1/p+1/2}}$ by (5) and so $nd_s(\mathbf{A})k = nd_s(\mathbf{A})\sqrt{\bar{\kappa}}$. We then have $n\sqrt{d_s(\mathbf{A})k\bar{\kappa}} = n\sqrt{d_s(\mathbf{A})}k^{3/2} = n^{\frac{5/p-1/2}{2/p+1}}\sqrt{d_s(\mathbf{A})}$. Finally, the $nk^{\omega-1}$ is dominated by the $n\sqrt{d_s(\mathbf{A})}k^{3/2}$ term so we drop it.

Finally, for dense $\mathbf{A}$ we apply the third runtime which gives

$$\tilde{O}\left(n^{\omega(\log_n k)} + \frac{n^2\sqrt{\bar{\kappa}}}{\epsilon^5 p}\right) = \tilde{O}\left(n^{\omega(\log_n k)} + \frac{n^{3/2+1/p-(\log_n k)(1/p-1/2)}}{\epsilon^{5+1/p} \cdot p}\right).$$

We now balance the terms, again ignoring ϵ dependence. Writing $\gamma = \log_n k$, $\omega(\gamma) = 2$ for $\gamma < \alpha$ where $\alpha > 0.31389$ and $2 + (\omega - 2)\frac{\gamma-\alpha}{1-\alpha}$ for $\gamma \geq \alpha$ [21]. Assuming $\gamma > \alpha$ we set: $2+(\omega-2)\frac{\gamma-\alpha}{1-\alpha} = \frac{3}{2} + \frac{1}{p} - \frac{\gamma}{p} + \frac{\gamma}{2}$ which gives $\gamma \approx \frac{1/p-.3294}{1/p+.0435} > \alpha$ for all $p < 2$ (so our assumption that $\gamma \geq \alpha$ was valid.) This yields total runtime $\tilde{O}\left(n^{\frac{2.3729-.0994p}{1+.0435p}}\right)$. Without using fast matrix multiplication, the first term in the runtime becomes n^2k and so we balance costs by setting: $n^{2+\gamma} = n^{3/2+1/p-\gamma/p+\gamma/2}$ which gives $\gamma = \frac{1/p-1/2}{1/p+1/2}$ and total runtime $\tilde{O}\left(n^{\frac{3+p/2}{1+p/2}}\right)$. ◀

References

1. Zeyuan Allen-Zhu and Yuanzhi Li. Faster principal component regression and stable matrix Chebyshev approximation. *Proceedings of the 34th International Conference on Machine Learning (ICML)*, 2017.
2. Orly Alter, Patrick O Brown, and David Botstein. Singular value decomposition for genome-wide expression data processing and modeling. *Proceedings of the National Academy of Sciences*, 97(18):10101–10106, 2000.
3. Alexandr Andoni, Robert Krauthgamer, and Ilya P. Razenshteyn. Sketching and embedding are equivalent for norms. In *Proceedings of the 47th Annual ACM Symposium on Theory of Computing (STOC)*, pages 479–488, 2015.
4. Haim Avron and Sivan Toledo. Randomized algorithms for estimating the trace of an implicit symmetric positive semi-definite matrix. *Journal of the ACM*, 58(2):8, 2011.
5. Monami Banerjee and Nikhil R Pal. Feature selection with SVD entropy: some modification and extension. *Information Sciences*, 264:118–134, 2014.
6. Walter Baur and Volker Strassen. The complexity of partial derivatives. *Theoretical Computer Science*, 22(3):317–330, 1983.
7. Constantine Bekas, Alessandro Curioni, and I Fedulova. Low cost high performance uncertainty quantification. In *Proceedings of the 2nd Workshop on High Performance Computational Finance*, page 8. ACM, 2009.

8 Christos Boutsidis, Petros Drineas, Prabhanjan Kambadur, and Anastasios Zouzias. A randomized algorithm for approximating the log determinant of a symmetric positive definite matrix. *Linear Algebra and its Applications*, 533:95–119, 2017.

9 Vladimir Braverman, Stephen R Chestnut, Robert Krauthgamer, and Lin F Yang. Sketches for matrix norms: Faster, smaller and more general. *arXiv:1609.05885*, 2016.

10 Emmanuel J. Candès and Benjamin Recht. Exact matrix completion via convex optimization. *Communications of the ACM*, 55(6):111–119, 2012.

11 Petre Caraiani. The predictive power of singular value decomposition entropy for stock market dynamics. *Physica A: Statistical Mechanics and its Applications*, 393:571–578, 2014.

12 Michael B. Cohen, Yin Tat Lee, Cameron Musco, Christopher Musco, Richard Peng, and Aaron Sidford. Uniform sampling for matrix approximation. In *Proceedings of the 6th Conference on Innovations in Theoretical Computer Science (ITCS)*, pages 181–190, 2015.

13 Jason V Davis, Brian Kulis, Prateek Jain, Suvrit Sra, and Inderjit S Dhillon. Information-theoretic metric learning. In *Proceedings of the 24th International Conference on Machine Learning (ICML)*, pages 209–216, 2007.

14 James Demmel. An arithmetic complexity lower bound for computing rational functions, with applications to linear algebra. *submitted to SIMAX*, 2013.

15 Amit Deshpande, Madhur Tulsiani, and Nisheeth K. Vishnoi. Algorithms and hardness for subspace approximation. In *Proceedings of the 22nd Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 482–496, 2011.

16 Edoardo Di Napoli, Eric Polizzi, and Yousef Saad. Efficient estimation of eigenvalue counts in an interval. *Numerical Linear Algebra with Applications*, 2016.

17 Haishun Du, Qingpu Hu, Manman Jiang, and Fan Zhang. Two-dimensional principal component analysis based on Schatten p-norm for image feature extraction. *Journal of Visual Communication and Image Representation*, 32:55–62, 2015.

18 JK Fitzsimons, MA Osborne, SJ Roberts, and JF Fitzsimons. Improved stochastic trace estimation using mutually unbiased bases. *arXiv:1608.00117*, 2016.

19 Jerome Friedman, Trevor Hastie, and Robert Tibshirani. Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, 9(3):432–441, 2008.

20 Roy Frostig, Cameron Musco, Christopher Musco, and Aaron Sidford. Principal component projection without principal component analysis. In Maria-Florina Balcan and Kilian Q. Weinberger, editors, *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*, volume 48 of *JMLR Workshop and Conference Proceedings*, pages 2349–2357. JMLR.org, 2016. URL: `http://jmlr.org/proceedings/papers/v48/frostig16.html`.

21 François Le Gall and Florent Urrutia. Improved rectangular matrix multiplication using powers of the Coppersmith-Winograd tensor. *arXiv:1708.05622*, 2017.

22 Gene H. Golub and Charles F Van Loan. *Matrix computations*, volume 3. JHU Press, 2012.

23 Alon Gonen, Francesco Orabona, and Shai Shalev-Shwartz. Solving ridge regression using sketched preconditioned SVRG. In *Proceedings of the 33rd International Conference on Machine Learning (ICML)*, 2016.

24 Rongbao Gu, Wei Xiong, and Xinjie Li. Does the singular value decomposition entropy have predictive power for stock market? evidence from the Shenzhen stock market. *Physica A: Statistical Mechanics and its Applications*, 439:103–113, 2015.

25 Ivan Gutman. Total π-electron energy of benzenoid hydrocarbons. In *Advances in the Theory of Benzenoid Hydrocarbons II*, pages 29–63. Springer, 1992.

26 Ivan Gutman. The energy of a graph: old and new results. In *Algebraic Combinatorics and Applications*, pages 196–211. Springer, 2001.

27 Insu Han, Dmitry Malioutov, Haim Avron, and Jinwoo Shin. Approximating the spectral sums of large-scale matrices using Chebyshev approximations. *SIAM Journal on Scientific Computing*, 39(4), 2017.

28 Moritz Hardt, Katrina Ligett, and Frank McSherry. A simple and practical algorithm for differentially private data release. In Peter L. Bartlett, Fernando C. N. Pereira, Christopher J. C. Burges, Léon Bottou, and Kilian Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25: 26th Annual Conference on Neural Information Processing Systems 2012. Proceedings of a meeting held December 3-6, 2012, Lake Tahoe, Nevada, United States.*, pages 2348–2356, 2012.

29 Nicholas J Higham. *Functions of matrices: theory and computation*. SIAM, 2008.

30 Michael F Hutchinson. A stochastic estimator of the trace of the influence matrix for Laplacian smoothing splines. *Communications in Statistics-Simulation and Computation*, 19(2):433–450, 1990.

31 Prateek Jain, Praneeth Netrapalli, and Sujay Sanghavi. Low-rank matrix completion using alternating minimization. In *Proceedings of the 45th Annual ACM Symposium on Theory of Computing (STOC)*, pages 665–674, 2013.

32 Rie Johnson and Tong Zhang. Accelerating stochastic gradient descent using predictive variance reduction. In *Advances in Neural Information Processing Systems 26 (NIPS)*, pages 315–323, 2013.

33 Ian Jolliffe. *Principal component analysis*. Wiley Online Library, 2002.

34 Ashish Khetan and Sewoong Oh. Matrix norm estimation from a few entries. In *Advances in Neural Information Processing Systems 30 (NIPS)*, 2017.

35 V. V. Klyuev and N. I. Kokovkin-Shcherbak. Minimization of the number of arithmetic operations in the solution of linear algebra systems of equations. *USSR Computational Mathematics and Mathematical Physics*, 5(1):25–43, 1965.

36 N. I. Kokovkin-Shcherbak. Minimization of numerical algorithms for solving arbitrary systems of linear equations. *Ukrainskii Matematicheskii Zhurnal*, 22(4):494–502, 1970.

37 Weihao Kong and Gregory Valiant. Spectrum estimation from samples. *Annals of Statistics*, 2016.

38 J Kuczyński and H Woźniakowski. Estimating the largest eigenvalue by the power and Lanczos algorithms with a random start. *SIAM Journal on Matrix Analysis and Applications*, 13(4):1094–1122, 1992.

39 François Le Gall. Faster algorithms for rectangular matrix multiplication. In *Proceedings of the 53rd Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, 2012.

40 Yin Tat Lee and Aaron Sidford. Path finding methods for linear programming: Solving linear programs in õ(vrank) iterations and faster algorithms for maximum flow. In *Proceedings of the 55th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 424–433, 2014.

41 Yin Tat Lee, Aaron Sidford, and Sam Chiu-wai Wong. A faster cutting plane method and its implications for combinatorial and convex optimization. In *Proceedings of the 56th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, 2015.

42 Chao Li and Gerome Miklau. Measuring the achievable error of query sets under differential privacy. *arXiv:1202.3399*, 2012.

43 Mu Li, Gary L. Miller, and Richard Peng. Iterative row sampling. In *Proceedings of the 54th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, 2013.

44 Yi Li, Huy L. Nguyen, and David P. Woodruff. On sketching matrix norms and the top singular vector. In *Proceedings of the 46th Annual ACM Symposium on Theory of Computing (STOC)*, pages 1562–1581, 2014.

45 Yi Li and David P. Woodruff. On approximating functions of the singular values in a stream. In *Proceedings of the 48th Annual ACM Symposium on Theory of Computing (STOC)*, 2016.

46 Yi Li and David P. Woodruff. Embeddings of schatten norms with applications to data streams. 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 60:1–60:14. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2017. doi:10.4230/LIPIcs.ICALP.2017.60.

47 Lin Lin, Yousef Saad, and Chao Yang. Approximating spectral densities of large matrices. *SIAM Review*, 58(1):34–65, 2016.

48 Yu Lu and Sahand N Negahban. Individualized rank aggregation using nuclear norm regularization. In *2015 53rd Annual Allerton Conference on Communication, Control, and Computing*, pages 1473–1479. IEEE, 2015.

49 Lei Luo, Jian Yang, Jinhui Chen, and Yicheng Gao. Schatten p-norm based matrix regression model for image classification. In *Pattern Recognition*. Springer, 2014.

50 Aleksander Madry, Damian Straszak, and Jakub Tarnawski. Fast generation of random spanning trees and the effective resistance metric. In *Proceedings of the 26th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 2019–2036, 2015.

51 Jacques Morgenstern. Note on a lower bound on the linear complexity of the fast Fourier transform. *Journal of the ACM (JACM)*, 20(2):305–306, 1973.

52 Cameron Musco and Christopher Musco. Randomized block Krylov methods for stronger and faster approximate singular value decomposition. In *Advances in Neural Information Processing Systems 28 (NIPS)*, pages 1396–1404, 2015.

53 Cameron Musco, Praneeth Netrapalli, Aaron Sidford, Shashanka Ubaru, and David P Woodruff. Spectrum approximation beyond fast matrix multiplication: Algorithms and hardness. *arXiv:1704.04163*, 2017.

54 Praneeth Netrapalli, UN Niranjan, Sujay Sanghavi, Animashree Anandkumar, and Prateek Jain. Non-convex robust PCA. In *Advances in Neural Information Processing Systems 27 (NIPS)*, pages 1107–1115, 2014.

55 Feiping Nie, Heng Huang, and Chris Ding. Low-rank matrix recovery via efficient Schatten p-norm minimization. In *Twenty-Sixth AAAI Conference on Artificial Intelligence*, 2012.

56 Beresford N Parlett. *The symmetric eigenvalue problem*. SIAM, 1998.

57 Carl Edward Rasmussen. Gaussian processes in machine learning. In *Advanced Lectures on Machine Learning*, pages 63–71. Springer, 2004.

58 Ran Raz and Amir Shpilka. Lower bounds for matrix product in bounded depth circuits with arbitrary gates. *SIAM Journal on Computing*, 32(2):488–513, 2003.

59 Farbod Roosta-Khorasani and Uri Ascher. Improved bounds on sample size for implicit matrix trace estimators. *Foundations of Computational Mathematics*, 2015.

60 Yousef Saad. *Numerical Methods for Large Eigenvalue Problems: Revised Edition*, volume 66. SIAM, 2011.

61 Shai Shalev-Shwartz and Tong Zhang. Accelerated proximal stochastic dual coordinate ascent for regularized loss minimization. In *Proceedings of the 31st International Conference on Machine Learning (ICML)*, pages 64–72, 2014.

62 Daniel A. Spielman and Nikhil Srivastava. Graph sparsification by effective resistances. In *Proceedings of the 40th Annual ACM Symposium on Theory of Computing (STOC)*, 2008.

63 Daniel A Spielman and Shang-Hua Teng. Nearly-linear time algorithms for graph partitioning, graph sparsification, and solving linear systems. In *Proceedings of the 36th Annual ACM Symposium on Theory of Computing (STOC)*, pages 81–90, 2004.

64 Andreas Stathopoulos, Jesse Laeuchli, and Kostas Orginos. Hierarchical probing for estimating the trace of the matrix inverse on toroidal lattices. *SIAM Journal on Scientific Computing*, 35(5):S299–S322, 2013.

65 Lloyd N. Trefethen and David Bau. *Numerical Linear Algebra*. SIAM, 1997.

66 Shashanka Ubaru, Jie Chen, and Yousef Saad. Fast estimation of $tr(f(a))$ via stochastic Lanczos quadrature. *SIAM Journal on Matrix Analysis and Applications (SIMAX)*, 2017.

67 Shashanka Ubaru and Yousef Saad. Fast methods for estimating the numerical rank of large matrices. In *Proceedings of the 33rd International Conference on Machine Learning (ICML)*, pages 468–477, 2016.

68 Shashanka Ubaru, Yousef Saad, and Abd-Krim Seghouane. Fast estimation of approximate matrix ranks using spectral densities. *Neural Computation*, 2017.

69 Roy Varshavsky, Assaf Gottlieb, Michal Linial, and David Horn. Novel unsupervised feature filtering of biological data. *Bioinformatics*, 22(14):e507–e513, 2006.

70 Virginia Vassilevska Williams. Multiplying matrices faster than Coppersmith-Winograd. In *Proceedings of the 44th Annual ACM Symposium on Theory of Computing (STOC)*, 2012.

71 Virginia Vassilevska Williams. Hardness of easy problems: Basing hardness on popular conjectures such as the strong exponential time hypothesis (invited talk). In *10th International Symposium on Parameterized and Exact Computation, IPEC 2015, September 16-18, 2015, Patras, Greece*, pages 17–29, 2015.

72 Virginia Vassilevska Williams and Ryan Williams. Subcubic equivalences between path, matrix and triangle problems. In *Proceedings of the 51st Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 645–654, 2010.

73 Shmuel Winograd. *Arithmetic Complexity of Computations*. CBMS-NSF Regional Conference Series in Applied Mathematics. SIAM, 1987. doi:10.1137/1.9781611970364.

74 David P. Woodruff. Sketching as a tool for numerical linear algebra. *Foundations and Trends in Theoretical Computer Science*, 10(1-2):1–157, 2014.

75 Y. Xie, Y. Qu, D. Tao, W. Wu, Q. Yuan, and W. Zhang. Hyperspectral image restoration via iteratively regularized weighted Schatten p-norm minimization. *IEEE Transactions on Geoscience and Remote Sensing*, PP(99):1–18, 2016.

76 Yuan Xie, Shuhang Gu, Yan Liu, Wangmeng Zuo, Wensheng Zhang, and Lei Zhang. Weighted Schatten p-norm minimization for image denoising and background subtraction. *IEEE transactions on Image Processing*, 25(10):4842–4857, 2016.

77 Yuchen Zhang, Martin J Wainwright, and Michael I Jordan. Distributed estimation of generalized matrix rank: Efficient algorithms and lower bounds. In *Proceedings of the 32nd International Conference on Machine Learning (ICML)*, 2015.

Size, Cost, and Capacity: A Semantic Technique for Hard Random QBFs*

Olaf Beyersdorff[1], Joshua Blinkhorn[2], and Luke Hinde[3]

1 **School of Computing, University of Leeds, UK**
o.beyersdorff@leeds.ac.uk
2 **School of Computing, University of Leeds, UK**
scjlb@leeds.ac.uk
3 **School of Computing, University of Leeds, UK**
sclpeh@leeds.ac.uk

Abstract

As a natural extension of the SAT problem, an array of proof systems for quantified Boolean formulas (QBF) have been proposed, many of which extend a propositional proof system to handle universal quantification. By formalising the construction of the QBF proof system obtained from a propositional proof system by adding universal reduction (Beyersdorff, Bonacina & Chew, ITCS '16), we present a new technique for proving proof-size lower bounds in these systems. The technique relies only on two semantic measures: the *cost* of a QBF, and the *capacity* of a proof. By examining the capacity of proofs in several QBF systems, we are able to use the technique to obtain lower bounds based on cost alone. As applications of the technique, we first prove exponential lower bounds for a new family of simple QBFs representing equality. The main application is in proving exponential lower bounds with high probability for a class of randomly generated QBFs, the first 'genuine' lower bounds of this kind, which apply to the QBF analogues of resolution, Cutting Planes, and Polynomial Calculus. Finally, we employ the technique to give a simple proof of hardness for the prominent formulas of Kleine Büning, Karpinski and Flögel.

1998 ACM Subject Classification F.2.2 Nonnumerical Algorithms and Problems: Complexity of proof procedures

Keywords and phrases quantified Boolean formulas, proof complexity, lower bounds

Digital Object Identifier 10.4230/LIPIcs.ITCS.2018.9

1 Introduction

The central question in *proof complexity* can be stated as follows: Given a logical theory and a provable theorem, what is the size of the shortest proof? This question bears tight connections to central problems in computational complexity [19, 25] and bounded arithmetic [42, 24].

Proof complexity is intrinsically linked to recent noteworthy innovations in solving, owing to the fact that any decision procedure implicitly defines a *proof system* for the underlying language. Relating the two fields in this way is illuminating for the practitioner; proof-size and proof-space lower bounds correspond directly to best-case running time and memory consumption for the corresponding solver. Indeed, proof complexity theory has become the main driver for the asymptotic comparison of solving implementations. However, in line with neighbouring fields (such as computational complexity), it is the central task of

* This is an extended abstract of the paper available at [7] https://128.84.21.199/abs/1712.03626.

demonstrating lower bounds, and of *developing general methods* for showing such results, that proves most challenging for theoreticians.

The desire for general techniques derives from the exceptional strength of modern implementations. Cutting-edge advances in solving, spearheaded by unparalleled progress in Boolean satisfiability (SAT), appear to provide a means for the efficient solution of computationally hard problems [54]. Contemporary SAT solvers routinely dispatch instances in millions of clauses [43], and are effectively employed as **NP**-oracles in more complex settings [44]. The state-of-the-art procedure is based on a propositional proof system called *resolution*, operating on *conjunctive normal form* (CNF) instances using a technique known as *conflict-driven clause learning* (CDCL) [50]. Besides furthering the intense study of resolution and its fragments [19], the evident success has inevitably pushed research frontiers beyond the **NP**-completeness of Boolean satisfiability.

1.1 Beyond propositional satisfiability

A case in point is the logic of *quantified Boolean formulas* (QBF), a theoretically important class that forms the prototypical **PSPACE**-complete language [53]. QBF extends propositional logic with existential and universal quantification, and consequently offers succinct encodings of concrete problems from conformant planning [48, 30, 20], ontological reasoning [40], and formal verification [6], amongst other areas [28, 17, 52]. There is a large body of work on practical QBF solving, and the relative complexities of the associated resolution-type proof systems are well understood [2, 10, 37].

The semantics of QBF has a neat interpretation as a two-player *evaluation game*. Given a QBF $\mathcal{Q} \cdot \phi$, the $\exists$- and $\forall$-players take turns to assign the existential and universal variables of the formula following the order of the quantifier prefix $\mathcal{Q}$. When all variables are assigned, the $\exists$-player wins if the propositional formula ϕ is satisfied; otherwise, the $\forall$-player takes the win. A folklore result states that a QBF is false if and only if the $\forall$-player can win the evaluation game by force; that is, if and only if there exists a winning strategy for the universal player. The concept of *strategy extraction* originates from QBF solving [35], whereby a winning strategy 'extracted' from the proof certifies the truth or falsity of the instance. In practice it is not merely the truth value of the QBF that is required – for real-world applications, certificates provide further useful information [52].

A major paradigm in QBF practice is *quantified conflict-driven clause learning* (QCDCL) [34], a natural extension of CDCL. The vast majority of QBF solvers build upon existing SAT techniques in a similar fashion. Such a notion can hardly be surprising when one considers that an existentially quantified QBF is merely a propositional formula. The novel challenge for the QBF practitioner, therefore, and the real test of a solver's strength, is in the handling of universal quantification.

Proof-theoretic analysis of associated QBF proof systems makes this notion abundantly clear. Consider *QU-Resolution* (QU-Res) [39, 33], a well-studied QBF proof system closely related to QCDCL solving.[1] That calculus simply extends propositional resolution with a *universal reduction* rule, which allows universal literals to be deleted from clauses under certain conditions. On existentially quantified QBFs, therefore, QU-Res is identical to resolution, and proof-size lower bounds for the latter lift immediately to the former. From the viewpoint of quantified logic, lower bounds obtained in this way are rightly considered

[1] The calculus QU-Res, proposed by Van Gelder in [33], generalises Q-Res, introduced by Kleine Büning et al. in [39], by allowing resolution over universally quantified pivots.

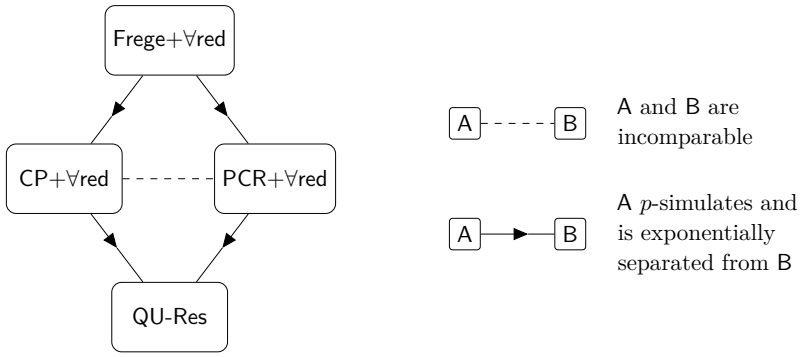


Figure 1 The simulation order of the four QBF proof systems featured in this paper. A proof system A p-simulates the system B if each B-proof of a formula Φ can be translated in polynomial time into an A-proof of Φ [25]. If neither A nor B p-simulates the other, then they are incomparable.

non-genuine; they belong in the realm of propositional proof complexity, and tell us nothing about the relative strengths of resolution-based QBF solvers.

Universal reduction is applicable to many suitable propositional proof systems P, giving rise to a general model for QBF systems in the shape of P+∀red [8], which adds to the propositional rules of P the universal reduction rule '∀red'. As a consequence, the phenomenon of genuineness extends well beyond resolution. In this paper, in addition to resolution we consider three stronger systems: Cutting Planes (CP), a well-studied calculus that works with linear inequalities; the algebraic system Polynomial Calculus (with Resolution, PCR); and Frege's eponymous 'textbook' system for propositional logic. Their simulation order is depicted in Figure 1.

What is generally desired (and seemingly elusive) in the QBF community is the development of *general* techniques for *genuine* lower bounds. The current work embraces maximal generality, and contributes a new technique for genuine QBF lower bounds in the general setting of P+∀red.

1.2 When is a lower bound genuine?

Naturally, the aforementioned objections to non-genuine QBF lower bounds may be raised in the abstract setting of P+∀red, as that system encompasses the propositional proof system P. Indeed, given any unsatisfiable propositional formulas that require large proofs in P, one can easily construct any number of contrived QBF families – even with arbitrarily many quantifier alternations – each of which require large proofs in P+∀red, but whose hardness stems from the original propositional formulas. That such lower bounds ought to be identified as non-genuine was highlighted in [21] (cf. also [15]).

The essential point in such cases is that the proofs are large simply because they require many propositional inferences, i.e. many applications of rules of P. Large proofs that do not harbour propositional hardness of this type must therefore contain many universal reductions. Thus, we are brought naturally to a pleasant characterisation of genuine hardness in P+∀red: Genuinely hard QBFs require superpolynomially-many universal reduction steps; all other lower bounds are non-genuine.[2]

[2] This notion can be made formal, as in the oracle model of [15].

In summary, a lower bound on the number of universal reduction steps is always genuine. The technique we introduce in this paper works by counting universal reduction steps, and we therefore deal exclusively in genuine results.

1.3 Random formulas

In the design and testing of solvers, large sets of formulas are needed to make effective comparisons between implementations. While many formulas have been constructed by hand, often representing some combinatorial principle, it is of clear benefit to have a procedure to randomly generate such formulas. The search for a better understanding of when such formulas are likely to be true or false, and their likely hardness for solvers, brings us to the study of the proof complexity of random CNFs and QBFs.

In propositional proof complexity, random 3-SAT instances, the most commonly studied random CNFs, are relatively well understood. There is a constant r such that if a random CNF on n variables contains more than rn clauses, then the CNF is unsatisfiable with probability approaching 1 [32]; the upper bound for r has regularly been improved (see [29], and references therein for previous upper bounds). Further, if the number of clauses is below $n^{6/5-\epsilon}$, the CNF requires exponential-size resolution refutations with high probability [3]. Hardness results for random CNFs are also known for Polynomial Calculus [1, 4] and for Cutting Planes [36, 31].

In contrast, comparatively little is known about randomly generated QBFs. The addition of universally quantified variables raises questions as to what model should be used to generate such QBFs – care is needed to ensure a suitable balance between universal and existential variables.[3] The best-studied model is that of (1,2)-QCNFs [22], for which bounds on the threshold number of clauses needed for a false QBF were shown in [26]. However, to the best of our knowledge, nothing has yet been shown on the proof complexity of randomly generated QBFs. Proving such lower bounds constitutes the major application of our new technique.

1.4 Our contributions

The primary contribution of this work is the proposal of a *novel and semantically-grounded technique* for proving genuine QBF lower bounds in P+$\forall$red, representing a significant forward step in the understanding of reasons for hardness in the proof complexity of quantified Boolean formulas. Our central result, the Size-Cost-Capacity Theorem, provides an *absolute lower bound on the number of universal reductions* for a QBF refutation – in *any* P+$\forall$red proof system – stated as the ratio of two natural measures: the *cost* of a QBF and the *capacity* of a proof. As such, we obtain superpolynomial proof-size lower bounds whenever cost is high and capacity is small.

To that end, we demonstrate that P+$\forall$red proofs have unit capacity when P is resolution or Cutting Planes, and that capacity is at most proof size when P is Polynomial Calculus with Resolution. We therefore obtain lower bounds in these three proof systems based solely on cost. This is a rather pleasant state of affairs, since we are able to apply our technique in three interesting cases simply by demonstating the exponential cost of a family of QBFs. Moerover, in doing so we obtain exponential lower bounds for QU-Res, CP+$\forall$red and PCR+$\forall$red simultaneously.

[3] If any clause contains only universal variables, then there is a constant-size refutation using only this clause.

For our first application, we exemplify our technique with a new family of hard QBFs called the *equality formulas*. We strongly suggest that these formulas, notable for their simplicity and conspicuous exponential cost, will henceforth occupy a prominent place in QBF proof complexity. As our principal application, we prove exponential lower bounds for a large class of randomly generated QBFs by demonstrating that they have high cost with high probability. This is the first time that genuine lower bounds have been shown *en masse* for randomly generated QBFs. As a third example, we show how our technique can be applied to give a simple proof of hardness for the well-known family of QBFs from [39] (cf.[9]).

In addition, we also determine exact conditions on P by which P+$\forall$red is properly defined and receptive to our method, by introducing the notion of a propositional *base system* – a line-based propositional proof system satisfying three natural conditions.

1.5 Organisation of the paper

We continue with the necessary background in Section 2, and provide the details of our P+$\forall$red framework in Section 3. Section 4 presents our lower bound technique, including definitions of cost and capacity, and the statement of our central result, the Size-Cost-Capacity Theorem. Applications of Size-Cost-Capacity, including the details on random formulas, are the subject of Section 5. This is followed in Section 6 by some discussion on the relation of our work to existing QBF techniques, and the merits and future perspectives of our contribution's conceptual innovations. We close the paper in Section 7 with some conclusions and open problems.

2 Preliminaries

2.1 Quantified Boolean formulas

A *conjunctive normal form* (CNF) formula is a conjunction of clauses, each of which is a disjunction of literals. We represent a CNF as a set of clauses, and a clause as a set of literals.

A *quantified Boolean formula* (QBF) in *closed prenex form* is typically denoted $\Phi := \mathcal{Q} \cdot \phi$. In the *quantifier prefix* $\mathcal{Q} := \mathcal{Q}_1 X_1 \cdots \mathcal{Q}_n X_n$, the X_i are pairwise-disjoint sets of Boolean variables (or *blocks*)[4] each of which is quantified either existentially or universally by the *associated quantifier* $\mathcal{Q}_i \in \{\exists, \forall\}$, and consecutive blocks are oppositely quantified. The *propositional part* ϕ is a propositional formula all of whose variables $\mathrm{vars}(\phi)$ are quantified in $\mathcal{Q}$.

By the variables of Φ we mean the set $\mathrm{vars}(\Phi) := \bigcup_{i=1}^{n} X_i$. The set of existential variables of Φ, denoted $\mathrm{vars}_\exists(\Phi)$, is the union of those X_i whose associated quantifier $\mathcal{Q}_i$ is $\exists$, and we define the universal variables of Φ similarly. The prefix $\mathcal{Q}$ imposes a linear order $<_\mathcal{Q}$ on the variables of Φ, such that $x_i <_\mathcal{Q} x_j$ holds whenever $x_i \in X_i$, $x_j \in X_j$ and $i < j$, in which case we say that x_i *is left of* x_j (x_j *is right of* x_i) with respect to $\mathcal{Q}$. We extend the linear order $<_\mathcal{Q}$ to sets of variables in the natural way.

A *literal* l is a Boolean variable x or its negation $\neg x$, and we write $\mathrm{var}(l) := x$. A *total assignment* τ to a set $\mathrm{vars}(\tau) = X$ of Boolean variables is a function $\tau : X \to \{0, 1\}$, typically represented as a set of literals in which the literal $\neg x$ (resp. x) represents the assignment $x \mapsto 0$ (resp. $x \mapsto 1$). The set of all total assignments to X is denoted $\langle X \rangle$. A *partial*

[4] Whereas a block $X = \{x_1, \ldots, x_m\}$ is a set, it is written explicitly in a prefix as a string of variables $x_1 \cdots x_m$.

assignment to X is a total assignment to a subset of X. The *projection* of τ to a set X' of Boolean variables is the assignment $\{l \in \tau : \mathrm{var}(l) \in X'\}$.

The *restriction* of Φ by an assignment τ is $\Phi[\tau] := \mathcal{Q}[\tau] \cdot \phi[\tau]$, where $\mathcal{Q}[\tau]$ is obtained from $\mathcal{Q}$ by removing each variable in $\mathrm{vars}(\tau)$ (and any redundant quantifiers), and $\phi[\tau]$ is the restriction of ϕ by τ. Restriction of propositional formulas is defined by the conventional inductive semantics of propositional logic; that is, $\phi[\tau]$ is obtained from ϕ by substituting each occurrence of a variable in $\mathrm{vars}(\tau)$ by its associated truth value, and simplifying the resulting formula in the usual way.

2.2 QBF semantics

Semantics are neatly described in terms of strategies in the two-player *evaluation game*. The game takes place over n rounds, during which the variables of a QBF $\Phi := \mathcal{Q} \cdot \phi$ are assigned strictly in the linear order of the prefix $\mathcal{Q} := \exists E_1 \forall U_1 \cdots \exists E_n \forall U_n$.[5] In the i^{th} round, the existential player selects an assignment α_i to E_i and the universal player responds with an assignment β_i to U_i. At the conclusion the players have constructed a total assignment $\tau := \bigcup_{i=1}^{n} (\alpha_i \cup \beta_i) \in \langle \mathrm{vars}(\Phi) \rangle$. The existential player wins iff $\phi[\tau] = \top$; the universal player wins iff $\phi[\tau] = \bot$.

A strategy for the universal player details exactly how she should respond to all possible moves of the existential player. Formally, a $\forall$-*strategy* for Φ is a function $S : \langle \mathrm{vars}_\exists(\Phi) \rangle \to \langle \mathrm{vars}_\forall(\Phi) \rangle$ that satisfies the following for each $\alpha, \alpha' \in \mathrm{dom}(S)$ and each $i \in [n]$: if α and α' agree on $E_1 \cup \cdots \cup E_i$, then $S(\alpha)$ and $S(\alpha')$ agree on $U_1 \cup \cdots \cup U_i$.[6] We say that S is *winning* iff $\phi[\alpha \cup S(\alpha)] = \bot$ for each $\alpha \in \mathrm{dom}(S)$.

▶ **Proposition 2.1** (folklore). *A QBF is false if and only if it has a winning $\forall$-strategy.*

2.3 QBF resolution

Resolution is a well-studied refutational proof system for propositional CNF formulas with a single inference rule: the *resolvent* $C_1 \cup C_2$ may be derived from clauses $C_1 \cup \{x\}$ and $C_2 \cup \{\neg x\}$. Resolution is *refutationally* sound and complete: that is, the empty clause can be derived from a CNF iff it is unsatisfiable. Resolution becomes implicationally complete with the addition of the weakening rule, which allows literals to be added to clauses arbitrarily.

QU-Resolution (QU-Res) [39, 33] is a resolution-based proof system for QBFs of the form $\mathcal{Q} \cdot \phi$, where ϕ is a CNF. The calculus supplements resolution with a *universal reduction rule* which allows (literals in) universal variables to be removed from a clause C provided that they are right of all existentials in C with respect to $\mathcal{Q}$. Tautological clauses are explicitly forbidden; for any variable x, one may not derive a clause containing both x and $\neg x$. The rules of QU-Res are given in Figure 2. Note that we choose to include weakening of clauses as a valid inference rule, to emphasize the implicational completeness of the underlying propositional system.

A QU-Res *derivation* of a clause C from Φ is a sequence $C_1, \ldots, C_m$ of clauses in which (a) each C_i is either introduced as an axiom (i.e. $C_i \in \phi$) or is derived from previous clauses in the sequence using resolution or universal reduction, and (b) the *conclusion* $C = C_m$ is the unique clause that is not an antecedent in the application of one of these inference rules. A *refutation* of Φ is a derivation of the empty clause from Φ.

---

[5] An arbitrary QBF can be written in this form by allowing E_1 and U_n to be empty.

[6] Two assignments agree on a set if and only if their projections to that set are identical.

Axiom:	$\dfrac{}{C}$	C is a clause in the matrix ϕ.
Weakening:	$\dfrac{C}{C \cup W}$	Each variable appearing in W is in $\mathrm{vars}(\Phi)$. The consequent $C \cup W$ is non-tautologous.
Resolution:	$\dfrac{C_1 \cup \{x\} \qquad C_2 \cup \{\neg x\}}{C_1 \cup C_2}$	The resolvent $C_1 \cup C_2$ is non-tautologous.
Universal reduction:	$\dfrac{C \cup U}{C}$	U contains only universal literals. Each variable in U is right of all existential variables in C, with respect to $\mathcal{Q}$.

Figure 2 The rules of QU-Resolution. The input QBF is $\Phi = \mathcal{Q} \cdot \phi$, where ϕ is a propositional CNF containing no tautologous clauses.

3 Our framework

3.1 A formal definition of P+∀red

We associate the basic concept of a *line-based propositional proof system* P with the following four features:

(a) A set of *lines* $\mathcal{L}_\mathsf{P}$, containing at least the two lines $\top$ and $\bot$ that represent trivial truth and trivial falsity, respectively.

(b) A set of *inference rules* $\mathcal{I}_\mathsf{P}$ and an *axiom function* that maps each propositional formula ϕ to a set of axioms $\mathcal{A}_\mathsf{P}(\phi) \subseteq \mathcal{L}_\mathsf{P}$. The axiom function should be polynomial-time computable, and the validity of inferences should be polynomial-time checkable.

(c) A *variables function* that maps each line $L \in \mathcal{L}_\mathsf{P}$ to a finite set of Boolean variables $\mathrm{vars}(L)$, satisfying $\mathrm{vars}(\top) = \mathrm{vars}(\bot) = \emptyset$. Additionally, $\mathrm{vars}(L) \subseteq \mathrm{vars}(\phi)$ for each line L in a P-derivation from ϕ.[7]

(d) A *restriction operator* (denoted by square brackets) that takes each line $L \in \mathcal{L}_\mathsf{P}$, under restriction by any partial assignment τ to $\mathrm{vars}(L)$, to a line $L[\tau] \in \mathcal{L}_\mathsf{P}$. If τ is a total assignment, then $L[\tau]$ is either $\top$ or $\bot$.

Universal reduction is a widely used rule of inference in QBF proof systems, by which universal variables may be assigned under certain conditions. More precisely, a line L may be restricted by an assignment to a set of universal variables U provided each $u \in U$ is right of each existential in $\mathrm{vars}(L)$, with respect to the prefix of the input QBF. We state the rule formally in Figure 3.

The primary purpose of universal reduction is to lift a line-based propositional proof system P to QBF, as in the following definition.

▶ **Definition 3.1** (P+∀red [8]). Let P be a line-based propositional proof system. Then P+∀red is the system consisting of the inference rules of P in addition to universal reduction, in which references to the input formula ϕ in the rules of P are interpreted as references to the propositional part of the input QBF $\mathcal{Q} \cdot \phi$.

[7] Note that this does not exclude extended Frege systems (EF), whose lines can be represented as Boolean circuits as in [38, p. 71].

$$\frac{L}{L[\beta]}$$

- β is a partial assignment to the universal variables of Φ.
- each universal in $\mathrm{vars}(\beta)$ is right of each existential in $\mathrm{vars}(L)$, with respect to $\mathcal{Q}$.

Figure 3 The universal reduction rule, where $\Phi = \mathcal{Q} \cdot \phi$ is the input QBF.

The above definition, however, does not gaurantee that P+∀red is sound and complete. To do that, we must work a little harder, and identify some further properties required of P.

Before proceeding, we extend our notation from P to P+∀red in the natural way, denoting the lines available in P+∀red (syntactically equivalent to the lines available in P) by $\mathcal{L}_{\mathsf{P}+\forall\mathsf{red}}$, and writing $\mathrm{vars}_\exists(L)$ and $\mathrm{vars}_\forall(L)$ for the subsets of $\mathrm{vars}(L)$ consisting of the variables quantified existentially and universally, with respect to the prefix of the input QBF. Also, we observe that Res + ∀red and QU-Res are (virtually) identical proof systems,[8] and we will henceforth use the latter term.

The size of a P+∀red refutation π, denoted $|\pi|$, is defined similarly as for the propositional system P. (For example, the size of a QU-Res refutation is the number of clauses appearing in it.) For formal definitions of the other propositional systems and their proof sizes, we refer the reader to the full paper.

3.2 Propositional base systems

We first introduce a useful object in our framework: for any line $L \in \mathcal{L}_\mathsf{P}$, an associated Boolean function B_L. Observe that the purpose of the restriction operator is to encompass the natural semantics of P – for that reason, we made the natural stipulation that restriction by a total assignment to the variables of a line yields either trivial truth or trivial falsity. We may therefore associate with any line $L \in \mathcal{L}_\mathsf{P}$ the Boolean function on $\mathrm{vars}(L)$ that computes the propositional models of L, with respect to the semantics of the restriction operator for P.

▶ **Definition 3.2** (associated Boolean function)**.** Let P be a line-based propositional proof system and let $L \in \mathcal{L}_\mathsf{P}$. The *associated Boolean function* for L is $B_L : \langle \mathrm{vars}(L) \rangle \to \{0,1\}$, defined by $B_L(\tau) = 1$ if $L[\tau] = \top$, and $B_L(\tau) = 0$ otherwise.

Beyond the established notion of 'line-based', we identify three natural conditions on P by which P+∀red is a bona fide, sound and complete QBF proof system. The first of these guarantees that the propositional models of the axioms are exactly those of the input formula, and the second guarantees soundness and completeness *in the classical sense of propositional logic*.[9] The third property ensures that the restriction operator behaves sensibly; that is, the propositional models of the restricted line are computed by the restriction of the associated Boolean function. We introduce the term *base system* for those possessing all three.

▶ **Definition 3.3** (base system)**.** A *base system* P is a line-based propositional proof system satisfying the following three properties:

[8] The only difference between them is that it is allowable to derive universal tautologies and trivial truth in Res + ∀red. Such inferences, however, are never useful.

[9] The (proof-complexity-theoretic) concepts of soundness and completeness for arbitrary proof systems in the sense of Cook and Reckhow are weaker than their counterparts in propositional logic.

- *Axiomatic equivalence.* For each propositional formula ϕ and each $\tau \in \langle \mathrm{vars}(\phi) \rangle$, $\phi[\tau] = \top$ iff each $A \in \mathcal{A}_{\mathsf{P}}(\phi)$ satisfies $A[\tau] = \top$;
- *Inferential equivalence.* For each set of lines $\mathcal{L} \subseteq \mathcal{L}_{\mathsf{P}}$ and each line $L \in \mathcal{L}_{\mathsf{P}}$, L can be derived from $\mathcal{L}$ iff $\mathcal{L}$ semantically entails L;
- *Restrictive closure.* For each $L \in \mathcal{L}_{\mathsf{P}}$ and each partial assignment τ to $\mathrm{vars}(L)$, the Boolean functions $B_{L[\tau]}$ and $B_L|_\tau$ are identical.

On account of the low-level generality, the following theorem requires a non-trivial proof.

▶ **Theorem 3.4.** *If* P *is a base system, then* P+∀red *is a sound and complete QBF proof system.*

Formalising the framework of base systems renders our technique applicable to the complete spectrum of P+∀red proof systems. All the concrete propositional calculi considered in this work (i.e. those appearing in Figure 1) are demonstrably base systems.

4 Genuine QBF lower bounds with Size-Cost-Capacity

Using an established approach (e.g. [8]), the soundness of P+∀red is proved by demonstrating that a winning strategy for the ∀-player can be extracted from a refutation. However, with careful construction and analysis of the strategy extraction algorithm, we are able to obtain a much more valuable result – an absolute lower bound on the number of universal reductions steps.

Given a P+∀red refutation π of a QBF Φ, *round-based strategy extraction* works by first restricting π according to the ∃-player's move, then collecting the response for the ∀-player from some line in π, and iterating until the evaluation game concludes. We therefore reason as follows: A lower bound on the total number of responses contributed by π, coupled with an upper bound on the number of responses contributed per line, yields a lower bound on the number of lines in the refutation. In light of this observation, we define the two measures called *cost* and *capacity*.

4.1 Defining cost

Given a countermodel S for a false QBF Φ, it is natural to ask how many responses are used for each universal block, since the breadth of responses seems to capture, in some sense at least, the 'size' or 'complexity' of the winning strategy. Let us denote the maximum number (over all universal blocks) of responses in a single block by $\mu(S)$. We contend that $\mu(S)$ is useful measure of a countermodel, with respect to strategy extraction in particular, and so we define the cost of Φ as the minimum $\mu(S)$ over all countermodels.

▶ **Definition 4.1** (cost)**.** Let $\Phi := \forall U_1 \exists E_1 \cdots \forall U_n \exists E_n \cdot \phi$ be a false QBF. Further, for each winning ∀-strategy S for Φ and each $i \in [n]$, let S_i be the function that maps each $\alpha \in \langle \mathrm{vars}_\exists(\Phi) \rangle$ to the projection of $S(\alpha)$ to U_i, and let $\mu(S) := \max\{|\mathrm{rng}(S_i)| : i \in [n]\}$. The *cost* of Φ is $\mathrm{cost}(\Phi) := \min\{\mu(S) : S \text{ is a winning } \forall\text{-strategy for } \Phi\}$.

It should be clear that any winning strategy contains at least $\mathrm{cost}(\Phi)$ responses to some universal block. With respect to strategy extraction, therefore, cost is a natural semantically-grounded measure that provides a lower bound on the total number of extracted responses.

4.2 Defining capacity

In order to define capacity, we first introduce the concept of a response map. Strictly speaking, given a line $L \in \mathcal{L}_{\mathsf{P}+\forall\mathsf{red}}$ and a total assignment α to the existential variables of L, a response map returns a total assignment to the universal variables that is guaranteed to falsify $L[\alpha]$, as long as such an assignment exists.

▶ **Definition 4.2** (response map)**.** Let P be a base system. A *response map* $\mathcal{R}$ for $\mathsf{P}+\forall\mathsf{red}$ is any function with domain $\{(L,\alpha) : L \in \mathcal{L}_{\mathsf{P}+\forall\mathsf{red}}, \alpha \in \langle \mathrm{vars}_\exists(L)\rangle\}$ that maps each (L,α) to some $\beta \in \langle \mathrm{vars}_\forall(L)\rangle$ such that the following holds: If $B_L|_\alpha$ is zero anywhere, then it is zero at β.

Response maps play a vital role in the machinery of strategy extraction in the general setting of $\mathsf{P}+\forall\mathsf{red}$; indeed, for our framework to take effect, it is crucial that strategy extraction can be defined *with respect to an arbitrary response map.*[10] The purpose of capacity, however, is only to provide an upper bound on the number of responses per line. To that end, we define the concept of a *response set* for a line $L \in \mathcal{L}_{\mathsf{P}+\forall\mathsf{red}}$, which is simply a valid set of responses for L according to some response map.

▶ **Definition 4.3** (response set)**.** Let P be a base system, let $\mathcal{R}$ be a response map for $\mathsf{P}+\forall\mathsf{red}$, and let $L \in \mathcal{L}_{\mathsf{P}+\forall\mathsf{red}}$. The set $\{\mathcal{R}(L,\alpha) : \alpha \in \langle \mathrm{vars}_\exists(L)\rangle\}$ is a *response set* for L.

Now, we observe that one may choose to select a response map minimising the size of the response sets for the lines of $\mathcal{L}_{\mathsf{P}+\forall\mathsf{red}}$; moreover, round-based strategy extraction returns a winning $\forall$-strategy regardless of the choice of response map. By selecting such a minimal response map $\mathcal{R}$, we will therefore limit the capacity for any line to contribute multiple responses to the extracted strategy. Thus we associate with each P derivation the maximum number of responses that can be extracted from a single line in that derivation, with respect to a minimal response map. This is the intuition behind capacity; it captures the best-case upper bound we can place on the number of responses contributed per line.

▶ **Definition 4.4** (capacity)**.** Let P be a base system, let $\pi = L_1, \ldots, L_m$ be a $\mathsf{P}+\forall\mathsf{red}$ derivation, and let $\mu(L_i) := \min\{|R| : R \text{ is a response set for } L_i\}$, for each $i \in [m]$. The *capacity* of π is $\mathrm{capacity}(\pi) := \max\{\mu(L_i) : i \in [m]\}$.

4.3 The Size-Cost-Capacity Theorem

Putting the two measures together, we obtain our main result, the *Size-Cost-Capacity Theorem.*

▶ **Theorem 4.5** (Size-Cost-Capacity Theorem)**.** *Let* P *be a base system, and let* π *be a* $\mathsf{P}+\forall\mathsf{red}$ *refutation of a QBF* Φ*. Then*

$$|\pi| \geq \frac{\mathrm{cost}(\Phi)}{\mathrm{capacity}(\pi)}.$$

We emphasize that Size-Cost-Capacity works by counting universal reduction steps, which illustrates that all results obtained by application of our technique are genuine QBF lower bounds in the aforementioned sense.

For the specific applications in this paper, our technique comprises three very useful corollaries of the Size-Cost-Capacity Theorem, obtained in combination with capacity upper

[10] For the details, we kindly refer the reader to the full paper [7].

bounds for specific systems. For example, we prove that all QU-Res and CP+∀red refutations have capacity equal to 1, and hence deduce that *cost alone* gives an absolute proof-size lower bound there.

▶ **Corollary 4.6.** *Let π be a* QU-Res *or* CP+∀red *refutation of a QBF Φ. Then $|\pi| \geq \mathrm{cost}(\Phi)$.*

The case for the QBF version of Polynomial Calculus with Resolution (PCR+∀red) is much more challenging, and requires some linear algebra, owing to the underlying algebraic composition of Polynomial Calculus. Interestingly, it turns out that the capacity of a refutation there is no greater than its size, thus proof size is at least the square root of cost.

▶ **Corollary 4.7.** *Let π be a* PCR+∀red *refutation of a QBF Φ. Then $|\pi| \geq \sqrt{\mathrm{cost}(\Phi)}$.*

Equipped with these results, showing that the cost of a QBF is superpolynomial yields immediate proof-size lower bounds for all three systems simultaneously.

5 Applications of Size-Cost-Capacity

5.1 The equality formulas: a new family of hard QBFs

As a first application of our lower-bound technique, we introduce an interesting new family of hard QBFs.

▶ **Definition 5.1** (equality formulas)**.** For $n \in \mathbb{N}$, the n^{th} *equality formula* is

$$EQ(n) := \exists x_1 \cdots x_n \forall u_1 \cdots u_n \exists t_1 \cdots t_n \cdot \left(\bigwedge_{i=1}^{n} (x_i \vee u_i \vee \neg t_i) \wedge (\neg x_i \vee \neg u_i \vee \neg t_i) \right) \wedge \left(\bigvee_{i=1}^{n} t_i \right).$$

The equality formulas are so called because the only winning strategy for the ∀-player in the evaluation game is as follows: play $u_i = x_i$ for each $i \in [n]$. Consequently the winning strategy is not only unique, it contains all 2^n assignments to the universal variables. These two properties in tandem imply that the equality formulas have exponential cost.

▶ **Proposition 5.2.** *For each $n \in \mathbb{N}$, $\mathrm{cost}(EQ(n)) = 2^n$.*

Applying Size-Cost-Capacity via Corollaries 4.6 and 4.7, we obtain exponential proof-size lower bounds in all three systems QU-Res, CP+∀red and PCR+∀red.

▶ **Theorem 5.3.** *The equality formulas require refutations of size $2^{\Omega(n)}$ in each of the systems* QU-Res*,* CP+∀red *and* PCR+∀red*.*

Whereas it is plausible that the equality formulas are the simplest to which our technique applies, they are without doubt the simplest known hard QBFs. When considering QBF proof complexity lower bounds, particularly in P+∀red systems, we must concern ourselves with formulas with at least a Σ_3 prefix, of which the equality formulas are one of the simplest examples. If a QBF has a Σ_2 prefix, then it is true if and only if the existential parts of the clauses can all be satisfied, i.e. it is equivalent to a SAT problem. Similarly, a refutation of a QBF with a Π_2 prefix consists of a refutation of a subset of the existential clauses corresponding to a particular assignment to the universal variables. A Π_3 formula can also be regarded as essentially a SAT problem using similar reductions as for both Σ_2 and Π_2, so Σ_3 is the smallest prefix where we can expect to find genuine QBF lower bounds.

Closer inspection reveals that this lower bound is of a very specific type – it is a genuine QBF lower bound (the formulas are not harbouring propositional hardness) that does not

derive from a circuit lower bound (the winning strategy is not hard to compute in an associated circuit class). In existing QBF literature, the only other example of such a family comes from the famous formulas of Kleine Büning et al. [39] (cf. Subsection 5.3). Those formulas are significantly more complex, and exhibit unbounded quantifier alternation compared to the (bounded) Σ_3 prefix of the equality formulas.

5.2 The first hard random QBFs

For the major application of our technique, we define a class of random QBFs and prove that, with high probability, they are hard in all three systems QU-Res, CP+$\forall$red and PCR+$\forall$red. We generate instances that combine the overall structure of the equality formulas with the literature's existing model of random QBFs [22].

▶ **Definition 5.4.** For each $1 \leq i \leq n$, let $C_i^1, \ldots, C_i^{cn}$ be distinct clauses picked uniformly at random from the set of clauses containing 1 literal from the set $X_i := \{x_i^1, \ldots, x_i^m\}$ and 2 literals from $Y_i := \{y_i^1, \ldots, y_i^n\}$. Define the randomly generated QBF $Q(n, m, c)$ as:

$$Q(n,m,c) := \exists Y_1 \ldots Y_n \forall X_1 \ldots X_n \exists t_1 \ldots t_n \cdot \bigwedge_{i=1}^{n} \bigwedge_{j=1}^{cn} \left(\neg t_i \vee C_i^j\right) \wedge \bigvee_{i=1}^{n} t_i.$$

The specification of how many existential and universal variables each clause should contain is a common and necessary restriction on random QBFs [22, 26]. This prevents the occurrence of a clause containing only universal variables – if such a clause exists, there is a constant size refutation of this clause alone in any P+$\forall$red system. The motivation behind the additional structure in the construction of $Q(n, m, c)$ is that its truth value is equivalent to the disjunction of its 'component parts'; that is $Q(n, m, c) \equiv \bigvee_{i=1}^{n} \Psi_i$, where $\Psi_i := \exists Y_i \forall X_i \cdot \bigwedge_{j=1}^{cn} C_i^j$ for each $i \in [n]$.

These Ψ_i are some of the simplest QBFs one can generate, so $Q(n, m, c)$ is a natural choice for random QBFs. Indeed, the model used to generate the clauses of Ψ_i is also used to generate random formulas for the evaluation of QBF solvers [46, 18].

Drawing on the existing literature [27, 23, 26], we show that suitable choices of the parameters m and c force each Ψ_i to be false with high probability. The individual Ψ_i are essentially equivalent to a random 2-SAT problem, and this step is just an application of results on the satisfiability of such instances.

Moreover, we also prove a cost lower bound. Perhaps surprisingly, this cost lower bound is constructed by applying results on the unsatisfiability of random 2-SAT instances [27] and the truth of random (1,2)-QCNFs [26]. These results both concern only the truth value of the corresponding formulas, and taken individually seem unrelated to cost. However, by carefully choosing the number of clauses so as to allow the application of both results, we can construct a cost lower bound using the following argument.

The Ψ_i are false with high probability, but rearranging the quantifiers to $\forall X_i \exists Y_i \cdot \bigwedge_{j=1}^{cn} C_i^j$ gives a QBF which is true with probability $1 - o(1)$. In other words, with high probability, the universal response in Ψ_i must depend on the existential assignment. That is, it must change depending on the existential assignment, and so with probability $1 - o(1)$, linearly many of the Ψ_i require at least two distinct responses in any winning strategy. By refining our choice of m slightly, this allows us to conclude that $Q(n, m, c)$, with high probability, is a false QBF with large cost.

▶ **Proposition 5.5.** *Let $1 < c < 2$ be a constant, and let $m \leq (1 - \epsilon)\log_2(n)$ for some constant $\epsilon > 0$. With probability $1 - o(1)$, $Q(n, m, c)$ is false and $\mathrm{cost}(Q(n, m, c)) = 2^{\Omega(n^\epsilon)}$.*

Invoking Size-Cost-Capacity yields immediate hardness results. The following theorem constitutes the first proof-size lower bounds for randomly generated formulas in the QBF proof complexity literature. We emphasize that these are genuine QBF lower bounds in the aforementioned sense; they are not merely hard random CNFs lifted to QBF. As for any application of our technique, the refutations are large precisely because they require many universal reduction steps.

▶ **Theorem 5.6.** *Let $1 < c < 2$ be a constant, and let $m \leq (1-\epsilon)\log_2(n)$ for some constant $\epsilon > 0$. With probability $1 - o(1)$, $Q(n, m, c)$ is false, and any* QU-Res, CP+∀red *or* PCR+∀red *refutation of $Q(n, m, c)$ requires size $2^{\Omega(n^\epsilon)}$.*

5.3 New proofs of known lower bounds

Our third and final application uses Size-Cost-Capacity to provide a new proof of the hardness of the prominent QBFs of Kleine Büning, Karpinski and Flögel [39]. We consider a common modification of the formulas, denoted by $\lambda(n)$, in which each universal variable is 'doubled'. This modification is known to lift lower bounds from Q-Res to QU-Res [2], where we can apply Size-Cost-Capacity.

By rearranging the quantifier prefix to quantify all the additional universal variables in the penultimate quantifier block, we obtain a cost lower bound for this weaker formula, and so prove the following result.

▶ **Corollary 5.7.** *Any* QU-Res, CP+∀red *or* PCR+∀red *proof of $\lambda(n)$ requires size $2^{\Omega(n)}$.*

As QU-Res lower bounds on these modified formulas are shown to be equivalent to Q-Res lower bounds on the original formulas, our technique even proves the original lower bounds from [39] (cf. also [10]), and provides some insight as to the source of hardness.

6 Discussion

6.1 Relation to previous work

It is fair to say that there is a scarcity of general methods for showing genuine lower bounds in systems like P+∀red. In contrast, a number of techniques for propositional calculi have emerged from the intense study of resolution [19, 49].

Researchers have of course attempted to lift these techniques to quantified logic, but with mixed success. The seminal size-width relations for resolution [5], which describe proof size in terms of proof width, are rendered ineffectual by universal quantification [11]. The prover-delayer techniques of [14, 45] have been successfully lifted to QBF, but only apply to the weaker tree-like systems [13], whereas solving techniques such as QCDCL are based on the stronger DAG-like versions. Feasible interpolation [41] is an established propositional technique that has been successfully adapted [12], but it is applicable only to a small class of hand-crafted QBFs of a rather specific syntactic form.

Strategy extraction for QBF lower bounds has been explored previously by exploiting connections to circuit complexity [10, 8, 16]. In particular, [8] established tight relations between circuit and proof complexity, lifting even strong circuit lower bounds for $\mathbf{AC^0}[p]$ circuits [47, 51] to QBF lower bounds for $\mathbf{AC^0}[p]$-Frege+∀red [8], unparalleled in the propositional domain. In fact, for strong proof systems such as Frege+∀red, this strategy extraction technique is sufficient to prove any genuine QBF lower bound, in the sense that any superpolynomial lower bound for Frege+∀red arises either due to a lower bound for Frege, or due to a lower bound for Boolean circuits [16]. However, for weaker systems such

as QU-Res, this does not hold; there exist lower bounds which are neither propositional nor circuit lower bounds [15]. The underlying reasons for such hardness results are at present not well understood. A characterisation of such lower bounds, and the proposal of associated lower-bound techniques, would be an important development for QBF proof complexity.

The major drawback of the existing approach of [10, 8, 16], of course, is the rarity of superpolynomial lower bounds from circuit complexity [55], especially for larger circuit classes to which the stronger QBF proof systems connect. With Size-Cost-Capacity we employ a much different approach to strategy extraction. Our technique is motivated by semantics and *does not interface with circuit complexity whatsoever*. Instead, lower bounds are determined directly from the semantic properties of the instance, and consequently we make advances out of the reach of previous techniques.

6.2 Innovations and future perspectives

Our main conceptual innovation is the introduction of *Size-Cost-Capacity*, a semantically-grounded general technique for proving genuine QBF lower bounds.

In this paper, we focus the technique on the P+∀red family of QBF calculi, and prove the first known lower bounds for randomly generated QBFs. The primary appeal of the technique is its semantic nature. We believe that lower bounds based on semantic properties of instances, as opposed to syntactic properties of proofs, work to further our understanding of the hardness phenomenon across the wider range of QBF proof systems. We strongly suggest that Size-Cost-Capacity is applicable beyond P+∀red, and future work will likely establish the hardness of random QBFs in even stronger QBF systems (for example in the expansion based calculus IR-calc [9]).

Size-Cost-Capacity also opens new research avenues concerning the reasons for QBF hardness – a topic that is currently insufficiently understood. Recall that in strong proof systems such as Frege+∀red, superpolynomial proof size lower bounds can be completely characterised: they are either a propositional lower bound or a circuit lower bound [16]. All the QBF families that we consider have no underlying propositional hardness, and winning ∀-strategies can be computed by small circuits, even in very restricted circuit classes. As such, all these QBFs are easy for Frege+∀red.

However, for weaker proof systems, such as QU-Res, CP+∀red and PCR+∀red, propositional hardness and circuit lower bounds alone are not the complete picture. In particular, the lower bounds we show using Size-Cost-Capacity do not fit into either class. That our technique relies on capacity upper bounds which do not hold for strong proof systems leads us to suggest that we have identified a new reason for hardness in those proof systems where the above characterisation does not hold. As such, our work opens the door for a better understanding, and makes steps towards the complete characterisations of reasons for hardness that are currently lacking in the literature.

7 Conclusions

By formalising the conditions on P in the construction of P+∀red, we have developed a new technique for proving QBF lower bounds in P+∀red. The technique depends only on the two natural concepts of the cost of a QBF and the capacity of a proof. Determining the capacity of several well-studied proof systems allowed us to present lower bounds based on cost alone. We have also demonstrated that this technique is not restricted to a few carefully constructed QBFs, but is in fact applicable to a large class of randomly generated formulas, providing the first such lower bound for random QBFs.

References

1 Michael Alekhnovich and Alexander A. Razborov. Lower bounds for polynomial calculus: Non-binomial case. In *Symposium on Foundations of Computer Science (FOCS)*, pages 190–199. IEEE Computer Society, 2001.

2 Valeriy Balabanov, Magdalena Widl, and Jie-Hong R. Jiang. QBF resolution systems and their proof complexities. In Carsten Sinz and Uwe Egly, editors, *International Conference on Theory and Applications of Satisfiability Testing (SAT)*, volume 8561 of *Lecture Notes in Computer Science*, pages 154–169. Springer, 2014.

3 Paul Beame and Toniann Pitassi. Simplified and improved resolution lower bounds. In *Symposium on Foundations of Computer Science (FOCS)*, pages 274–282. IEEE Computer Society, 1996.

4 Eli Ben-Sasson and Russell Impagliazzo. Random CNFs are hard for the polynomial calculus. *Computational Complexity*, 19(4):501–519, 2010.

5 Eli Ben-Sasson and Avi Wigderson. Short proofs are narrow - resolution made simple. *Journal of the ACM*, 48(2):149–169, 2001.

6 Marco Benedetti and Hratch Mangassarian. QBF-based formal verification: Experience and perspectives. *Journal on Satisfiability, Boolean Modeling and Computation (JSAT)*, 5(1-4):133–191, 2008.

7 Olaf Beyersdorff, Joshua Blinkhorn, and Luke Hinde. Size, cost, and capacity: A semantic technique for hard random QBFs. *Electronic Colloquium on Computational Complexity (ECCC)*, 24:35, 2017.

8 Olaf Beyersdorff, Ilario Bonacina, and Leroy Chew. Lower bounds: From circuits to QBF proof systems. In Madhu Sudan, editor, *ACM Conference on Innovations in Theoretical Computer Science (ITCS)*, pages 249–260. ACM, 2016.

9 Olaf Beyersdorff, Leroy Chew, and Mikoláš Janota. On unification of QBF resolution-based calculi. In Erzsébet Csuhaj-Varjú, Martin Dietzfelbinger, and Zoltán Ésik, editors, *International Symposium on Mathematical Foundations of Computer Science (MFCS)*, volume 8635 of *Lecture Notes in Computer Science*, pages 81–93. Springer, 2014.

10 Olaf Beyersdorff, Leroy Chew, and Mikoláš Janota. Proof complexity of resolution-based QBF calculi. In Ernst W. Mayr and Nicolas Ollinger, editors, *International Symposium on Theoretical Aspects of Computer Science (STACS)*, volume 30 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 76–89. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2015.

11 Olaf Beyersdorff, Leroy Chew, Meena Mahajan, and Anil Shukla. Are short proofs narrow? QBF resolution is not simple. In *Symposium on Theoretical Aspects of Computer Science (STACS)*, pages 15:1–15:14, 2016.

12 Olaf Beyersdorff, Leroy Chew, Meena Mahajan, and Anil Shukla. Feasible interpolation for QBF resolution calculi. *Logical Methods in Computer Science*, 13, 2017.

13 Olaf Beyersdorff, Leroy Chew, and Karteek Sreenivasaiah. A game characterisation of tree-like Q-resolution size. *Journal of Computer and System Sciences*, 2017. in press.

14 Olaf Beyersdorff, Nicola Galesi, and Massimo Lauria. A characterization of tree-like resolution size. *Information Processing Letters*, 113(18):666–671, 2013.

15 Olaf Beyersdorff, Luke Hinde, and Ján Pich. Reasons for hardness in QBF proof systems. In *Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS)*, 2017.

16 Olaf Beyersdorff and Ján Pich. Understanding Gentzen and Frege systems for QBF. In Martin Grohe, Eric Koskinen, and Natarajan Shankar, editors, *Symposium on Logic in Computer Science (LICS)*, pages 146–155. ACM, 2016.

17 Roderick Bloem, Robert Könighofer, and Martina Seidl. SAT-based synthesis methods for safety specs. In Kenneth L. McMillan and Xavier Rival, editors, *International Conference*

on Verification, Model Checking, and Abstract Interpretation (VMCAI), volume 8318 of *Lecture Notes in Computer Science*, pages 1–20. Springer, 2014.

18 Robert Brummayer, Florian Lonsing, and Armin Biere. Automated testing and debugging of SAT and QBF solvers. In Ofer Strichman and Stefan Szeider, editors, *International Conference on Theory and Practice of Satisfiability Testing (SAT)*, volume 6175 of *Lecture Notes in Computer Science*, pages 44–57. Springer, 2010.

19 Samuel R. Buss. Towards NP-P via proof complexity and search. *Ann. Pure Appl. Logic*, 163(7):906–917, 2012.

20 Michael Cashmore, Maria Fox, and Enrico Giunchiglia. Partially grounded planning as quantified Boolean formula. In Daniel Borrajo, Subbarao Kambhampati, Angelo Oddi, and Simone Fratini, editors, *International Conference on Automated Planning and Scheduling (ICAPS)*. AAAI, 2013.

21 Hubie Chen. Proof complexity modulo the polynomial hierarchy: Understanding alternation as a source of hardness. *ACM Transactions on Computation Theory*, 9(3):15:1–15:20, 2017.

22 Hubie Chen and Yannet Interian. A model for generating random quantified Boolean formulas. In Leslie Pack Kaelbling and Alessandro Saffiotti, editors, *International Joint Conference on Artificial Intelligence (IJCAI)*, pages 66–71. Professional Book Center, 2005.

23 Vasek Chvátal and Bruce A. Reed. Mick gets some (the odds are on his side). In *Symposium on Foundations of Computer Science (FOCS)*, pages 620–627. IEEE Computer Society, 1992.

24 Stephen A. Cook and Phuong Nguyen. *Logical Foundations of Proof Complexity*. Cambridge University Press, Cambridge, 2010.

25 Stephen A. Cook and Robert A. Reckhow. The relative efficiency of propositional proof systems. *Journal of Symbolic Logic*, 44(1):36–50, 1979.

26 Nadia Creignou, Hervé Daudé, Uwe Egly, and Raphaël Rossignol. Exact location of the phase transition for random (1, 2)-QSAT. *RAIRO - Theoretical Informatics and Applications*, 49(1):23–45, 2015.

27 Wenceslas Fernandez de la Vega. Random 2-SAT: results and problems. *Theoretical Computer Science*, 265(1-2):131–146, 2001.

28 Nachum Dershowitz, Ziyad Hanna, and Jacob Katz. Space-efficient bounded model checking. In Fahiem Bacchus and Toby Walsh, editors, *International Conference on Theory and Applications of Satisfiability Testing (SAT)*, volume 3569 of *Lecture Notes in Computer Science*, pages 502–518. Springer, 2005.

29 Josep Díaz, Lefteris M. Kirousis, Dieter Mitsche, and Xavier Pérez-Giménez. A new upper bound for 3-SAT. In Ramesh Hariharan, Madhavan Mukund, and V. Vinay, editors, *Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS)*, volume 2 of *LIPIcs*, pages 163–174. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2008.

30 Uwe Egly, Martin Kronegger, Florian Lonsing, and Andreas Pfandler. Conformant planning as a case study of incremental QBF solving. *Annals of Mathematics and Artificial Intelligence*, 80(1):21–45, 2017.

31 Noah Fleming, Denis Pankratov, Toniann Pitassi, and Robert Robere. Random CNFs are hard for cutting planes. *Computing Research Repository*, abs/1703.02469, 2017.

32 John Franco and Marvin C. Paull. Probabilistic analysis of the Davis Putnam procedure for solving the satisfiability problem. *Discrete Applied Mathematics*, 5(1):77–87, 1983.

33 Allen Van Gelder. Contributions to the theory of practical quantified Boolean formula solving. In Michela Milano, editor, *International Conference on Principles and Practice of Constraint Programming (CP)*, volume 7514 of *Lecture Notes in Computer Science*, pages 647–663. Springer, 2012.

34 Enrico Giunchiglia, Paolo Marin, and Massimo Narizzano. Reasoning with quantified Boolean formulas. In Armin Biere, Marijn Heule, Hans van Maaren, and Toby Walsh, editors, *Handbook of Satisfiability*, volume 185 of *Frontiers in Artificial Intelligence and Applications*, pages 761–780. IOS Press, 2009.

35 Alexandra Goultiaeva, Allen Van Gelder, and Fahiem Bacchus. A uniform approach for generating proofs and strategies for both true and false QBF formulas. In Toby Walsh, editor, *International Joint Conference on Artificial Intelligence (IJCAI)*, pages 546–553. IJCAI/AAAI, 2011.

36 Pavel Hrubes and Pavel Pudlák. Random formulas, monotone circuits, and interpolation. *Electronic Colloquium on Computational Complexity*, 24:42, 2017.

37 Mikoláš Janota and João Marques-Silva. Expansion-based QBF solving versus Q-resolution. *Theoretical Computer Science*, 577:25–42, 2015.

38 Emil Jeřábek. *Weak pigeonhole principle, and randomized computation*. PhD thesis, Faculty of Mathematics and Physics, Charles University, Prague, 2005.

39 Hans Kleine Büning, Marek Karpinski, and Andreas Flögel. Resolution for quantified Boolean formulas. *Information and Computation*, 117(1):12–18, 1995.

40 Roman Kontchakov, Luca Pulina, Ulrike Sattler, Thomas Schneider, Petra Selmer, Frank Wolter, and Michael Zakharyaschev. Minimal module extraction from DL-lite ontologies using QBF solvers. In Craig Boutilier, editor, *International Joint Conference on Artificial Intelligence (IJCAI)*, pages 836–841. AAAI Press, 2009.

41 Jan Krajíček. Interpolation theorems, lower bounds for proof systems, and independence results for bounded arithmetic. *Journal of Symbolic Logic*, 62(2):457–486, 1997.

42 Jan Krajíček. *Bounded Arithmetic, Propositional Logic, and Complexity Theory*, volume 60 of *Encyclopedia of Mathematics and Its Applications*. Cambridge University Press, Cambridge, 1995.

43 Sharad Malik and Lintao Zhang. Boolean satisfiability from theoretical hardness to practical success. *Communications of the ACM*, 52(8):76–82, 2009.

44 Kuldeep S. Meel, Moshe Y. Vardi, Supratik Chakraborty, Daniel J. Fremont, Sanjit A. Seshia, Dror Fried, Alexander Ivrii, and Sharad Malik. Constrained sampling and counting: Universal hashing meets SAT solving. In Adnan Darwiche, editor, *Beyond NP*, volume WS-16-05 of *AAAI Workshops*. AAAI Press, 2016.

45 Pavel Pudlák and Russell Impagliazzo. A lower bound for DLL algorithms for k-SAT (preliminary version). In David B. Shmoys, editor, *Symposium on Discrete Algorithms*, pages 128–136. ACM/SIAM, 2000.

46 Luca Pulina. The ninth QBF solvers evaluation - preliminary report. In Florian Lonsing and Martina Seidl, editors, *International Workshop on Quantified Boolean Formulas (QBF)*, volume 1719 of *CEUR Workshop Proceedings*, pages 1–13. CEUR-WS.org, 2016.

47 Alexander A. Razborov. Lower bounds for the size of circuits of bounded depth with basis $\{\wedge, \oplus\}$. *Mathematical Notes*, 41(4):333–338, 1987.

48 Jussi Rintanen. Asymptotically optimal encodings of conformant planning in QBF. In *National Conference on Artificial Intelligence (AAAI)*, pages 1045–1050. AAAI Press, 2007.

49 Nathan Segerlind. The complexity of propositional proofs. *Bulletin of Symbolic Logic*, 13(4):417–481, 2007.

50 João P. Marques Silva and Karem A. Sakallah. GRASP - a new search algorithm for satisfiability. In Rob A. Rutenbar and Ralph H. J. M. Otten, editors, *International Conference on Computer-Aided Design (ICCAD)*, pages 220–227. IEEE Computer Society / ACM, 1996.

51 R. Smolensky. Algebraic methods in the theory of lower bounds for Boolean circuit complexity. In Alfred V. Aho, editor, *ACM Symposium on Theory of Computing (STOC)*, pages 77–82. ACM, 1987.

52 Stefan Staber and Roderick Bloem. Fault localization and correction with QBF. In João Marques-Silva and Karem A. Sakallah, editors, *International Conference on Theory and Applications of Satisfiability Testing (SAT)*, volume 4501 of *Lecture Notes in Computer Science*, pages 355–368. Springer, 2007.

53 Larry J. Stockmeyer and Albert R. Meyer. Word problems requiring exponential time: Preliminary report. In Alfred V. Aho, Allan Borodin, Robert L. Constable, Robert W. Floyd, Michael A. Harrison, Richard M. Karp, and H. Raymond Strong, editors, *ACM Symposium on Theory of Computing (STOC)*, pages 1–9. ACM, 1973.

54 Moshe Y. Vardi. Boolean satisfiability: Theory and engineering. *Communications of the ACM*, 57(3):5, 2014.

55 Heribert Vollmer. *Introduction to Circuit Complexity - A Uniform Approach.* Texts in Theoretical Computer Science. Springer, 1999.

Stabbing Planes*†

Paul Beame[1], Noah Fleming[2], Russell Impagliazzo[3], Antonina Kolokolova[4], Denis Pankratov[5], Toniann Pitassi[6], and Robert Robere[7]

1 **University of Washington, Seattle, USA**
beame@cs.washington.edu
2 **University of Toronto, Toronto, Canada**
noahfleming@cs.toronto.edu
3 **University of California, San Diego, USA**
russell@cs.ucsd.edu
4 **Memorial University of Newfoundland, St. John's, Canada**
kol@mun.ca
5 **University of Toronto, Toronto, Canada**
denisp@cs.toronto.edu
6 **University of Toronto and Institute for Advanced Study, Toronto, Canada**
toni@cs.toronto.edu
7 **University of Toronto, Toronto, Canada**
robere@cs.toronto.edu

Abstract

We introduce and develop a new semi-algebraic proof system, called Stabbing Planes that is in the style of DPLL-based modern SAT solvers. As with DPLL, there is only one rule: the current polytope can be subdivided by branching on an inequality and its "integer negation." That is, we can (nondeterministically choose) a hyperplane $ax \geq b$ with integer coefficients, which partitions the polytope into three pieces: the points in the polytope satisfying $ax \geq b$, the points satisfying $ax \leq b-1$, and the middle slab $b-1 < ax < b$. Since the middle slab contains no integer points it can be safely discarded, and the algorithm proceeds recursively on the other two branches. Each path terminates when the current polytope is empty, which is polynomial-time checkable. Among our results, we show somewhat surprisingly that Stabbing Planes can efficiently simulate Cutting Planes, and moreover, is strictly stronger than Cutting Planes under a reasonable conjecture. We prove linear lower bounds on the *rank* of Stabbing Planes refutations, by adapting a lifting argument in communication complexity.

1998 ACM Subject Classification F.0 General

Keywords and phrases communication complexity, cutting planes, proof complexity

Digital Object Identifier 10.4230/LIPIcs.ITCS.2018.10

1 Introduction

While defined in terms of non-deterministic algorithms for the Tautology problem, proof complexity has also provided indispensable tools for understanding deterministic algorithms for search problems, and in particular, for Satisfiability algorithms. Many algorithms for

* This work was partially supported by NSERC and by NSF awards CCF-1524246, CCF-1412958, CCF-1213151. Part of the work was done while at Simons institute.
† Full version of this paper is available at[1], https://arxiv.org/abs/1710.03219

search can be classified according to the types of reasoning they implicitly use for case-analysis and pruning unpromising branches. Particular families of search algorithms can be characterized by *formal proof systems*; the size of proofs in these formal proof system, the time of the non-deterministic algorithm, captures the time taken on the instance by an *ideal implementation* of the search algorithm. This allows us to factor understanding the power of search algorithms of a given type into two questions:

1. How powerful is the proof system? For which kinds of input are there small proofs?
2. How close can actual implementations of the search method come to the ideal non-deterministic algorithm?

As an illustrative example, let us recall the *DPLL* algorithm [10, 9], which is one of the simplest algorithms for SAT and forms the basis of modern conflict-driven clause learning SAT solvers. Let $\mathcal{F} = C_1 \wedge C_2 \wedge \cdots \wedge C_m$ be a CNF formula over variables $x_1, x_2, \ldots, x_n$. A DPLL search tree for solving the SAT problem for $\mathcal{F}$ is constructed as follows. Begin by choosing a variable x_i (non-deterministically, or via some heuristic), and then recurse on the formulas $\mathcal{F} \restriction x_i = 0$, $\mathcal{F} \restriction x_i = 1$. If at any point we have found a satisfying assignment, the algorithm outputs *SAT*. Otherwise, if we have falsified every literal in some clause C, then we record the clause and halt the recursion. If every recursive branch ends up being labelled with a clause and a falsifying assignment, then the original formula $\mathcal{F}$ is unsatisfiable and one can take the tree as a proof of this fact; in fact, such a DPLL tree is equivalent to a *tree-like* Resolution refutation of the formula $\mathcal{F}$.

Modern SAT solvers still have a DPLL algorithm at the core (now with a highly tuned *branching heuristic* that chooses the "right" order for variables and assignments to recurse on in the search tree), but extends the basic recipe in two ways: smart handling of *unit clauses* (if $\mathcal{F}$ contains a clause with a single variable x under the current partial assignment, x can be immediately set so that the clause is satisfied), and *clause learning* to speed up search: if a partial assignment ρ falsifies a clause, then the algorithm derives a new clause C_ρ by a resolution proof that "explains" this conflict, and adds the new clause to the formula $\mathcal{F}$.

It is the synergy between these three mechanisms – branching heuristics, unit propagation, and clause learning – that results in the outstanding performance of modern SAT solvers. In other words, while these algorithms are all formalizable in the same simple proof system, the sophistication of modern SAT-solvers comes from the attempt to algorithmically find small proofs when they exist. In many ways, the simplicity of the proof system enables this sophistication in proof-search methods.

In this work, we introduce a natural generalization of the DPLL-style branching algorithm to reasoning over integer-linear inequalities, formalized as a new semi-algebraic proof system that we call the *Stabbing Planes* (SP) system. We will give a more detailed description later, but intuitively, Stabbing Planes has the same branching structure as DPLL, but generalizes branching on single variables to branching on linear inequalities over the variables. We feel the closeness to DPLL makes Stabbing Planes a better starting point for understanding search algorithms based on linear inequalities, as in integer linear programming (ILP) based solvers, than established proof systems such as Cutting Planes.

We compare the power of Stabbing Planes proofs to these other proof systems. Recall that Cutting Planes (CP) is a proof system for reasoning over linear inequalities using linear combination and division with rounding rules, and Krajícek's system R(CP) combines resolution and CP rules. We show that tree-like R(CP) is polynomially equivalent to Stabbing Planes (Theorem 9). However, the new formulation as Stabbing Planes proofs both gives greater motivation to studying R(CP) and greatly clarifies the power of this proof system. Our main results about this system are:

1. Stabbing Planes has quasi-polynomial size and poly-log rank proofs of any tautology provable using linear algebra over a constant modulus. In particular, this is true for the Tseitin graph tautologies, that are very frequently used to prove lower bounds for other proof systems. (Theorem 6)
2. Stabbing Planes can simulate tree-like Cutting Planes proofs with only constant factor increases in size and rank (Theorem 11), and general Cutting Planes proofs with a polynomial increase in size (Theorem 12)
3. Lower bounds on real communication protocols imply rank lower bounds for Stabbing Planes proofs (Lemma 17)
4. Stabbing Planes proofs cannot be balanced (Theorem 21).

Together, these show that Stabbing Planes is at least as strong as established proof systems using inequalities, and possibly much stronger. So the proof system combines strength as a proof system with a simple branching structure that raises the possibility of elegant algorithms based on this proof system, in particular for pseudoBoolean solvers.

We now give a more precise description of the proof system. Let us formalize the system in stages. First, observe that the setting is quite different: we are given a system $A_1 x \geq b_1, A_2 \cdot x \geq b_2, \ldots, A_m \cdot x \geq b_m$ of integer-linear inequalities over real-valued variables $x_1, x_2, \ldots, x_n$ (for simplicity we will always assume that the inequalities $0 \leq x_i \leq 1$ are present for each variable x_i), and we seek to prove that no $\{0,1\}$-solution exists. The basic DPLL algorithm works in this setting with little modification: one can still query variables and assign them to $\{0,1\}$ values; now we label leaves of the search tree with any inequality $a_i \cdot x \geq b_i$ in the system that is falsified by the sequence of assignments made on the path from the root to the leaf. If every leaf ends up being labelled with a falsified inequality, then the tree certifies that the system of inequalities has no $\{0,1\}$-solutions.

However, with the expanded domain we can consider the DPLL tree *geometrically*. To be more specific, imagine replacing each $\{0,1\}$ query to a variable x_i in the decision tree with two "inequality queries" $x_i \leq 0$ and $x_i \geq 1$. Each node u in the tree after this replacement is now naturally associated with a polyhedral set $\mathcal{P}_u$ of points satisfying each of the the input inequalities *and* each of the inequalities on the path to this node. Since we began with a DPLL refutation, it is clear that for any leaf ℓ the polyhedral set $\mathcal{P}_\ell$ associated with the leaf is empty, as any $\{0,1\}$ solution would have survived each of the inequalities queried on some path in the tree and thus would exist in one of the polyhedral sets at the leaves.

The stabbing planes system is then the natural generalization of the previous object: an SP refutation consists of a generalized DPLL refutation where each node is labelled with an *arbitrary* integral linear inequality $Ax \geq b$ (that is, the vector A and the parameter b are both integral), and the outgoing edges are labelled with the inequalities $Ax \geq b$ and $Ax \leq b - 1$. Clearly, any integral vector $x \in \mathbb{Z}^n$ will satisfy at least one of the inequalities labelling the outgoing edges, and so if the polyhedral set at each leaf (again, obtained by intersecting the original system with the inequalities on the path to the leaf) is empty then we have certified that the original system of inequalities has no integral solutions. (See Figure 1 for a simple example.) The main innovation of Stabbing Planes is its simplicity: refutations are simply decision trees that query linear inequalities. Note that the more obvious extension of DPLL to linear inequalities would branch on $Ax \geq b$ and its *actual* negation, $Ax < b$. However with this branching rule, we would have to add additional rules in order to have completeness. By branching on an inequality and its "integer negation", we are able to get by with just one rule analogous to the resolution rule in DPLL.

From the perspective of SAT solving, even though tree-like Resolution and the search for satisfying assignments encapsulated by DPLL are equivalent, it is the search point of view of DPLL that has led to the major advances in SAT algorithms now found in modern

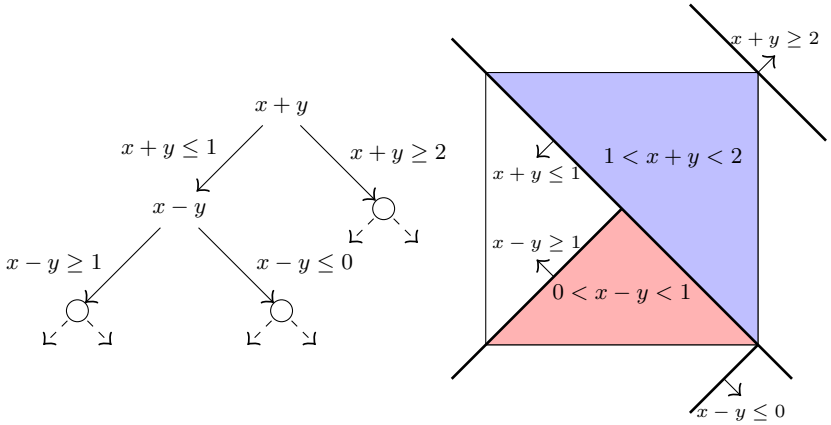


Figure 1 A partial SP refutation and the result on the unit square. The shaded areas are "removed" from the polytope, and we recurse on each side.

conflict-directed clause learning (CDCL) SAT solvers. A natural hypothesis is that it is much easier to invent useful heuristics in the language of query-based algorithms, as opposed to algorithms based on the resolution rule. Stabbing Planes offers a similar benefit with respect to reasoning about inequalities.

With the exception of mixed integer programming (MIP) solvers (such as CPLEX [19]), current solvers that, like Stabbing Planes, manipulate integer linear inequalities over Boolean variables are generally built on the same backtracking-style infrastructure as DPLL and CDCL SAT solvers but maintaining information as integer linear inequalities as opposed to clausal forms. The solvers are known as *pseudoBoolean* solvers and have been the subject of considerable effort and development.

PseudoBoolean solvers work very well at handling the kinds of symmetric counting problems associated with, for example, the pigeonhole principle (PHP), which is known to be hard for CDCL SAT solvers, as well as other problems where the input constraints are much more succinctly and naturally expressed in inequality rather than clausal form. Innovations in pseudoBoolean solvers include use of normal forms for expressing constraints, techniques to generalize fast unit propogation and watched literals from DPLL to the analogue for integer linear inequalities, as well as methods to learn from conflicts and simplify learned constraints when integer coefficients from derived inequalities get too large [29, 5]. Despite all of this, even for the best pseudoBoolean solvers, the benefits of expressibility are usually not enough compensation for the added costs of manipulating and deriving new inequalities and they outperform CDCL solvers only in very limited cases in practice [5].

A key limitation of these pseudoBoolean solvers is the fact that all branching is based on assigning values to individual variables; i.e., dividing the problem by slabs parallel to one of the coordinate axes. Stabbing Planes eliminates this constraint on the search and allows one to apply a divide and conquer search based on arbitrary integer linear constraints that are not necessarily aligned with one of these coordinate axes. This opens up the space of algorithmic ideas considerably and should allow future pseudoBoolean solvers to take fuller advantage of the expressibility of integer linear constraints. For example, a Stabbing Planes search could choose to branch on a linear inequality that is derived from the geometric properties of the rational hull of the current constraints by, say, splitting the volume, or by doing a balanced split at a polytope vertex, since properties of the rational hull can be determined efficiently. Such operations could be done in conjunction with solvers such as CPLEX to obtain the best of both kinds of approaches.

Beyond the prospect of Stabbing Planes yielding improved backtracking search for pseudoBoolean solvers, Stabbing Planes should allow the same kind of learning of inequalities from conflicts that is being done in existing pseudoBoolean solvers, and hence get the benefits of both. In this work we do not focus on the theoretical benefits of learning from conflicts because we already can show considerable theoretical benefit from the more general branching alone and because the theoretical benefits of the restricted kinds of learned linear inequalities from conflicts available even in existing solvers are not at all clear.

From a proof complexity perspective, the SP system turns out to be polynomially equivalent to the semi-algebraic proof system *tree-like* R(CP), introduced by Krajíček [23]. Roughly speaking one can think of R(CP) as a mutual generalization of both Cutting Planes and Resolution – the lines of an R(CP) proof are clauses of integer linear inequalities, and in a single step one can take two clauses and either apply a cutting-planes rule to a single inequality in each clause or apply a resolution-style "cut". However, SP perspective turns out to be quite useful: we show that SP has quasi-polynomial size refutations of the Tseitin principle, and also that SP can polynomially-simulate Cutting Planes (neither result was previously known to hold for tree-like R(CP)).

We also investigate the relationship between SP refutations and communication complexity. Given an unsatisfiable CNF $\mathcal{F}$ and any partition of the variables (X, Y) of $\mathcal{F}$ into two sets, one can consider the following two-party search problem $\mathsf{Search}_{X,Y}(\mathcal{F})$: Alice receives an assignment to the X-variables, Bob receives an assignment to the Y-variables, and they must communicate and output a falsified clause of $\mathcal{F}$ under their joint assignment. At this time *all* strong lower bound results for Cutting Planes refutations essentially follow from studying the communication complexity of $\mathsf{Search}_{X,Y}(\mathcal{F})$. In particular, depth-$d$ (respectively, length-L tree-like, length-L space s) CP refutation of $\mathcal{F}$ yields an d-round (respectively, $O(\log L)$-round, $O(s \log L)$-round) real communication protocols for $\mathsf{Search}_{X,Y}(\mathcal{F})$, and a length-$L$ CP refutation of $\mathcal{F}$ yields a size L real communication game [24, 27, 6, 11, 16].

Each of these results has been used to derive strong lower bounds on Cutting Planes refutations by proving the corresponding lower bound against the search problem [24, 11, 17, 13, 27, 6]. Furthermore, the above lower bound techniques hold even for the stronger *semantic* Cutting Planes system (the lines of which are integer linear inequalities, and from two integer linear inequalities we are allowed to make *any* sound deduction over integer points) [12]. This makes the known lower bounds much stronger, and it is quite surprising that all one needs to exploit for strong lower bounds is that the lines are linear inequalities (rather than exploiting some weakness of the deduction rules). However, this strength also illustrates a weakness of current techniques, as once the lines of a proof system $\mathcal{P}$ become expressive enough, semantic proof techniques (i.e. ones that work for the semantic version of the proof systems) completely break down since every tautology has a short semantic proof. Therefore, it is of key importance to develop techniques which truly exploit the "syntax" of proof systems, and not just the expressive power of the lines.

Hence, it is somewhat remarkable that we are able to show that these results still hold if we replace real communication protocols with SP refutations. That is, we show:

- A depth-d CP refutation of $\mathcal{F}$ yields a depth-d SP refutation of $\mathcal{F}$.
- A length-L tree-like CP refutation of $\mathcal{F}$ yields a depth $O(\log L)$ SP refutation of $\mathcal{F}$.
- A length-L, space s CP refutation of $\mathcal{F}$ yields a depth $O(s \log L)$ SP refutation of $\mathcal{F}$.
- A length-L CP refutation of $\mathcal{F}$ yields a size $O(L)$ SP refutation of $\mathcal{F}$.

Since SP is a syntactic system this further motivates studying its depth- and size-complexity. We can use semantic techniques to get some lower bounds for SP: we show that a size-S and depth-d SP refutation yields a real communication protocol with cost $O(d)$ and for which

the protocol tree has size $O(S \cdot n)$. This simulation yields new proofs of some depth lower bounds already known in the literature; however, these lower bounds are complemented by showing that neither SP refutations nor real communication protocols can be balanced. This should be viewed in a positive light: the depth- and size-complexity problems are truly different for SP, and furthermore, one seemingly cannot obtain size lower bounds for SP by proving depth lower bounds for real communication protocols (in contrast to, say, tree-like Cutting Planes). In sum, SP appears to be a very good candidate for a proof system on the "boundary" where current techniques fail to prove strong size lower bounds.

The rest of the paper is outlined as follows. After some preliminaries in Section 2, we give a simple refutation of the Tseitin problem in SP in Section 3. In Section 4, we prove a raft of simulation and equivalence results for SP – showing it is equivalent to R(CP), relating it to Cutting Planes in various measures such as depth, length, and space, and showing how an SP proof yields a real communication protocol for the canonical search problem. Finally, in Section 5, we prove depth lower bounds for SP and some impossibility results for balancing.

2 Preliminaries

Before we define the new proof system formally, we need to make a few general definitions that are relevant to semi-algebraic proof systems.

An *integer linear inequality* (or simply a *linear inequality*) in the variables $x = x_1, \ldots, x_n$ is $Ax \geq b$, where $A \in \mathbb{Z}^n$ and $b \in \mathbb{Z}$. A system of linear inequalities $\mathcal{F}$ is *unsatisfiable* if there is no Boolean assignment $\alpha \in \{0,1\}^n$ which simultaneously satisfies every inequality in $\mathcal{F}$. We sometimes refer to inequalities as lines and write $L \equiv Ax \geq b$. The integer negation of a line L is the inequality $\neg L \equiv Ax \leq b - 1$.

An unsatisfiable formula in a conjuctive normal form (CNF) defines an unsatisfiable system of linear inequalities $\mathcal{F}$ in a natural way. A clause $\bigvee_{i=1}^{k} x_i \vee \bigvee_{i=1}^{l} \neg x_i$, is translated into the inequality $\sum_{i=1}^{k} x_i + \sum_{i=1}^{l}(1 - x_i) \geq 1$, and $\mathcal{F}$ is the set of translations of all clauses. We assume that $\mathcal{F}$ always contains the axioms $x_i \geq 0$ and $-x_i \geq -1$ for all variables x_i, as we are interested in propositional proof systems for refuting unsatisfiable Boolean formulas.

▶ **Definition 1.** A *propositional proof system* $\mathcal{P}$ is a non-deterministic polynomial time Turing machine (TM) deciding the language of unsatisfiable CNF formulas. Given an unsatisfiable CNF, the NP-certificate is called *the proof* or *the refutation*.

The strength of proof systems is compared using the notion of polynomial simulation.

▶ **Definition 2.** Let $\mathcal{P}_1$ and $\mathcal{P}_2$ be two proof systems. We say that $\mathcal{P}_1$ *polynomially simulates* $\mathcal{P}_2$ if for every unsatisfiable formula $\mathcal{F}$, the shortest refutation of $\mathcal{F}$ in $\mathcal{P}_1$ is at most polynomially longer than the shortest refutation in $\mathcal{P}_2$. $\mathcal{P}_1$ is *strictly stronger* than $\mathcal{P}_2$ if $\mathcal{P}_1$ polynomially simulates $\mathcal{P}_2$, but the converse does not hold. Finally, we say that $\mathcal{P}_1$ and $\mathcal{P}_2$ are *incomparable* if neither can polynomially simulate the other.

We now describe the proof system Stabbing Planes, our central object of study.

▶ **Definition 3.** Let $\mathcal{F}$ be an unsatisfiable system of linear integral inequalities. A *Stabbing Planes (*SP*) refutation* of $\mathcal{F}$ is a threshold decision tree: a directed binary tree in which each edge is labelled with a linear integral inequality. If the right outgoing edge of a node is labelled with $Ax \geq b$, then the left outgoing edge has to be labelled with its integer negation, $Ax \leq b-1$. We refer to Ax (or the pair of inequalities $Ax \leq b-1, Ax \geq b$) as *the query* corresponding to the node. The *slab* corresponding to the query is $\{x^* \in \mathbb{R}^n \mid b - 1 < Ax^* < b\}$.

Let the set of all paths from the root to a leaf in the tree be denoted by $\{p_1, \ldots, p_\ell\}$. Each leaf i is labelled with a non-negative linear combination of inequalities in $\mathcal{F}$ with the inequalities along the path p_i that yields $0 \geq 1$.

The *size* of a SP refutation is the number of bits needed to represent every inequality in the refutation. The *length* of a SP refutation is the number of nodes in the threshold tree. The size (length) of refuting a system of linear inequalities $\mathcal{F}$ in SP is the minimum size (length) of any SP refutation of $\mathcal{F}$. The *rank* or *depth* of a SP refutation $\mathcal{P}$ is the longest root-to-leaf path in the threshold tree of $\mathcal{P}$. The rank (depth) of refuting $\mathcal{F}$ in SP is the minimum rank (depth) over all SP refutations of $\mathcal{F}$.

Refutations in SP have an intuitive geometric interpretation: each step of a refutation can be viewed as nondeterministically removing a slab from the solution space and recursing on the resulting polytopes on both sides of the slab. The aim is to recursively cover the solution space with slabs until every feasible point within this polytope is removed. An example of this can be seen in Figure 1 in the introduction. In particular, the polytope at any step of the recursion is empty if and only if there exists a convex combination of the axioms and inequalities labelling the corresponding root-to-leaf path in the refutation equivalent to $0 \geq 1$. This is summarized in the following fact which follows directly from the Farkas' lemma. The "moreover" part of the following fact is an application of Carathéodory's theorem, and will be useful for technical reasons later in the paper. We refer the interested reader to [30] for some background on polytope theory.

▶ **Fact 4.** *Let $\mathcal{F} = \{A_1 x \geq b_1, \ldots, A_m x \geq b_m\}$ be a system of integer linear inequalities. The polytope defined by $\mathcal{F}$ is empty if and only if there is a non-negative (rational) linear combination of the inequalities of $\mathcal{F}$ which evaluates to $0 \geq 1$. Moreover, the non-negative linear combination can be taken to be supported on $\leq n$ of the inequalities from $\mathcal{F}$, where n is the dimension of the space to which x belongs.*

It is straightforward to see that SP is a sound and complete proof system. Completeness follows from a simple observation that SP polynomially simulates DPLL. To see that SP is sound, let $\mathcal{R}$ be a SP refutation of some formula $\mathcal{F}$. Observe that for any node in $\mathcal{R}$ with outgoing edges labelled $Ax \geq b$ and $Ax \leq b - 1$, any $0 - 1$ assignment to the variables $\alpha \in \{0,1\}^n$ must satisfy exactly one of the two inequalities. Therefore, if a Boolean solution $\alpha \in \{0,1\}^n$ satisfies $\mathcal{F}$, then for at least one of the leaves of $\mathcal{R}$, one cannot derive $0 \geq 1$. This follows by Fact 4 because the polytope formed by the inequalities labelling root-to-leaf path is non-empty (α lies in this polytope) .

Next we recall a well-known and extensively-studied proof system: Cutting Planes (CP). For an introduction to Cutting Planes, we refer an interested reader to Chapter 19 in [21].

▶ **Definition 5.** Let $\mathcal{F}$ be an unsatisfiable system of linear inequalities. A *Cutting Planes* (CP) *refutation* of $\mathcal{F}$ is a sequence of linear inequalities $\{L_1, \ldots, L_\ell\}$ such that $L_\ell = 0 \geq 1$ and each L_i is either an axiom $\in \mathcal{F}$ or is obtained from previous lines via one of the following inference rules. Let α, β be positive integers.

$$\text{Linear Combination: } \frac{Ax \geq a \qquad Bx \geq b}{(\alpha A + \beta B)x \geq \alpha a + \beta b} \qquad \text{Division: } \frac{\alpha A x \geq b}{Ax \geq \lceil \frac{b}{\alpha} \rceil}$$

We refer to ℓ as the *length* of the refutation. The *length* of refuting $\mathcal{F}$ in CP is the minimum length of a CP refutation of $\mathcal{F}$.

The directed acyclic graph (DAG) $G = (V, E)$ associated with a CP refutation $\{L_1, \ldots, L_\ell\}$ is defined as follows. We have $V = \{L_1, \ldots, L_\ell\}$ and $(u, v) \in E$ if and only if the line labelling v was derived by an application of an inference rule involving the line labelling u. Without

loss of generality, we may assume that there is only one vertex with out degree 0, which we call the root. The root of G is labelled with L_ℓ and the leaves are labelled with the axioms.

The *rank* or *depth* of the refutation is the length of the longest root-to-leaf path in G. The rank of refuting an unsatisfiable system of linear inequalities $\mathcal{F}$ is the minimum rank of any refutation of $\mathcal{F}$ in the given proof system. Finally, *tree-like* CP is defined by restricting proofs to be such that the underlying graph G is a tree.

It is not known if an arbitrary SP refutation can be transformed into a refutation with coefficients of polynomial bitsize. Therefore we currently must make the distinction between SP refutation size and length. Fortunately, all of our results hold in the best possible scenario; our upper bounds are low weight (polynomial-length); the simulations are length preserving, and our lower bounds hold for any weight.

3 Motivating Example: SP Refutations of Tseitin Formulas

Tseitin contradictions are among the most well-studied unsatisfiable formulas in proof complexity, and are the quintessential formulas that are believed to be hard for CP [21]. Despite the fact that exponential lower bounds for CP are known for many natural families of formulas (including recent lower bounds for random $O(\log n)$-CNF formulas), there are no nontrivial lower bounds known for the Tseitin contradictions, and for good reason: the only known lower bound method for CP reduces the problem of refuting a formula in CP to a monotone circuit problem, for which the corresponding monotone circuit problem for Tseitin contradictions is easy.

In this section, we demonstrate the power of Stabbing Planes by showing that there exists a shallow quasi-polynomial size SP refutation of the Tseitin formulas. This, together with our simulation results from Section 4, show that SP is provably more powerful than CP in terms of depth, and strongly suggests that SP is strictly more powerful than CP.

Tseitin contradictions are any unsatisfiable family of mod-2 equations subject to the constraint that every variable occurs in exactly two equations. An instance of Tseitin, denoted $\text{Tseitin}(G, \ell)$ is defined by a connected undirected graph $G = (V, E)$ and a node labelling $\ell \in \{0,1\}^V$ of odd total weight: $\sum_{v \in V} \ell_v = 1 \bmod 2$. For each edge $e \in E$ there is a variable x_e in $\text{Tseitin}(G, \ell)$, and for each vertex $v \in V$ an equation $\sum_{e \ni v} x_e \equiv \ell_v \mod 2$, stating that the sum of the variables x_e incident with v is $\ell_v \bmod 2$. The edge equations sum to zero mod 2 since every variable occurs exactly twice, but the vertex equations sum to one mod 2, since the node labelling is odd, and therefore the equations are unsatisfiable. When G has degree D, we can express $\text{Tseitin}(G, \ell)$ as a D-CNF formula containing $|V| \cdot 2^{D-1}$ clauses.

The obvious way to refute $\text{Tseitin}(G, \ell)$ under an assignment x is to find a vertex w for which the corresponding vertex equation is falsified. This can be achieved by the following divide-and-conquer procedure, which maintains a set $U \subseteq V$ such that $w \in U$. The process begins by setting $U = V$. Then, V is partitioned arbitrarily into two sets V_1, V_2 of roughly the same size. Query x_e for all edges e crossing the cut (V_1, V_2), and suppose that the sum of all such x_e is odd (the case when it is even is similar). We know that either $\sum_{v \in V_1} \ell_v$ or $\sum_{v \in V_2} \ell_v$ is even: if the first sum is even then the Tseitin formula restricted to V_1 contains a contradiction, and otherwise the formula restricted to V_2 contains a contradiction. In either case, we can remove roughly half of the graph and recurse.

By keeping track of a few more variables, we can repeat this procedure recursively until $|U| = 1$. Since we reduce the size of U by half each time, this procedure results in the recursion depth logarithmic in $|V|$. It turns out that this procedure can be realized in Stabbing Planes, where recursion depth roughly corresponds to the depth of the refutation. This results in a quasi-polynomial size refutation.

▶ **Theorem 6.** *Let $G = (V, E)$ be an undirected graph, and let ℓ be a $\{0,1\}$ vertex labelling with odd total weight. Then* Tseitin(G, ℓ) *has an* SP *refutation of size $n^{O(\log n + D/\log n)}$ and rank $O(D + \log^2 n)$, where $n = |V|$ and D is the maximum degree in G.*

Proof. If $U \subseteq V$ is a set of vertices, then let $E(U) = \{uv \in E \mid u, v \in U\}$, and $\mathrm{Cut}(U) = \{uv \in E \mid u \in U, v \in \overline{U}\}$. Similarly, if $U_1, U_2 \subseteq V$ are disjoint then we let $\mathrm{Cut}(U_1, U_2) = \{uv \in E \mid u \in U_1, v \in U_2\}$. We construct the SP refutation recursively. During the recursion we maintain a set U of current vertices (initially $U = V$). At each recursive step, we split U into two halves U_1 and U_2 and query the total weight k of the edges crossing (U_1, U_2) via SP inequalities. Knowing k, a few additional queries allows us to determine which of U_1 or U_2 contains a contradiction, and we then recurse on the corresponding set of vertices.

We construct a proof while maintaining the following invariant: for the current subset of vertices $U \subseteq V$, we have queried linear inequalities implying that $\sum_{e \in \mathrm{Cut}(U)} x_e = k$ for some $0 \le k \le |\mathrm{Cut}(U)|$ such that $k \not\equiv \sum_{v \in U} \ell_v \bmod 2$. Note that this invariant ensures that our Tseitin instance restricted to the edges incident on U is unsatisfiable, since summing up all vertex constraints within U yields $\sum_{e \in \mathrm{Cut}(U)} x_e + \sum_{e \in E(U)} 2x_e \equiv k \not\equiv \sum_{v \in U} \ell_v \pmod 2$.

Initially we have $U = V$ and the invariant clearly holds. Now, let U be the current set of vertices. By the invariant we know that $\sum_{e \in \mathrm{Cut}(U)} x_e = k$ for some $k \not\equiv \sum_{v \in U} \ell_v \bmod 2$. Partition U into two halves U_1 and U_2 arbitrarily, subject to $|U_1| = \lfloor |U|/2 \rfloor$. We first determine the value of the edges going between U_1 and U_2 by querying $\sum_{e \in \mathrm{Cut}(U_1, U_2)} x_e \ge \beta$ for $\beta = 1, \ldots, |\mathrm{Cut}(U_1, U_2)|$. To each leaf of this tree we attach a second binary search tree for determining the value $|\mathrm{Cut}(U_1)|$ by querying $\sum_{e \in \mathrm{Cut}(U_1)} x_e \ge \gamma$ for $\gamma = 1, \ldots, |\mathrm{Cut}(U_1)|$. After these queries, at each leaf of the "combined" tree we will have $\sum_{e \in \mathrm{Cut}(U_1, U_2)} x_e = \beta$ and $\sum_{e \in \mathrm{Cut}(U_1)} x_e = \gamma$ for some β and γ. Furthermore, since $|\mathrm{Cut}(U_1)| + |\mathrm{Cut}(U_2)| = |\mathrm{Cut}(U)| + 2|\mathrm{Cut}(U_1, U_2)|$, we will have $\sum_{e \in \mathrm{Cut}(U_2)} x_e = \delta$, where $\delta + \gamma = k + 2\beta$, $0 \le \delta \le |\mathrm{Cut}(U_2)|$.

For any leaf of this tree where $\delta > |\mathrm{Cut}(U_2)|$, we can derive a contradiction by summing the axioms $-x_e \ge -1$ for all $e \in \mathrm{Cut}(U_2)$ with $\sum_{e \in \mathrm{Cut}(U_2)} x_e \ge \delta$. Otherwise, for the remaining leaves observe that $\delta + \gamma \equiv k \not\equiv \sum_{v \in U_1} \ell_v + \sum_{v \in U_2} \ell_v \pmod 2$. Now, if $\gamma \not\equiv \sum_{v \in U_1} \ell(v) \bmod 2$, then recurse on U_1. Otherwise, $\delta \not\equiv \sum_{v \in U_2} \ell(v) \bmod 2$. Then recurse on U_2.

Our recursion terminates when U contains a single vertex v. By the invariant, we have derived $\sum_{e \in \mathrm{Cut}(v)} x_e \equiv k \not\equiv \ell(v) \bmod 2$ for some $0 \le k \le |\mathrm{Cut}(\{v\})|$. The axioms of Tseitin$(G, \ell)$ rule out Boolean assignments to the variables x_e for $e \in \mathrm{Cut}(\{v\})$, which contradict $\ell(v)$; these axioms do not prohibit incorrect fractional assignments. Therefore, to derive a contradiction, we still need to enforce that the variables x_e for $e \in \mathrm{Cut}(\{v\})$ take $\{0,1\}$ values. We achieve this by querying all variables x_e for $e \in \mathrm{Cut}(\{v\})$ via SP inequalities $x_e \ge 1, x_e \le 0$. This results in a complete binary tree of depth $\le D$. Clearly, $0 \ge 1$ is immediately obtained at the leaves that disagree with $\sum_{e \in \mathrm{Cut}(\{v\})} x_e = k$. At the leaves that agree with $\sum_{e \in \mathrm{Cut}(\{v\})} x_e = k$, the inequality $0 \ge 1$ immediately follows from the assignment to the edges incident to v and one of the axioms of Tseitin.

Finally, we analyze the size and rank of the constructed SP proof. In each recursive step, we make $O((nD)^2)$ queries to determine weights of edges crossing the two cuts. Each recursive step is computed by a pair of binary trees, each of depth at most $\log(nD)$. Our recursion terminates in $\log n$ rounds because we halve the number of current vertices in each step. Once the recursion terminates, we query the variables corresponding to edges incident to the single remaining vertex – this contributes 2^D factor to size and increases depth by at most D. Overall, the SP proof has size $n^{O(\log n + D/\log n)}$ and rank $O(D + \log^2 n)$. ◀

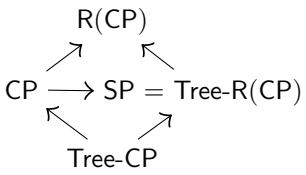
▶ **Corollary 7.** SP *is strictly stronger than* CP *with respect to proof rank.*

Proof. By Theorem 11, any Cutting Planes refutation of rank r can be converted into a SP refutation of rank $O(r)$. Buresh-Oppenheim et al. proved $\Omega(n)$ lower bound on the rank of Cutting Planes of the Tseitin formulas on constant-degree expander graphs [7], while Theorem 6 shows that SP can refute such Tseitin formulas in rank $O(\log^2 n)$. ◀

4 Simulation Theorems

In this section, we prove simulation theorems relating the SP proof system to other similar proof systems in the literature. We begin by showing that SP is polynomially equivalent to the tree-like R(CP) system (introduced by Krajicek in [23]), which can be thought of as tree-like Resolution with clauses of inequalities and allowing CP rules. Since tree-like R(CP) simulates tree-like CP, the natural question is whether SP (and consequently R(CP)) can simulate general CP. We answer this question positively by providing two simulations. First of all, we observe that SP can depth-simulate CP. This simulation, while preserving depth of the proof, can lead to an exponential increase in the size. Via a different simulation we show that SP can size-simulate CP. This time around, while the simulation preserves the size of a CP refutation, it can significantly increase the depth. It is an interesting open question whether there is a simulation of CP by SP that can simultaneously preserve depth and size of CP refutations.

To complete the picture, we note that general R(CP) can trivially simulate tree-like R(CP) (and consequently SP) and CP. We also show that tree-like CP refutations can be efficiently converted into balanced (logarithmic-depth) SP refutations – this shows that tree-like CP refutations, which cannot in general be balanced, *can* be balanced in SP.



We then turn to the question of space-time simulations. Recall, that a proof system can be thought of as a non-deterministic Turing machine. The notion of space of CP refutations intuitively corresponds to the minimum size of the work tape of such a non-deterministic Turing machine that is required to carry out the computation. In this analogy, the notion of length of CP refutations corresponds to the running time of the given TM. We show that general CP refutations that use length ℓ and space s can be turned into depth $O(s \log \ell)$ SP refutations. Thus, sufficiently strong lower bounds on the depth of SP refutations lead to time-space tradeoffs for CP.

4.1 Equivalence of SP with tree-like RCP

We show the SP system is polynomially equivalent to the R(CP) proof system. R(CP) proof system is formally defined as follows.

▶ **Definition 8.** The R(CP) proof system is a syntactic proof system defined as follows. The lines of the R(CP) system are disjunctions of integer linear inequalities $\Gamma = L_1 \vee L_2 \vee \cdots \vee L_\ell$, and the lines are equipped with the following deductive rules. Let Γ be an arbitrary disjunction of integer linear inequalities, let $Ax \geq b, Cx \geq d$ be arbitrary integer linear inequalities, and let α, β be any positive integers.

$$\text{Linear Combination: } \frac{(Ax \geq b) \vee \Gamma \qquad (Cx \geq d) \vee \Gamma}{(\alpha A + \beta C)x \geq (\alpha b + \beta d) \vee \Gamma} \qquad \text{Division: } \frac{(\alpha Ax \geq b) \vee \Gamma}{(Ax \geq \lceil b/\alpha \rceil) \vee \Gamma}$$

$$\text{Axiom Introduction: } \frac{}{(Ax \geq b) \vee (Ax \leq b-1)} \qquad \text{Weakening: } \frac{\Gamma}{(Ax \geq b) \vee \Gamma}$$

$$\text{Cut: } \frac{(Ax \geq b) \vee \Gamma \qquad (Ax \leq b-1) \vee \Gamma}{\Gamma} \qquad \text{Elimination: } \frac{(0 \geq 1) \vee \Gamma}{\Gamma}$$

An R(CP) proof is *tree-like* if the proof DAG is a tree.

The following two theorems state the polynomial equivalence between SP and R(CP). Due to space considerations, we refer the reader to the full version [1] of this paper for the proofs of the two theorems.

▶ **Theorem 9.** *Let $\mathcal{C}$ be an unsatisfiable CNF, and let $C_1, C_2, \ldots, C_m$ be the representation of $\mathcal{C}$ as an integer-linear system of inequalities. For any* SP *refutation of $\mathcal{C}$ with size s and depth d there is a tree-like* R(CP) *refutation of $\mathcal{C}$ of size $O(s(d^2 + dm))$ and width $d + 1$.*

▶ **Theorem 10.** *Let $\mathcal{C}$ be an unsatisfiable CNF, and let $C_1, C_2, \ldots, C_m$ be the representation of $\mathcal{C}$ as an integer linear system of equations. For any tree-like* R(CP) *proof of the disjunction $\neg C_1 \vee \neg C_2 \vee \cdots \vee \neg C_m$ with size s and depth d there is an* SP *refutation of $\mathcal{C}$ of size at most $2s$ and rank at most $2d$.*

4.2 SP simulations of Cutting Planes

SP simulates CP in two ways: via a depth-preserving simulation and via a size-preserving simulation. It is an interesting open problem if both depth and size can be preserved simultaneously during a simulation. Due to space requirements, we simply state the depth-preserving simulation and leave its proof to the full version of the paper [1].

▶ **Theorem 11.** *For every Cutting Planes refutation of rank d, there is a* SP *refutation of the same tautology with rank at most $2d$. Moreover, if the* CP *refutation is tree-like of size s then the resulting* SP *refutation is of size $O(s)$ and rank $2d$.*

Next, we present the size-preserving simulation with full details.

▶ **Theorem 12.** SP *polynomially simulates* CP.

Proof. Let $\mathcal{R} = \{A_1 x \geq a_1, A_2 x \geq a_2, \ldots, A_m x \geq a_m\}$ be a CP refutation of an unsatisfiable set of integer linear inequalities $\mathcal{F}$. We construct a SP refutation of $\mathcal{F}$ line by line, following $\mathcal{R}$. Our SP refutation is a tree where the right-most path is of length $m+1$ with edges labelled $A_1 \geq b_1, \ldots, A_m \geq b_m$. The left child of node $i \leq m$ along this path is labelled with $0 \geq 1$, which is derived as a non-negative linear combination of $A_j x \geq a_j$ for $j < i$, $A_i x \leq a_i - 1$, and $\mathcal{F}$. The last node in the path is also labelled with $0 \geq 1$. Since $A_m x \geq a_m \equiv 0 \geq 1$, the last node is trivially labelled with $0 \geq 1$. Thus, we only need to show that the left child of every node can be legally labelled with $0 \geq 1$. If $A_i x \geq a_i$ is an axiom, we can derive $0 \geq 1$ by subtracting $A_i x \leq a_i - 1$ from $A_i x \geq a_i \in \mathcal{F}$. If $A_i x \geq a_i$ is a non-negative combination of two previous inequalities, i.e., $A_i x \geq a_i$ is $\alpha A_{j_1} x + \beta A_{j_2} x \geq \alpha a_{j_1} + \beta a_{j_2}$ for some $j_1, j_2 < i$ and $\alpha, \beta \in \mathbb{Z}_{\geq 0}$, we can derive $0 \geq 1$ by subtracting $A_i x \leq a_i - 1$ from the non-negative linear combination of $A_{j_1} x \geq a_{j_1}$ and $A_{j_2} x \geq a_{j_2}$ used to derive $A_i x \geq a_i$. Finally, suppose that $A_i x \geq a_i$ is obtained by an application of the division rule to $A_j x \geq a_j$ for some $j < i$, i.e., $A_i x \geq a_i$ is $\frac{A_j}{c} x \geq \left\lceil \frac{a_j}{c} \right\rceil$ where $c \in \mathbb{N}$ divides every entry in A_j. On the path to this node in our SP refutation we queried $A_j x \geq a_j$. Dividing this inequality by c and subtracting $A_i x \leq a_i - 1$, we obtain $0 \geq a_j/c - (\lceil a_j/c \rceil - 1)$. This gives us $0 \geq \frac{a_j}{c} + 1 - \left\lceil \frac{a_j}{c} \right\rceil$. Since the right-hand side is strictly positive this can be normalized to give $0 \geq 1$. ◀

4.3 CP and Balanced SP

It is known that CP refutations cannot be balanced (i.e. size-s refutations being turned into size $O(s)$ depth $O(\log s)$ refutation) in CP. Here, we show that CP proofs can be turned into balanced SP proofs. More specifically, we prove the following.

▶ **Theorem 13.** *Suppose there is a size s tree-like* CP *refutation of a set of linear integer inequalities $\mathcal{F}$. Then there is a size s depth $O(\log s)$* SP *refutation of $\mathcal{F}$.*

Proof. The construction is recursive. Let T be the tree corresponding to $\mathcal{P}$. If $|T| = O(1)$, use one of the previous simulation theorems to create an SP refutation of $\mathcal{P}$. Now, let v be a node in T such that the subtree T_v rooted at v satisfies $|T|/3 \leq |T_v| \leq 2|T|/3$, which must exist since the size measure is additive. Let $Bx \geq b$ be the line in $\mathcal{P}$ corresponding to v. Our SP simulation starts by querying Bx. If $Bx \geq b$ then we apply the recursive construction to $T \setminus T_v$, treating $Bx \geq b$ as a new axiom. Otherwise, if $Bx \leq b-1$ (which contradicts the the input set of inequalities $\mathcal{F}$) we apply the recursive construction to T_v. The size is clearly preserved, and the depth of the proof becomes logarithmic, since we are reducing the size of the proof to be simulated by a constant factor on each branch of a query. ◀

We can also show that bounded depth and space CP refutations yield balanced SP proofs. See full version [1] for the proof of the following theorem.

▶ **Theorem 14.** *Suppose that we have a length ℓ, space s* CP *refutation of an unsatisfiable set of linear integral inequalities. Then there is a depth $O(s \log \ell)$* SP *refutation of the same set of linear integral inequalities.*

5 Impossibility Results

In this section, we prove near-optimal lower bounds on SP rank via reductions to randomized and real communication complexity. We then tackle the harder problem of proving unrestricted superpolynomial size lower bounds for SP. Although we are unable to prove such lower bounds we explain why current approaches fail. Essentially all lower bounds for CP have been obtained by reducing to a communication complexity problem; in the case of tree-like CP, the reduction is to the communication complexity of a corresponding search problem. For more general dag-like CP, the reduction is to the size of "communication games" [13, 17] (communication games are a dag-like model of communication that gives an equivalence between communication *size* and monotone circuit size, analogous to the famous equivalence between communication *depth* and monotone formula size). Although tree-like CP proofs cannot be balanced in general, communication protocols (both deterministic and randomized) can be balanced, and thus tree-like CP lower bounds follow from communication complexity lower bounds. Similarly, we show that it is not possible to balance SP refutations, and thus we cannot in general obtain size lower bounds directly from rank lower bounds. Moreover, we show that SP refutations imply real communication protocols, and unlike ordinary communication protocols, we show that real protocols *cannot* in general be balanced. This rules out proving length lower bounds on SP refutations from (real) communication complexity lower bounds.

5.1 SP Refutations Imply Communication Protocols

Real communication protocols were introduced by Krajíček [24]. In this model, the players are allowed to communicate by sending real-valued functions of their inputs to a referee who announces their comparison.

▶ **Definition 15.** A *real communication protocol* is a full binary tree in which every non-leaf node v is labeled with a pair of functions $a^v : \mathcal{X} \to \mathbb{R}, b^v : \mathcal{Y} \to \mathbb{R}$, and the leaves are labelled with elements in $\mathcal{Z}$. Two players, Alice and Bob, receive inputs from $\mathcal{X} \times \mathcal{Y}$, with Alice receiving $x \in \mathcal{X}$ and Bob receiving $y \in \mathcal{Y}$. Beginning at the root, the players traverse the tree as follows: at each node, they send real values $a^v(x)$ and $b^v(y)$ to a "referee" who returns (to both of them) a bit indicating the result of the comparison $a^v(x) \geq b^v(y)$; the players proceed to the left child if $a^v \geq b^v$, and to the right child otherwise. Once they reach a leaf, the protocol halts, and the players output the value in $\mathcal{Z}$ labelling the leaf; it computes a function $f : \mathcal{X} \times \mathcal{Y} \to \mathcal{Z}$ in the natural way.

The *cost* of a real protocol is the depth of the tree, or equivalently the maximum number of rounds of communications with the referee over any input (x, y), and the *size* is the number of nodes in the protocol. Similarly, the cost (size) of computing a function f is the smallest cost (size) real protocol computing f.

Krajíček showed that from a low-rank CP refutation of an unsatisfiable CNF, one can obtain a real communication protocol for solving a related search problem [24]. We describe this search problem next.

▶ **Definition 16.** Let $\mathcal{F} = C_1 \wedge C_2 \wedge \cdots \wedge C_m$ be an unsatisfiable CNF and (X, Y) be a partition of the variables. The relation $\mathsf{Search}_{X,Y}(\mathcal{F}) \subseteq \{0,1\}^X \times \{0,1\}^Y \times [m]$ consists of all triples (x, y, i) such that the total assignment $z = (x, y)$ to all of the variables of C_i falsifies the clause i.

The search problem is the natural interpretation of a refutation in the setting of communication. Indeed, essentially every lower bound for CP has been proved by reducing to the communication complexity of the search problem. In a similar manner, we show that SP refutations can be turned into both randomized and real protocols for the search problem which preserve the rank of the refutation.

▶ **Lemma 17.** *Let $\mathcal{F}$ be an unsatisfiable CNF formula and (X, Y) be any partition of the variables. Any* SP *refutation of $\mathcal{F}$ of rank r implies a real communication protocol of cost $O(r + \log n)$ and an $O(r \log n + \log^2 n)$ randomized bounded-error protocol for* $\mathsf{Search}_{X,Y}(\mathcal{F})$.

The protocol consists of traversing the SP tree until a leaf is reached (r rounds), then finding a clause falsified at the leaf in $O(\log n)$ rounds: note that evaluating any linear inequality can be done using a single bit of communication.

For the second part of the lemma, Alice and Bob run the ε-error $O(\log m + \log \varepsilon^{-1})$-bit protocol of Nisan [26] for deciding an m-bit linear inequality. By the well-known result of Muroga [25], any inequality on n Boolean variables only requires coefficients representable by $O(n \log n)$ bits (recall that Alice and Bob's input will always be a Boolean assignment and so this suffices). Because there are at most $r + \log n$ inequalities evaluated along any root-to-leaf path in the refutation, the protocol is repeated at most $r + \log n$ many times. By a union bound, we require $\varepsilon < c/(\log n + r)$, where c is some constant bound on the error that we allow. Therefore, every inequality can be computed in $O(\log n + \log r)$ many bits to compute, giving a $O(r \log n + \log^2 n)$ bounded-error randomized protocol for $\mathsf{Search}_{X,Y}(\mathcal{F})$.

5.2 Rank Lower Bounds For SP

To take advantage of Lemma 17, we need to find some candidate formulas on which to prove rank lower bounds and then study the search problem obtained from applying this transformation. We do so for both the Tseitin formulas and a variant of the pebbling

contradictions, a reformulation of the classical black pebbling games as an unsatisfiable 3-CNF formula, originally introduced by Ben-Sasson et al. [4, 3].

The black pebbling game can be phrased as a contradictory 3-CNF as follows: Let G be a DAG with a set of source nodes $S \subseteq V(G)$ (having fanin 0), a unique sink node t (with fanin 2 and fanout 0), and the remaining nodes each having fanin exactly 2. The pebbling contradictions Peb_G consists of the following $n+1$ clauses over variables $v \in V(G)$: sink axiom: a single clause $\neg t$; source axioms: a clause s for every source $s \in S$; pebbling axioms: a clause $\neg u \vee \neg v \vee w$ for every $w \in V \setminus S$ with immediate children u, v.

Unfortunately, both the pebbling contradictions and the Tseitin formulas have short SP refutations. In particular, for any graph G, the polytope formed by the constraints of Peb_G is empty and therefore a nonnegative combination of the constraints yielding $0 \geq 1$ exists, this is a valid rank-1 SP refutation. For Tseitin, this follows from the poly-logarithmic rank upper bound in Theorem 6. We modify these formulas to make them harder to solve.

A standard technique for amplifying the hardness of computing some function $f : \mathcal{X}^n \to \mathcal{Z}$ is by *lifting* that function. This is done by obscuring the input variables by replacing them each by a small function $g : \mathcal{A} \to \mathcal{X}$ known as a *gadget*, which must be evaluated before learning the input to the original function. For an input $\alpha \in \mathcal{A}^n$, the function f *lifted* by gadget g is then $(f \circ g^n)(\alpha) = (g(\alpha_1), \ldots, g(\alpha_n))$. The intuition is that this lifted function $f \circ g^n$ should be much harder than the original because the players must first evaluate the gadget $g(\alpha_i)$ to learn each bit of the actual input to the function f. Furthermore, intuition says that if the gadget is sufficiently difficult to compute, then the model will be reduced to using much more rudimentary methods to evaluate the lifted function.

The standard hard-to-compute gadget is the *pointer* or *index* gadget, $\mathsf{IND}_\ell : [\ell] \times \{0,1\}^\ell \to \{0,1\}$. For an input $(x, y) \in [\ell] \times \{0,1\}^\ell$, x is a $\log \ell$-bit string encoding a pointer into the ℓ-bit string $y \in \{0,1\}^\ell$. The output of $\mathsf{IND}_\ell(x, y)$ is $y[x]$, the x-th bit in the string y. This is most often applied in communication complexity, where typically the variable partition between the players is that Alice is given $x \in [\ell]$ and Bob is given $y \in \{0,1\}^\ell$. In any standard model of communication, for this partition of the variables, it is difficult to imagine any communication protocol which could compute the index gadget with significant advantage over the trivial protocol; sending every bit of Alice's pointer x to Bob.

Raz and McKenzie formalized this intuition, in what has become known as a lifting theorem [28, 15]. They show that deterministic communication protocols cannot compute any function f lifted by the index gadget significantly better than simply mimicking a decision tree computing f, and performing the trivial protocol for evaluating the index gadget every time a bit of the input to f is needed.

Lifting theorems for real communication were originally proved by Bonet et al. [6] based on the techniques of Raz-McKenzie. Their theorem lifts lower bounds on the decision tree complexity of a function f to lower bounds on the cost of real communication protocols computing $f \circ \mathsf{IND}_\ell^n$. The decision tree complexity $\mathsf{DT}(f)$ of a function f is simply the minimum depth need by any decision tree to compute f. We use a simplified lifting theorem for real communication by de Rezende et al. [11], which we state next.

▶ **Theorem 18.** *(de Rezende et al. [11]) Let f be a function with domain $\{0,1\}^n$ and let $\ell = n^4$. If there is a real communication protocol of cost c that solves $f \circ \mathsf{IND}_\ell^n$ where Alice is given $x \in [\ell]^n$ and Bob is given $y \in \{0,1\}^{n\ell}$, then there is a decision tree solving f using $O(c/\log \ell)$ queries.*

Our goal is now to is to combine this theorem with Lemma 17 in order to prove lower bounds on the rank of SP refutations of $\text{Peb}_G \circ \mathsf{IND}_\ell^n$. Syntactically speaking though,

$\text{Peb}_G \circ \mathsf{IND}_\ell^n$ is not a valid input to our proof system. Therefore, we must show that the lifted function can indeed be phrased as a small CNF formula. The following encoding is due to Beame et al. [2]:

Let $\mathcal{F} = C_1 \vee \ldots \vee C_m$ be a CNF formula over variables $x_1, \ldots, x_n$. The CNF representing $\mathcal{F} \circ \mathsf{IND}_\ell^n$ is defined on new sets of variables $y_{i,j}$ and $z_{i,j}$ for all $i \in [n]$ and $j \in [\ell]$. This CNF has the following set of clauses

- Pointer clauses: for each $i \in [n]$, a clause $y_{i,1} \vee \ldots \vee y_{i,\ell}$.
- $\mathcal{F}$-clauses: for each clause $C_i \in \mathcal{F}$, where $C_i = y_{i_1} \vee \ldots \vee y_{i_k} \vee \neg x_{i_{k+1}} \vee \ldots \vee \neg x_{i_s}$ and for every $(j_1, \ldots, j_n) \in [\ell]^n$, a clause
  $(y_{i_1,j_1} \to z_{i_1,j_1}) \vee \ldots \vee (y_{i_k,j_k} \to z_{i_k,j_k}) \vee (y_{i_{k+1},j_{k+1}} \to \neg z_{i_{k+1},j_{k+1}}) \vee \ldots \vee (y_{i_s,j_s} \to \neg z_{i_s,j_s})$.

We will abuse notation and use $\mathcal{F} \circ \mathsf{IND}_\ell^n$ to denote the function, as well as it's CNF formulation, and use context to differentiate between the two.

A final subtlety that should be mentioned is that applying Theorem 18 to an SP refutation of $\text{Peb}_G \circ \mathsf{IND}_\ell^n$ yields a protocol for $\mathsf{Search}_{X,Y}(\text{Peb}_G \circ \mathsf{IND}_\ell^n)$ which is not in the correct form to apply Theorem 18 ($\mathsf{Search}_{X,Y}(\text{Peb}_G \circ \mathsf{IND}_\ell^n)$ is a function of a lifted function, whereas Theorem 18 can only be applied to lifted functions). Luckily, this is not a significant issue; Huynh et al. [18] show that, for any unsatisfiable CNF $\mathcal{F}$, any real communication protocol for $\mathsf{Search}_{X,Y}(\mathcal{F} \circ \mathsf{IND}_\ell^n)$, where $X = [\ell]^n$ and $Y = \{0,1\}^{n\ell}$, implies a real communication protocol for $\mathsf{Search}_{X,Y}(\mathcal{F}) \circ \mathsf{IND}_\ell^n$ with the same parameters.

It is now straightforward to obtain lower bounds on the rank of SP refutations. For the lifted pebbling formulas, SP rank lower bounds follow from combining Lemma 17 and Theorem 18 with a lower bound on the complexity of decision trees solving Peb_G proved by de Rezende et al.[11].

▶ **Theorem 19.** *There exists a graph G of indegree 2 on n vertices such that the unsatisfiable CNF formula $\text{Peb}_G \circ \mathsf{IND}_\ell^n$, for $\ell = n^4$, on $n(\ell + \log \ell)$ variables requires rank $\Omega(\sqrt{n \log n})$ to refute in* SP.

Proof. de Rezende et al. [11] showed the existence of a graph G on n vertices with indegree 2 such that the decision tree complexity of outputting a falsified clause of Peb_G is $\Omega(\sqrt{n/\log n})$. Applying Theorem 18 and combining this with the fact that shallow SP refutations give efficient protocols for the associated search problem (Lemma 17), proves the desired $\Omega(\sqrt{n \log n})$ lower bound on the rank of SP refutations of $\text{Peb}_G \circ \mathsf{IND}_\ell^n$. ◀

Finally, a similar technique can be applied to obtain a lower bound on the rank of SP refutations for a lifted variant of the Tseitin formulas. This follows from the lower bound on the randomized communication complexity of the search problem for the Tseitin formulas lifted by a small constant-size gadget, which was obtained by Göös and Pitassi [14]. In particular, they use the *versatile gadget*, $\mathsf{VER} : \mathbb{Z}_4 \times \mathbb{Z}_4 \to \{0,1\}$, which is defines as $\mathsf{VER}(x,y) = 1 \iff x + y \pmod 4 \in \{2,3\}$.

▶ **Theorem 20.** *(Göös and Pitassi [14]) There exists a constant-degree graph G on n vertices such that, if ℓ is any $\{0,1\}$ vertex labelling with odd total weight and (X,Y) is any partition of the variables, any bounded-error randomized communication protocol for $\mathsf{Search}_{X,Y}(\text{Tseitin}(G,\ell) \circ \mathsf{VER}^n)$ on $O(n)$ variables, requires $\Omega(n/\log n)$ bits of communication.*

Furthermore, Göös and Pitassi showed how, for any CNF formula $\mathcal{F}$, the composed function $\mathcal{F} \circ \mathsf{VER}^n$ can be encoded as a CNF formula. For brevity, we refer the reader to Göös and Pitassi [14] for the definition of encoding of $\mathcal{F} \cdot \mathsf{VER}^n$ as a CNF formula, and recall that if $\mathcal{F}$ is a CNF with m clauses and width w, then $\mathcal{F} \circ \mathsf{VER}^n$ contains at most $m \cdot 2^{4w}$

clauses. The width of every clause in the Tseitin formulas are bounded by maximal degree d in the underlying graph. Using this fact, we are able to obtain near-optimal lower bounds on the rank of SP refutations by combining Theorem 20 with Lemma 18 as in the proof of Theorem 19. This lower bound should be contrasted with the logarithmic-rank SP upper bound on Tseitin from Theorem 6.

▶ **Theorem 21.** *There a constant-degree graph G on n vertices such that if ℓ is any $\{0,1\}$ vertex labelling with odd total weight, the CNF formula* $\text{Tseitin}(G,\ell)\circ\mathsf{VER}^n$*, on $O(n)$ variables and clauses, requires* SP *refutations of rank $\Omega(n/\log^2 n)$.*

5.3 SP Refutations Cannot Be Balanced

Optimistically, one could hope that the length and rank of SP refutations may be closely related and therefore that we could leverage these rank bounds to obtain lower bounds on the length of SP refutations. We answer this question negatively, showing that there exists a contradictory CNF which admits short refutations, but for which these refutations must be almost maximally deep. That is, we show that SP refutations cannot be balanced; an SP refutation of length S does not imply one of rank $O(\log S)$. This shows that in SP the rank of refutations is a distinct complexity measure from the length.

In order to obtain time-space tradeoffs, de Rezende et al. [11] proved Resolution upper bounds on the lifted pebbling contradictions. Combining this upper bound (which can be simulated efficiently in SP) with the lower bound from Theorem 19 exhibits a formula that requires small size, but near-maximal rank to refute in SP.

▶ **Theorem 22.** SP *refutations cannot be balanced.*

Proof. Suppose that a SP refutation of length S implied the existence of a SP refutation of the same formula of rank $O(\log S)$. Let G be the graph from Theorem 19 on n vertices, and let Peb_G be the pebbling contradictions defined on this graph. It follows immediately from Lemmas 7.2 and 7.3 from de Rezende et al. [11] that for any graph of indegree 2 on n vertices, that there is a Resolution refutation of $\text{Peb}_G \circ \mathsf{IND}^n_\ell$ of length $O(n\ell^3)$. Since SP can p-simulate Resolution, this implies a $\mathsf{poly}(n)$ upper bound on the same formula in SP. Under the assumption that SP refutation can be balanced, this would imply a SP refutation of depth $O(\log n)$ of $\text{Peb}_G \circ \mathsf{IND}^n_\ell$, contradicting the lower bound from Corollary 19. ◀

Although it is a well-known fact that tree-like Cutting Planes refutations cannot be balanced, Impagliazzo et al. [20] show that the randomized communication protocols for the search problem obtained from CP refutations can be balanced. Using this fact, they show that a length S tree-like Cutting Planes proof implies a depth $O(\log S)$ protocol for the search problem. This implies that communication cost lower bounds for the search problem imply length lower bounds for tree-like Cutting Planes refutations.

Optimistically one could hope that a similar approach could be applied to SP refutations. This is reinforced by the fact that the real communication protocols for the search problem obtained from SP refutations (Lemma 17) maintains the same topology as the refutation. That is, the cost and size of the resulting protocol are approximately equal to the rank and length of the proof (unfortunately, this is not the case for the randomized protocols obtained from SP refutations). Therefore, one might hope that even though SP cannot be balanced, the corresponding real communication protocols for the search problem can be balanced. Thus, lower bounds on the rank of real-communication protocols for the search problem would imply lower bounds on the size of SP proofs.

▶ **Corollary 23.** *Any* SP *refutation of length S and rank r of an unsatisfiable formula $\mathcal{F}$ implies a real communication protocol of size $O(S \cdot n)$ and cost $O(r + \log n)$ for* $\mathsf{Search}_{X,Y}(\mathcal{F})$.

This follows from observing that the protocol obtained in Lemma 17 also preserves the topology (and therefore both the rank and the length) of the refutation.

5.4 Real Communication Protocols Cannot Be Balanced

Analogous to Theorem 22 (SP proofs cannot be balanced) in this section we will show that real communication protocols cannot be balanced. This should be contrasted with other standard models of communication such as randomized and deterministic, which can be balanced. In particular, we exhibit a function which has a real communication protocol of small size, but for which every real protocol must have high cost. Towards this end, we prove lower bounds on the real communication complexity of the famous NP^{cc}-complete problem set disjointness function[1]

The set disjointness function DISJ_n defined as follows: each player is given an n-bit string, interpreted as indicator vectors for an underlying set of n elements, and they are asked to determine whether their sets are disjoint. That is, the players aim to solve the function $\mathrm{DISJ}_n(x, y) = \bigvee_{i \in [n]} (x_i \wedge y_i)$. To our knowledge, the only known technique for obtaining lower bounds on the real communication of any problem are via lifting theorems, reducing the task of proving lower bounds on lifted functions to the decision tree complexity of the un-lifted function. Although DISJ_n can be seen as a lifted function (the OR_n function lifted by the two-bit AND gadget), these real communication lifting theorems work only for super-constant sized gadgets, and therefore cannot be applied directly to DISJ_n. We circumvent this difficulty by exploiting the fact that DISJ_n is NP^{cc}-complete. To do so, we find a lifted function in NP^{cc} to which our simulation theorems can be applied. Consider the n-bit OR_n function composed with the index gadget, $\mathrm{OR} \circ \mathsf{IND}_\ell^n$, for some ℓ defined later.

▶ **Lemma 24.** $\mathrm{OR}_n \circ \mathsf{IND}_\ell^n \in \mathsf{NP}^{cc}$, *for any* $\ell \leq 2^{polylog(n)}$.

Proof. First, observe that the index gadget $\mathsf{IND}_\ell(x_i, y_i)$, for a single bit i of the input to the OR_n function, can be computed by a brute force protocol in $\log \ell$ bits of communication. Alice simply sends to Bob the $\log \ell$ bits of her input $x_i = x_{i,1}, \ldots, x_{i,\log \ell}$. Bob is then able to evaluate $\mathsf{IND}_\ell(x_i, y_i)$.

Now, consider the following NP^{cc} protocol for $\mathrm{OR}_n \circ \mathsf{IND}_\ell^n$: Alice and Bob are given as a proof, a $\log n$-bit string indicating the index $i \in [n]$ of the OR_n function where $\mathsf{IND}_\ell(x_i, y_i) = 1$. They then perform the brute force protocol to evaluate $\mathsf{IND}_\ell(x_i, y_i)$ and verify that the outcome is indeed 1. In total, this requires $\log \ell + \log n + 1 = polylog(n)$ bits of NP^{cc}-communication. ◀

To obtain lower bounds on $\mathrm{OR}_n \circ \mathsf{IND}_\ell^n$, we appeal to the real communication simulation theorem (Theorem 18), reducing communication lower bounds for $\mathrm{OR}_n \circ \mathsf{IND}_\ell^n$ on the lifted problem to the well-known linear decision tree lower bounds on the OR_n function.

▶ **Lemma 25.** *Let $\ell = n^4$. The cost of any real communication protocol computing* $\mathrm{OR}_n \circ \mathsf{IND}_\ell^n$ *is $\Omega(n \log \ell)$.*

Proof. Combining the $\Omega(n)$ lower bound on the decision tree complexity of computing the OR_n function with the simulation theorem of de Rezende et al. [11] proves the result. ◀

[1] Recall that the NP^{cc} communication complexity of a function f is the minimum size of any monochromatic rectangle cover of the communication matrix of f.

▶ **Theorem 26.** *The cost of any real communication protocol for* DISJ_n *is* $\Omega((n \log n)^{1/5})$.

Proof. Let $\ell = n^4$. Consider the following reduction from $\mathrm{OR} \circ \mathsf{IND}_\ell^n$ to an instance of set disjointness. By Lemma 24, the NP^{cc}-complexity of $\mathrm{OR}_n \circ \mathsf{IND}_\ell^n$ is $\log \ell + \log n + 1$. That is, there exists a cover of the 1s of the communication matrix of $\mathrm{OR}_n \circ \mathsf{IND}_\ell^n$ with at most $2n\ell$ rectangles. Enumerating the 1-rectangles gives us an instance of set disjointness: on input (x, y) to $\mathrm{OR}_n \circ \mathsf{IND}_\ell^n$, Alice constructs the $2n\ell$-bit string which is the indicator vector $I_x(x)$ of the set of 1 rectangles which x belongs to, similarly Bob constructs $I_y(y)$ the same for y. Thus $\mathrm{OR}_n \circ \mathsf{IND}_\ell^n(x, y) = 1$ iff $\mathrm{DISJ}_{2n\ell}(I_x(x), I_y(y)) = 1$. Combining this with the lower bound from Lemma 25 gives a lower bound of $\Omega((n \log \ell)^{1/5})$ on the cost of any real communication protocol computing DISJ_n. ◀

▶ **Corollary 27.** *Real communication protocols cannot be balanced.*

Proof. Observe that $\mathrm{DISJ}_n = \vee_{i=1}^n (x_i \wedge y_i)$ can be computed in a $2n + 1$-node protocol by Alice and Bob comparing x_i and $2 - y_i$ for $i = 1, \ldots, n$. Suppose by contradiction that one could balance real communication protocols. The size $2n + 1$ protocol would therefore imply a cost $\log(2n + 1)$ real protocol for DISJ_n, contradicting the lower bound from Theorem 26. ◀

6 Conclusions

This paper introduces and develops the Stabbing Planes proof system as a natural extension of DPLL and pseudoBoolean solvers to handle a more expressive set of queries. Although it is equivalent to a tree-like version of a system already in the literature, this new perspective turns out to be quite useful for proving upper bounds. This paper is only a preliminary exploration of the SP proof system and leaves open many interesting problems from both a theoretical as well as a practical perspective.

As mentioned in the preliminaries, we do not have an analog to Cook et al. [8] for Stabbing Planes and so it is unknown whether for SP refutations, the length and size (number of bits) can be treated as the same measure. That is, is it possible to prove that any SP refutation of *length* l can be simulated by an SP planes refutation of size $\mathsf{poly}(l, n)$?

We have shown that that CP refutations of small rank can be simulated by SP refutation of small rank, and that CP refutations of small size can also be simulated by SP refutations of small size. Can we simulate both rank and size efficiently? That is, can any CP refutation of rank r and size s be simulated by a SP refutation of rank $\mathsf{poly}(r)$ and size $\mathsf{poly}(s)$?

Is it possible to prove superpolynomial lower bounds for SP? Krajíček [23] gave exponential lower bounds on the length of R(CP) refutations when both the width of the clauses, and the size of the coeficients appearing in the inequalities are sufficiently bounded. This was later improved by Kojevnikov [22] to remove the restriction on the size of the coefficients for tree-like R(CP). In particular, to obtain any lower bound at all, the width of the clauses appearing in the R(CP) refutations must be bounded by $o(n/\log n)$. From Theorem 9, a size S and rank D SP refutation implies an R(CP) proof of size $O(S)$ and width $O(D)$. Therefore, this result is also a size lower bound for bounded-depth SP. Unfortunately, it appears that these techniques are fundamentally limited to be applicable only to SP refutations with low depth, and so new techniques seem needed to overcome this barrier.

As mentioned in the introduction, we feel that SP has potential, in combination with state-of-the-art algorithms for SAT and ILP, for improved performance on some hard instances, and problems such as maxSAT and counting satisfying assignments. The upper bound on the Tseitin example illustrates the kind of reasoning that SP is capable of: arbitrarily splitting the

solution space into sub-problems based on some measure of progress. It would be interesting to build a SP-based solver, or to add SP-like branching to a solver such as CPLEX.

It has been a long-standing conjecture that CP does not have short refutations of the Tseitin formulas, as CP is unable to count mod2. On the other hand, Theorem 17 gives a quasi-polynomial upper bound on the Tseitin formulas in SP. Therefore, a natural approach to separating SP and CP is through proving CP lower bounds for Tseitin formulas.

References

1. Paul Beame, Noah Fleming, Russell Impagliazzo, Antonina Kolokolova, Denis Pankratov, Toniann Pitassi, and Robert Robere. Stabbing planes. *CoRR*, abs/1710.03219, 2017. `arXiv:1710.03219`.
2. Paul Beame, Trinh Huynh, and Toniann Pitassi. Hardness amplification in proof complexity. In *STOC'10*, pages 87–96, 2010.
3. Eli Ben-Sasson, Russell Impagliazzo, and Avi Wigderson. Near optimal separation of tree-like and general resolution. *Combinatorica*, 24(4):585–603, 2004.
4. Eli Ben-Sasson and Avi Wigderson. Short proofs are narrow - resolution made simple. *J. ACM*, 48(2):149–169, 2001.
5. Daniel Le Berre. Handling Pseudo-Boolean constraints in a CDCL solver: a practical survey. In *Dagstuhl Seminar 15171: Theory and Practice of SAT Solving*, April 2015.
6. Maria Luisa Bonet, Juan Luis Esteban, Nicola Galesi, and Jan Johannsen. On the relative complexity of resolution refinements and cutting planes proof systems. *SIAM J. Comput.*, 30(5):1462–1484, 2000.
7. Joshua Buresh-Oppenheim, Nicola Galesi, Shlomo Hoory, Avner Magen, and Toniann Pitassi. Rank bounds and integrality gaps for cutting planes procedures. *Theory of Computing*, 2(4):65–90, 2006.
8. William Cook, Collette Coullard, and György Turán. On the complexity of cutting-plane proofs. *Discrete Applied Mathematics*, 18(1):25–38, 1987.
9. Martin Davis, George Logemann, and Donald Loveland. A machine program for theorem-proving. *Commun. ACM*, 5(7):394–397, jul 1962.
10. Martin Davis and Hilary Putnam. A computing procedure for quantification theory. *J. ACM*, 7(3):201–215, 1960.
11. Susanna F. de Rezende, Jakob Nordstrom, and Marc Vinyals. How limited interaction hinders real communication (and what it means for proof and circuit complexity). In *FOCS'16*, pages 295–304, 2016.
12. Yuval Filmus, Pavel Hrubes, and Massimo Lauria. Semantic versus syntactic cutting planes. In *STACS'16*, pages 35:1–35:13, 2016.
13. Noah Fleming, Denis Pankratov, Toniann Pitassi, and Robert Robere. Random cnfs are hard for cutting planes. *Electronic Colloquium on Computational Complexity (ECCC)*, 24:45, 2017.
14. Mika Göös and Toniann Pitassi. Communication lower bounds via critical block sensitivity. In *STOC'14*, pages 847–856, 2014.
15. Mika Göös, Toniann Pitassi, and Thomas Watson. Deterministic communication vs. partition number. In *FOCS'15*, pages 1077–1088, 2015.
16. Pavel Hrubes and Pavel Pudlák. A note on monotone real circuits. *Electronic Colloquium on Computational Complexity (ECCC)*, 24:48, 2017.
17. Pavel Hrubeš and Pavel Pudlák. Random formulas, monotone circuits, and interpolation. *Electronic Colloquium on Computational Complexity (ECCC)*, 24:42, 2017.

18 Trinh Huynh and Jakob Nordstrom. On the virtue of succinct proofs: Amplifying communication complexity hardness to time-space trade-offs in proof complexity. In *STOC '12*, pages 233–248, 2012.

19 IBM ILOG. The CPLEX optimizer. URL: `https://www-01.ibm.com/software/commerce/optimization/cplex-optimizer/`.

20 Russell Impagliazzo, Toniann Pitassi, and Alasdair Urquhart. Upper and lower bounds for tree-like cutting planes proofs. In *LICS '94*, pages 220–228, 1994.

21 Stasys Jukna. *Boolean function complexity : advances and frontiers.* Algorithms and combinatorics. Springer, 2012.

22 Arist Kojevnikov. Improved lower bounds for tree-like resolution over linear inequalities. In *SAT'07*, pages 70–79, 2007.

23 Jan Krajícek. Discretely Ordered Modules as a First-Order Extension of the Cutting Planes Proof System. *The Journal of Symbolic Logic*, 63(4):1582–1596, 1998.

24 Jan Krajícek. Interpolation by a Game. *Mathematical Logic Quarterly*, 44:450–458, 1998.

25 Saburo Muroga. *Threshold logic and its applications.* Wiley-Interscience, 1972.

26 Noam Nisan. The communication complexity of threshold gates. In *In Proceedings of "Combinatorics, Paul Erdos is Eighty*, pages 301–315, 1994.

27 Pavel Pudlák. Lower bounds for resolution and cutting plane proofs and monotone computations. *Journal of Symbolic Logic*, 62(3):981–998, 1997.

28 Ran Raz and Pierre McKenzie. Separation of the monotone NC hierarchy. *Combinatorica*, 19(3):403–435, 1999.

29 Olivier Roussel and Vasco M Manquinho. Pseudo-Boolean and Cardinality Constraints. In *Handbook of satisfiability*, pages 695–733. IOS Press, 2009.

30 Günter M. Ziegler. *Lectures on Polytopes.* Springer-Verlag, New York, 1995.

A Candidate for a Strong Separation of Information and Communication

Mark Braverman[*1], **Anat Ganor**[†2], **Gillat Kol**[3], **and Ran Raz**[‡4]

1 **Department of Computer Science, Princeton University, USA**
2 **Department of Computer Science, Tel-Aviv University, Israel**
3 **Department of Computer Science, Princeton University, USA**
4 **Department of Computer Science, Princeton University, USA**

Abstract

The *weak interactive compression conjecture* asserts that any two-party communication protocol with communication complexity C and information complexity I can be compressed to a protocol with communication complexity $poly(I)polylog(C)$.

We describe a communication problem that is a candidate for refuting that conjecture. Specifically, while we show that the problem can be solved by a protocol with communication complexity C and information complexity $I = polylog(C)$, the problem seems to be hard for protocols with communication complexity $poly(I)polylog(C) = polylog(C)$.

1998 ACM Subject Classification F.2 Analysis of Algorithms and Problem Complexity

Keywords and phrases communication complexity, amortized communication complexity, communication compression, direct sum, information complexity

Digital Object Identifier 10.4230/LIPIcs.ITCS.2018.11

1 Introduction

The classical data compression theorem shows that every message can be compressed to its information content, measured using the entropy function. Can one prove a similar result in the interactive setting, where two parties engage in an interactive communication protocol? That is, can the transcript of every communication protocol be compressed to (roughly) its "information content" [2]?

The information content of an interactive protocol is typically measured using the *information complexity* measure [11, 16, 7, 1, 2]. In this paper we will mainly be interested in *internal* information complexity (a.k.a, information complexity and information cost). A related notion of *external* information complexity is also used in the literature. Roughly speaking, let π be a two-party communication protocol, and let μ be a distribution over the private inputs for the communicating parties. The (internal) information complexity of π over μ, denoted $\mathsf{IC}_\mu(\pi)$, is the number of information bits that the players learn about each other's input, when running the protocol π with inputs distributed according to μ (see Definition 3).

* Research supported in part by NSF Awards DMS-1128155, CCF- 1525342, and CCF-1149888, a Packard Fellowship in Science and Engineering, and the Simons Collaboration on Algorithms and Geometry.
† Research supported by the Israel Science Foundation (grant number 552/16) and the I-CORE Program of the planning and budgeting committee and The Israel Science Foundation (grant number 4/11).
‡ Research supported by the Simons Collaboration on Algorithms and Geometry and by the National Science Foundation grants No. CCF-1714779 and CCF-1412958.

Using the notion of information complexity, the above interactive compression problem can be formulated as asking whether for every protocol π and distribution μ with information complexity $I = IC_\mu(\pi)$, there exists a "compressed" protocol π' that produces (almost) the same output as π and has $CC_\mu^{avg}(\pi')$ close to I. Here, $CC_\mu^{avg}(\pi')$ stands for the *distributional communication complexity* of π' over μ, which is the expected number of bits communicated by π' when the inputs to the players are sampled according to μ (see Definition 2).

Several recent results show how to compress communication protocols in several cases, starting from [2] (see Section 2.3). However, none of these results gives a way of compressing a general protocol to a protocol that only communicates I bits, or even $poly(I)$ bits. We note that in some special cases, compression to $poly(I)polylog(C)$ or even $poly(I)$ are known to be possible (see Section 2.3).

The difficulty in compressing general protocols was recently explained by the authors, by proving exponential gaps between the distributional communication complexity and information complexity of some carefully designed communication tasks. In [8, 10], Ganor, Kol and Raz showed an explicit example of a boolean function with (internal) information complexity $\leq I$ and distributional communication complexity $\geq 2^{\Omega(I)}$ (see [17] for a simplified proof). In [9], Ganor, Kol and Raz analyzed a communication task proposed by Braverman [4], with (external) information complexity $\leq I$ and distributional communication complexity $\geq 2^{\Omega(I)}$.

One drawback of these results is that the protocols that achieve information complexity I have communication complexity double or even triple exponential in I. Therefore, while these results rule out "strong" compression to $poly(I)$, they leave open the possibility of "weak" compression to $poly(I)polylog(C)$.

▶ **Open Problem 1.** *Is it true that for every computational task f, distribution μ over the inputs and every communication protocol π that solves f with error $o(1)$, there exists a protocol π' that solves f with error $o(1)$, such that*

$$CC_\mu^{avg}(\pi') \leq poly(IC_\mu(\pi)) \cdot polylog(CC_\mu^{avg}(\pi))?$$

A general compression to $poly(I)polylog(C)$ as suggested by Problem 1, if exists, still yields very efficient compressed protocols that potentially constitute huge savings. Due to the equivalence between interactive compression and direct sum [2, 5], such a compression would also imply a near optimal *direct sum* result for distributional communication complexity, thus resolve this long standing open problem in the affirmative. Specifically, it will show that the distributional communication complexity of solving m independent copies of a communication task is almost as high as m times the distributional communication complexity of solving a single copy. Moreover, such an interactive compression result gives rise to a new paradigm for protocol design, where one is only mindful to the information revealed by the protocol, and then uses a compression scheme as a "black-box" to lower the required communication complexity.

In this work we suggest a candidate communication problem, called the *excited tree game*, for ruling out the $poly(I)polylog(C)$ compression scheme suggested by Problem 1. The game is defined in Section 3, and is parameterized by a parameter $c \in \mathbb{N}$. In Section 5, we construct a protocol for solving the game with information complexity $I = polylog(c)$ and communication complexity $C = O(c)$. In Section 4, we try to justify the conjecture that there is no protocol for solving the game with distributional communication complexity at most $poly(I)polylog(C) = polylog(c)$. Observe that this conjecture, if true, shows that the low information protocol we construct for the excited tree game cannot be compressed to $poly(I)polylog(C)$, thus answers Problem 1 in the negative. Proving this conjecture in full, however, seems very challenging.

2 Preliminaries

2.1 Communication Complexity

In the two player distributional model of communication complexity, each player gets an input, where the inputs are sampled from a joint distribution that is known to both players. The players' goal is to solve a computational task that depends on both inputs. The players can use both common and private random strings and are allowed to err with some small probability. The players communicate in rounds, where in each round one of the players sends a message to the other player. The communication complexity of a protocol is the total number of bits communicated by the two players. The communication complexity of a computational task is the minimum number of bits that the players need to communicate in order to solve the task with high probability, where the minimum is taken over all protocols. For excellent surveys on communication complexity see [14, 15]. In this work it would be more convenient to work with average communication complexity.

▶ **Definition 2 (Average Communication Complexity).** The *average communication complexity* of a protocol π over random inputs (X, Y) that are drawn according to a joint distribution μ, denoted $CC^{avg}_{\mu}(\pi)$, is the expected number of communication bits transmitted during the protocol, where the expectation is over (X, Y) and over the randomness. The ϵ average communication complexity of a computational task f with respect to a distribution μ is defined as

$$CC^{avg}_{\mu}(f, \epsilon) = \inf_{\pi} CC^{avg}_{\mu}(\pi),$$

where the infimum ranges over all protocols π that solve f with error at most ϵ on inputs that are sampled according to μ.

2.2 Information Complexity

Roughly speaking, the (internal) information complexity of a protocol is the number of information bits that the players learn about each other's input, when running the protocol. The information complexity of a communication task is the minimum number of information bits that the players learn about each other's input when solving the task, where the minimum is taken over all protocols. Formally,

▶ **Definition 3 (Information Cost).** The *information cost* of a protocol π over random inputs (X, Y) that are drawn according to a joint distribution μ, is defined as

$$IC_{\mu}(\pi) = \mathbf{I}(\Pi; X|Y) + \mathbf{I}(\Pi; Y|X),$$

where Π is a random variable which is the transcript of the protocol π with respect to μ. That is, Π is the concatenation of all the messages exchanged during the execution of π. The ϵ information cost of a computational task f with respect to a distribution μ is defined as

$$IC_{\mu}(f, \epsilon) = \inf_{\pi} IC_{\mu}(\pi),$$

where the infimum ranges over all protocols π that solve f with error at most ϵ on inputs that are sampled according to μ.

2.3 Known Compression Protocols

Several beautiful recent results show how to compress communication protocols in several cases. Barak, Braverman, Chen and Rao showed how to compress any protocol with information complexity I and communication complexity C, to a protocol with communication complexity $\sqrt{I \cdot C} \cdot polylog(C)$ [2]. They also suggest a protocol that communicates $I^{ext} \cdot polylog(C)$ bits, where I^{ext} is the external information complexity of the original protocol. Braverman and Rao showed how to compress any one round (or constant number of rounds) protocol with information complexity I to a protocol with communication complexity $O(I)$ [5]. Braverman showed how to compress any protocol with information complexity I to a protocol with communication complexity $2^{O(I)}$ [3] (see also [6, 12]). Building over [2], Kol and Sherstov showed how to compress any protocol with information complexity I to a protocol with communication complexity $I \cdot polylog(I)$ in the case where the underlying distribution is a product distribution [13, 18].

3 The Excited Tree Game

The excited tree game is a communication game for two players A and B. The game is played on a rooted, complete, binary tree $\mathcal{T}$, of depth c, where c is larger than a sufficiently large constant. Player A "owns" every non-leaf vertex in even layers and player B "owns" every non-leaf vertex in odd layers. For every non-leaf vertex v, the owner of v gets as an input a distribution $P_v = (p_v, 1-p_v)$ and the other player gets as an input a distribution $Q_v = (q_v, 1-q_v)$, both distributions are over the children of v. We think of every P_v as the "correct" distribution over the two children of v. The distributions $\{P_v, Q_v\}_v$ are chosen in a very specific way that is described below.

A *frontier* in the tree is a set of vertices that contains exactly one vertex (leaf or non-leaf) on every path from the root to a leaf. Given a vertex v and a frontier S in the tree, we say that v is *above* the frontier S if on the path from the root to v there is no vertex in S. We say that v is *on* the frontier if v is in S. If v is neither above the frontier nor on it, then it is *below* the frontier.

We denote by x, y the inputs to the players A, B respectively. That is, x is the set of all the distributions P_v or Q_v that are given to player A and y is the set of all the distributions P_v or Q_v that are given to player B. We define the distribution μ on the inputs to the players by an algorithm for sampling an input pair (x, y) (Algorithm 1 below).

Fix some $k = polylog(c)$ such that $\log^4(c) \leq k$. Let μ_1 be the uniform distribution over the interval $[-\frac{k}{\sqrt{c}}, \frac{k}{\sqrt{c}}]$ and let μ_2 be the uniform distribution over the interval $[-\frac{1}{\sqrt[8]{c}}, \frac{1}{\sqrt[8]{c}}]$.[1] In Algorithm 1 below, we sample for every non-leaf vertex v two values $x_1(v), x_2(v)$ according to μ_1, μ_2 respectively. Next, when we say "set v to be non-excited", we mean "set $p_v = \frac{1}{2} + x_1(v) + x_2(v)$ and $q_v = \frac{1}{2} + x_2(v) - x_1(v)$". By "set v to be excited", we mean "set $p_v = \frac{1}{2} + x_1(v) + x_2(v)$ and $q_v = \frac{1}{2} + x_1(v) - x_2(v)$". Note that without communication, none of the players can distinguish between an excited vertex and a non-excited vertex, since p_v and q_v have the same distribution in both cases.

Given the distributions P_v for every non-leaf vertex v and the frontier S in the tree, we define a distribution P_S over the vertices in S. For every vertex $w \in S$, let $v_0, v_1, ..., v_\ell$ be

[1] One can also consider other symmetric distributions in the range $[-1, 1]$ with expectation 0 and variance $O(\frac{k^2}{c}), O(\frac{1}{\sqrt[4]{c}})$ respectively, as well as other values for k, and changing the interval $[-\frac{1}{\sqrt[8]{c}}, \frac{1}{\sqrt[8]{c}}]$ to $[-\frac{1}{c^\beta}, \frac{1}{c^\beta}]$, for some other $0 < \beta < 1/2$.

Algorithm 1 Sample (x, y) according to μ

1. For every non-leaf vertex v we sample two values $x_1(v), x_2(v)$ according to μ_1, μ_2 respectively.
2. Let S be a frontier in the tree defined as follows: Pick every vertex to be in S, independently, with probability $\alpha = \frac{k}{c}$. Then, for every path from the root to a leaf, remove from S all vertices on that path, except for the vertex closest to the root, if such a vertex exists. If there is no vertex in S on a path from the root to a leaf, add that leaf to S.
3. Set every non-leaf vertex above the frontier S to be non-excited.
4. Set every non-leaf vertex on the frontier S or below it to be excited.

the vertices on the path in the tree from the root to w, where ℓ is the layer of w. That is, v_0 is the root, $v_\ell = w$ and for every $0 \leq i < \ell$, v_{i+1} is a child of v_i. Then, $P_S(w)$ is obtained by sampling every child on the path to w according to the correct distribution of its parent. That is,

$$P_S(w) = \prod_{i=0}^{\ell-1} P_{v_i}(v_{i+1}) \text{ where } P_{v_i}(v_{i+1}) = \begin{cases} p_{v_i} & \text{if } v_{i+1} \text{ is the left hand child of } v_i \\ 1 - p_{v_i} & \text{if } v_{i+1} \text{ is the right hand child of } v_i \end{cases}.$$

The players' mutual goal is to output the same vertex w on the frontier S, where S is the frontier defined in Algorithm 1, such that for almost all possible outputs w, the probability that they both output w is close to $P_S(w)$. More precisely, let x, y be the inputs to the players A, B respectively and let μ be the distribution over the inputs. Let $A(x, y), B(x, y)$ denote the output values of A, B respectively. Note that $A(x, y), B(x, y)$ are random variables that depend on the randomness. For a communication protocol π, we say that π solves the game with respect to μ with error ϵ if

$$\Pr[A(x, y) = B(x, y)] \geq 1 - \epsilon \quad \text{and} \quad \mathbb{E}\left[||A(x, y) - P_S||_1\right] \leq \epsilon,$$

where the probability is over inputs that are sampled according to μ and over the randomness, the expectation is over inputs that are sampled according to μ and $||\cdot||_1$ is the ℓ_1 norm. Note that $A(x, y)$ is referred to as a distribution as well as a random variable and that the distribution P_S depends on x and y.

In Section 5 we prove the following lemma.

▶ **Lemma 4.** *There exists a protocol that solves the excited tree game with respect to μ with error $o(1)$, with average communication complexity $O(c)$ and information complexity $polylog(c)$.*

Therefore, to answer Open Problem 1 in the negative, it is enough to answer the following question affirmatively.

▶ **Open Problem 5.** *Is it true that for $\epsilon = o(1)$ the ϵ average communication complexity of the excited tree game with respect to μ is at least $(\log(c))^{\omega(1)}$?*

4 Why Excited Tree?

At a very high level, the excited tree game can be viewed as follows. The game is played on a rooted, complete, binary tree $\mathcal{T}$, of depth c. A frontier S is chosen in $\mathcal{T}$. All the vertices above the frontier are set to be "non-excited" and the vertices below the frontier are set to

be "excited". The player's goal is to output a vertex on the frontier S, sampled according to the "correct" distribution.

Let us start with the rational behind the name *excited tree game*. In physics, an "excited" state is a state with a higher energy level than the ground ("non-excited") state. In the excited tree game, an excited vertex is a vertex with a higher "information level" than a non-excited vertex. For an excited vertex v the distance between the distributions P_v and Q_v is large and hence the information that the player who doesn't own v is missing is relatively large. For a non-excited vertex v the distance between the distributions P_v and Q_v is small and hence the information that the player who doesn't own v is missing is relatively small.

Since all the vertices above the frontier are non-excited, there is a relatively simple protocol with low information complexity for the excited tree game: Starting from the root, until reaching the frontier, at every vertex v, the player owning v samples a child of v according to P_v and sends a bit b_v to the other player, to indicate which child was sampled. Both players continue to the child of v that is indicated by the communicated bit. Since all the vertices above the frontier are non-excited, the information given by each bit b_v is small and hence the entire information complexity of the protocol is small. The only complication in this protocol is that the players have to stop when they reach the frontier. We show how to do that while keeping the information complexity of the protocol low.

To answer Open Problem 5 affirmatively, one needs to prove a lower bound of $(\log(c))^{\omega(1)}$ on the communication complexity of the excited tree game. While we don't have such a proof, we note that several approaches to solve the game with communication complexity $(\log(c))^{O(1)}$, seem to fail.

Two properties of the excited tree game that makes it difficult (or impossible...) to solve with low communication complexity are as follows:

1. Without communication, none of the players can distinguish between an excited vertex and a non-excited vertex, since p_v and q_v have the same distribution in both cases. Hence, without communication (or with relatively small communication) the players don't have a lot of information about which vertices are above the frontier and which vertices are below the frontier.
2. For every vertex v above the frontier, the restriction of the inputs of the two players to the subtree below v (conditioned on the event that v is above the frontier) has the same distribution as the distribution of the excited tree game played on a smaller tree. In fact, we could have defined the problem on an *infinite*, rooted, complete binary tree, and then the distribution of the restriction to the subtree below v (conditioned on the event that v is above the frontier) would have been exactly the same as the original distribution. (We chose to work with a finite tree for simplicity of the presentation).

In light of these properties, let us consider a few approaches for designing protocols with low communication complexity for solving the problem, based on known approaches for compression protocol.

A first approach (inspired by ideas initiated in [2] and used in many subsequent works) could be to try to simulate the above mentioned low-information protocol, by starting from the root and trying to sample, according to the correct distribution, vertices that are lower and lower in the tree, until reaching the frontier. A major difficulty with such attempts is the second property above. By the second property, even if the two players managed to agree on a vertex v above the frontier, sampled according to the correct distribution, they still have to solve a copy of pretty much the same problem as the one that they started with, and hence they made no (or very little) progress. The two players only make progress if the vertex v that they agreed on happens to be exactly on the frontier.

A second approach (inspired by ideas initiated in [2, 5, 3] and used in many subsequent works) could be to sample a leaf in the tree (or a vertex which is with high probability below the frontier) and climb up from that vertex to the frontier. A major difficulty with such attempts is that all the vertices below the frontier are excited, and hence the distributions that the two players have on the leaves are very far from each other, so it's hard for them to agree on a leaf. They could agree on a leaf sampled according to a pre-agreed distribution, known to both players, such as the uniform distribution, and climb up from that leaf to the frontier. However, that would not sample a frontier vertex according to the correct distribution. In general, the first property above (that the players don't know where the frontier is) makes such attempts very difficult.

A third approach that one may consider for attacking this and related problems (and that, to the best of our knowledge, has not been used before), is to try to sample a vertex v above the frontier (as in the first approach) and from that vertex to move down to the closest frontier vertex u in the subtree below v. This approach is based on the fact that there should be a frontier vertex at a distance of roughly $\log(\alpha^{-1}) = O(\log(c))$ below v. A difficulty with such attempts is that it is not clear how to find the closest frontier vertex u by a protocol with small communication complexity.

We note that turning these intuitions and ideas into a full proof for a lower bound on the communication complexity of the excited tree game seems very challenging.

5 Information Upper Bound

In this section, we prove Lemma 4. Let $(x, y) \in \mathrm{supp}(\mu)$ be an input pair for the excited tree game and let S be the frontier defined in Algorithm 1. Let π be the following protocol for the excited tree game, played on the input pair (x, y): Starting from the root, at every vertex v, the player owning v samples a child of v according to P_v and sends a bit b_v to the other player, to indicate which child was sampled. Both players continue to the child of v that is indicated by the communicated bit.

After receiving a bit b_v, the receiving party, without loss of generality the second player, sends a bit a_v, that supposedly indicates whether the players are above the frontier S or not, where $a_v = 1$ stands for "below or on the frontier" and $a_v = 0$ stands for "above the frontier". If v is a leaf, the second player sends $a_v = 1$. Otherwise, to determine the value of a_v, the second player considers the last $\ell = 4k\sqrt[4]{c}$ vertices $v_1, \ldots, v_\ell$ reached by the protocol and owned by the first player and the corresponding bits $b_{v_1}, \ldots, b_{v_\ell}$ that were sent by the first player (if less than ℓ bits were sent by the first player so far, the second player sends $a_v = 0$). For every $j \in [\ell]$, the second player compares b_{v_j} and $\tilde{q_{v_j}}$, where $\tilde{q_{v_j}}$ is 1 if $q_{v_j} \geq \frac{1}{2}$ and 0 otherwise. The second player sends $a_v = 1$ if less than $\frac{\ell}{2}$ of these pairs are equal, and otherwise, he sends $a_v = 0$.

Once the bit $a_v = 1$ was sent, the players run a binary search over the last 3ℓ vertices reached by the protocol, with the goal of finding the vertex on the frontier (if less than 3ℓ vertices were reached by the protocol so far, the binary search is over all the vertices reached by the protocol). In each iteration of the binary search, the players send their input distributions corresponding to the current vertex considered by the binary search. The probabilities are truncated so that each player sends k bits per vertex. For each such vertex v, the players calculate $|p'_v - q'_v|$, where p'_v, q'_v are the truncated p_v, q_v respectively. The binary search assumes that $|p'_u - q'_u| \leq \frac{3k}{\sqrt{c}}$ for all the vertices u among these 3ℓ vertices that are above the frontier, and that $|p'_u - q'_u| > \frac{3k}{\sqrt{c}}$ for all the vertices u among these 3ℓ vertices that are below the frontier. Under this assumption, the players output the vertex v which is the first vertex among these 3ℓ vertices for which $|p'_v - q'_v| > \frac{3k}{\sqrt{c}}$, if such a vertex exists. (Otherwise, the players output an error message).

5.1 Bounding the Error Probability

In Claims 6 and 7 we prove that with high probability, the bit $a_v = 1$ is sent below or on the frontier, but not too far below it. In Claim 8 we prove that if this is the case, then with high probability, the players output the vertex on the frontier reached by the protocol. Note that each vertex reached by the protocol is chosen according to the correct distribution. That is, if a vertex w was reached by the protocol, then when the players reached its parent v, they sampled w according to the distribution P_v. Therefore, Claims 6, 7 and 8 imply that the distribution of the vertex output by the players is close to the goal distribution P_S.

▶ **Claim 6.** *Let $(x, y) \in \mathrm{supp}(\mu)$ be an input pair for the excited tree game and let S be the frontier defined in Algorithm 1. Then, with probability at least $1 - c \cdot 2^{-k/2}$ over the input pair (x, y) and over the randomness, the bit $a_v = 1$ is sent when the players are below or on the frontier S.*

Proof. Let v be a non-excited vertex. First, consider the case that $p_v \geq \frac{1}{2}$. In this case,

$$\begin{aligned}
\Pr\left[q_v < \tfrac{1}{2} \mid p_v \geq \tfrac{1}{2}\right] &= \Pr\left[x_2(v) - x_1(v) < 0 \mid x_1(v) + x_2(v) \geq 0\right] \\
&= \Pr\left[x_2(v) < x_1(v) \mid x_2(v) \geq -x_1(v)\right] \\
&= \Pr\left[(x_2(v) < x_1(v)) \wedge (x_2(v) \geq -x_1(v))\right] \cdot \left(\Pr\left[x_2(v) \geq -x_1(v)\right]\right)^{-1} \\
&= 2\Pr\left[-x_1(v) \leq x_2(v) < x_1(v)\right] \\
&\leq 2\Pr\left[-\frac{k}{\sqrt{c}} \leq x_2(v) \leq \frac{k}{\sqrt{c}}\right] = \frac{2k}{c^{3/8}}.
\end{aligned}$$

Therefore, with high probability, $q_v \geq \frac{1}{2}$ and $\tilde{q_v} = 1$. It holds that

$$\begin{aligned}
\Pr\left[b_v = \tilde{q_v} \mid \left(p_v \geq \tfrac{1}{2}\right) \wedge \left(q_v \geq \tfrac{1}{2}\right)\right] &= \Pr\left[b_v = 1 \mid \left(p_v \geq \tfrac{1}{2}\right) \wedge \left(q_v \geq \tfrac{1}{2}\right)\right] \\
&= \mathbf{E}\left[p_v \mid \left(p_v \geq \tfrac{1}{2}\right) \wedge \left(q_v \geq \tfrac{1}{2}\right)\right],
\end{aligned}$$

where the last equality holds since the probability is over the inputs and over the randomness. Bounding the expectation we get that

$$\begin{aligned}
\mathbf{E}\left[p_v \mid \left(p_v \geq \tfrac{1}{2}\right) \wedge \left(q_v \geq \tfrac{1}{2}\right)\right] &= \mathbf{E}\left[p_v \mid (x_2(v) \geq -x_1(v)) \wedge (x_2(v) \geq x_1(v))\right] \\
&= \mathbf{E}\left[p_v \mid x_2(v) \geq |x_1(v)|\right] \\
&= \frac{1}{2} + \mathbf{E}\left[x_1(v) \mid x_2(v) \geq |x_1(v)|\right] + \mathbf{E}\left[x_2(v) \mid x_2(v) \geq |x_1(v)|\right] \\
&\geq \frac{1}{2} - \frac{k}{\sqrt{c}} + \mathbf{E}\left[x_2(v) \mid x_2(v) \geq |x_1(v)|\right] \\
&\geq \frac{1}{2} - \frac{k}{\sqrt{c}} + \mathbf{E}\left[x_2(v) \mid x_2(v) \geq 0\right] = \frac{1}{2} + \frac{1}{2c^{1/8}} - \frac{k}{\sqrt{c}}.
\end{aligned}$$

Similarly, when $p_v < \frac{1}{2}$, the probability that $q_v \geq \frac{1}{2}$ is at most $\frac{2k}{c^{3/8}}$. Therefore, with high probability $q_v < \frac{1}{2}$ and

$$\begin{aligned}
\Pr\left[b_v = \tilde{q_v} \mid \left(p_v < \tfrac{1}{2}\right) \wedge \left(q_v < \tfrac{1}{2}\right)\right] &= \mathbf{E}\left[1 - p_v \mid \left(p_v < \tfrac{1}{2}\right) \wedge \left(q_v < \tfrac{1}{2}\right)\right] \\
&\geq \frac{1}{2} - \frac{k}{\sqrt{c}} - \mathbf{E}\left[x_2(v) \mid x_2(v) < -|x_1(v)|\right] \\
&\geq \frac{1}{2} + \frac{1}{2c^{1/8}} - \frac{k}{\sqrt{c}}.
\end{aligned}$$

Put together, we get that for a non-excited vertex v, the probability that $b_v = \tilde{q_v}$ is at least

$$\left(1 - \frac{2k}{c^{3/8}}\right) \cdot \left(\frac{1}{2} + \frac{1}{2c^{1/8}} - \frac{k}{\sqrt{c}}\right) \geq \frac{1}{2} + \frac{1}{4c^{1/8}}.$$

When the players are above the frontier, all the vertices reached by the protocol are non-excited. If a player considers ℓ non-excited vertices $v_1, \ldots, v_\ell$ and their corresponding bits $b_{v_1}, \ldots, b_{v_\ell}$, then by Chernoff, the probability that less than $\frac{\ell}{2}$ of the pairs $b_{v_j}, \tilde{q_{v_j}}$ are equal is at most $e^{-2\ell/16c^{1/4}} \leq 2^{-k/2}$. That is, for every vertex v above the frontier, the probability that the bit $a_v = 1$ is sent is at most $2^{-k/2}$. Thus, by the union bound, the total probability that a bit $a_v = 1$ is sent above the frontier is at most $c \cdot 2^{-k/2}$. ◀

▶ **Claim 7.** *Let $(x, y) \in \mathrm{supp}(\mu)$ be an input pair for the excited tree game and let S be the frontier defined in Algorithm 1. Assume that the players are below or on the frontier. Then, with probability at least $1 - 2^{-k/2}$ over the input pair (x, y) and over the randomness, a player will send the bit $a_v = 1$ after at most 2ℓ steps.*

Proof. Let v be an excited vertex. First, consider the case that $p_v \geq \frac{1}{2}$. In this case,

$$\begin{aligned}
\Pr\left[q_v \geq \tfrac{1}{2} \mid p_v \geq \tfrac{1}{2}\right] &= \Pr\left[x_1(v) - x_2(v) \geq 0 \mid x_1(v) + x_2(v) \geq 0\right] \\
&= \Pr\left[x_1(v) \geq x_2(v) \mid x_2(v) \geq -x_1(v)\right] \\
&= \Pr\left[(x_1(v) \geq x_2(v)) \wedge (x_2(v) \geq -x_1(v))\right] \cdot \left(\Pr\left[x_2(v) \geq -x_1(v)\right]\right)^{-1} \\
&= 2\Pr\left[-x_1(v) \leq x_2(v) \leq x_1(v)\right] \\
&\leq 2\Pr\left[-\frac{k}{\sqrt{c}} \leq x_2(v) \leq \frac{k}{\sqrt{c}}\right] = \frac{2k}{c^{3/8}}.
\end{aligned}$$

Therefore, with high probability, $q_v < \frac{1}{2}$ and $\tilde{q_v} = 0$. It holds that

$$\begin{aligned}
\Pr\left[b_v \neq \tilde{q_v} \mid \left(p_v \geq \tfrac{1}{2}\right) \wedge \left(q_v < \tfrac{1}{2}\right)\right] &= \Pr\left[b_v = 1 \mid \left(p_v \geq \tfrac{1}{2}\right) \wedge \left(q_v < \tfrac{1}{2}\right)\right] \\
&= \mathbf{E}\left[p_v \mid \left(p_v \geq \tfrac{1}{2}\right) \wedge \left(q_v < \tfrac{1}{2}\right)\right],
\end{aligned}$$

where the last equality holds since the probability is over the inputs and over the randomness. Bounding the expectation we get that

$$\begin{aligned}
\mathbf{E}\left[p_v \mid \left(p_v \geq \tfrac{1}{2}\right) \wedge \left(q_v < \tfrac{1}{2}\right)\right] &= \mathbf{E}\left[p_v \mid (x_2(v) \geq -x_1(v)) \wedge (x_2(v) > x_1(v))\right] \\
&\geq \frac{1}{2} - \frac{k}{\sqrt{c}} + \mathbf{E}\left[x_2(v) \mid x_2(v) \geq |x_1(v)|\right] \\
&\geq \frac{1}{2} - \frac{k}{\sqrt{c}} + \mathbf{E}\left[x_2(v) \mid x_2(v) \geq 0\right] = \frac{1}{2} + \frac{1}{2c^{1/8}} - \frac{k}{\sqrt{c}}.
\end{aligned}$$

Similarly, when $p_v < \frac{1}{2}$, the probability that $q_v < \frac{1}{2}$ is at most $\frac{2k}{c^{3/8}}$. Therefore, with high probability $q_v \geq \frac{1}{2}$ and

$$\begin{aligned}
\Pr\left[b_v \neq \tilde{q_v} \mid \left(p_v < \tfrac{1}{2}\right) \wedge \left(q_v \geq \tfrac{1}{2}\right)\right] &= \mathbf{E}\left[1 - p_v \mid \left(p_v < \tfrac{1}{2}\right) \wedge \left(q_v \geq \tfrac{1}{2}\right)\right] \\
&\geq \frac{1}{2} - \frac{k}{\sqrt{c}} - \mathbf{E}\left[x_2(v) \mid x_2(v) \leq -|x_1(v)|\right] \\
&\geq \frac{1}{2} + \frac{1}{2c^{1/8}} - \frac{k}{\sqrt{c}}.
\end{aligned}$$

Put together, we get that for an excited vertex v, the probability that $b_v \neq \tilde{q_v}$ is at least

$$\left(1 - \frac{2k}{c^{3/8}}\right) \cdot \left(\frac{1}{2} + \frac{1}{2c^{1/8}} - \frac{k}{\sqrt{c}}\right) \geq \frac{1}{2} + \frac{1}{4c^{1/8}}.$$

If the players take 2ℓ steps after they reached an excited vertex, then the player who should send either $a_v = 0$ or $a_v = 1$ considers ℓ excited vertices $v_1, \ldots, v_\ell$ and their corresponding bits $b_{v_1}, \ldots, b_{v_\ell}$. By Chernoff, the probability that less than $\frac{\ell}{2}$ of the pairs $b_{v_j}, \tilde{q_{v_j}}$ are not equal is at most $e^{-2\ell/16c^{1/4}} \leq 2^{-k/2}$. That is, the probability that the player sends $a_v = 0$ is at most $2^{-k/2}$. ◀

▶ **Claim 8.** *Let $(x, y) \in \text{supp}(\mu)$ be an input pair for the excited tree game and let S be the frontier defined in Algorithm 1. Assume that the bit $a_v = 1$ is sent when the players are below or on the frontier S and not more than 2ℓ steps after the players reached a vertex on S. Let w be the vertex on S that they reached. Then, with probability at least $1 - \frac{48k^2}{c^{1/8}}$ over the input pair (x, y) and over the randomness, the players output the vertex w.*

Proof. Let v be a non-excited vertex. Recall that p'_v, q'_v are the truncated probabilities p_v, q_v respectively. It holds that $|p'_v - q'_v| \leq 2|x_1(v)| + 2^{-k}$, which is at most $\frac{3k}{\sqrt{c}}$ (with probability 1). For an excited vertex v, it holds that $|p'_v - q'_v| \geq 2|x_2(v)| - 2^{-k}$. The probability that $2|x_2(v)| - 2^{-k}$ is at most $\frac{3k}{\sqrt{c}}$ is less than $\frac{4k}{c^{3/8}}$. Taking a union bound over the 3ℓ vertices considered by the binary search, we get that with probability of at least $1 - \frac{48k^2}{c^{1/8}}$, for all the excited vertices u among the 3ℓ vertices considered by the binary search, we have that $|p'_u - q'_u| > \frac{3k}{\sqrt{c}}$.

Thus, with probability of at least $1 - \frac{48k^2}{c^{1/8}}$, we have that $|p'_u - q'_u| \leq \frac{3k}{\sqrt{c}}$ for all the vertices u among the 3ℓ vertices considered by the binary search, that are above the frontier, and $|p'_u - q'_u| > \frac{3k}{\sqrt{c}}$ for all the vertices u among these 3ℓ vertices that are below the frontier. Under this assumption, the binary search outputs the vertex v which is the first vertex among these 3ℓ vertices for which $|p'_v - q'_v| > \frac{3k}{\sqrt{c}}$. Therefore, with probability at least $1 - \frac{48k^2}{c^{1/8}}$, the players output the vertex w. ◀

5.2 Bounding the Information Cost

To upper bound the information cost of the protocol π we will use the method described in [8], that is based on the notion of divergence cost of a tree [2, 5].

▶ **Definition 9 (Relative Entropy).** Let $\phi_1, \phi_2 : \Omega \to [0, 1]$ be two distributions, where Ω is finite. The *relative entropy* between ϕ_1 and ϕ_2, denoted $\mathbf{D}(\phi_1 \| \phi_2)$, is defined as

$$\mathbf{D}(\phi_1 \| \phi_2) = \sum_{x \in \Omega} \phi_1(x) \log \left(\frac{\phi_1(x)}{\phi_2(x)} \right).$$

▶ **Definition 10 (Divergence Cost [2, 5]).** Consider a binary tree $\mathcal{T}$ whose root is r and distributions $P_v = (p_v, 1 - p_v), Q_v = (q_v, 1 - q_v)$ for every non-leaf vertex v in the tree. We think of P_v and Q_v as distributions over the two children of the vertex v. We define the *divergence cost* of the tree $\mathcal{T}$ recursively, as follows. $\mathbf{D}(\mathcal{T}) = 0$ if the tree has depth 0, otherwise,

$$\mathbf{D}(\mathcal{T}) = \mathbf{D}(P_r \| Q_r) + \mathop{\mathbf{E}}_{v \sim P_r} [\mathbf{D}(\mathcal{T}_v)], \tag{1}$$

where for every vertex v, $\mathcal{T}_v$ is the subtree of $\mathcal{T}$ whose root is v.

An equivalent definition of the divergence cost of $\mathcal{T}$ is obtained by following the recursion in Equation (1) and is given by the following equation:

$$\mathbf{D}(\mathcal{T}) = \sum_{v \in V} \tilde{p}_v \cdot \mathbf{D}(P_v \| Q_v), \tag{2}$$

where V is the vertex set of $\mathcal{T}$ and for a vertex $v \in V$, $\tilde{p}_v$ is the probability to reach v by following the distributions P_v, starting from the root. Formally, if v is the root of the tree $\mathcal{T}$, then $\tilde{p}_v = 1$, otherwise,

$$\tilde{p}_v = \begin{cases} \tilde{p}_u \cdot p_u & \text{if } v \text{ is the left-hand child of } u \\ \tilde{p}_u \cdot (1 - p_u) & \text{if } v \text{ is the right-hand child of } u. \end{cases}$$

We will bound the information cost of the protocol until the bit $a_v = 1$ is sent, that is, until the players decide that they are below or on the frontier. Denote the protocol that starts as π but ends when the bit $a_v = 1$ is sent by π'. Note that after the bit $a_v = 1$ is sent in π, the players exchange at most $O(k \cdot \log(\ell))$ bits, which adds at most $O(k \cdot \log(\ell))$ bits of information.

We denote by $\mathcal{T}_{\pi'}$ the binary tree associated with π'. That is, every vertex v of $\mathcal{T}_{\pi'}$ corresponds to a possible transcript of π' and the two edges going out of v are labeled by 0 and 1, corresponding to the next bit to be transmitted. The vertices of the tree $\mathcal{T}_{\pi'}$ have the following structure: Every vertex v of $\mathcal{T}_{\pi'}$ corresponds to a vertex $\tilde{v}$ of $\mathcal{T}$, the binary tree on which the excited tree game is played. For a vertex v in an odd layer of $\mathcal{T}_{\pi'}$, the next bit to be transmitted by π' on the vertex v is $b_{\tilde{v}}$. For a vertex v in an even layer of $\mathcal{T}_{\pi'}$, the next bit to be transmitted by π' on the vertex v is $a_{\tilde{v}}$.

Every input pair $(x, y) \in \text{supp}(\mu)$ for the excited tree game, induces a distribution $P_v = (p_v, 1 - p_v)$ for every vertex v of the tree $\mathcal{T}_{\pi'}$, where p_v is the probability that the next bit transmitted by the protocol π' on the vertex v and inputs x, y is 0. Namely, if v is in an odd layer of $\mathcal{T}_{\pi'}$, the distribution P_v is the input distribution $P_{\tilde{v}}$ of the player that owns $\tilde{v}$. If v is in an even layer of $\mathcal{T}_{\pi'}$ then $P_v = (1, 0)$ when the player sending $a_{\tilde{v}}$ decides that the players are above the frontier and $P_v = (0, 1)$ when $a_{\tilde{v}} = 1$ is sent (note that given x, y and v this decision is deterministic).

For every vertex v of $\mathcal{T}_{\pi'}$, we define an additional distribution $Q_v = (q_v, 1-q_v)$ (depending on the input pair (x, y)). For a vertex v in an odd layer of $\mathcal{T}_{\pi'}$, the distribution Q_v is the input distribution $Q_{\tilde{v}}$ of the player that doesn't own $\tilde{v}$. If v is in an even layer of $\mathcal{T}_{\pi'}$ then $Q_v = (1 - \frac{1}{c}, \frac{1}{c})$.

For the rest of the section, we think of $\mathcal{T}_{\pi'}$ as the tree $\mathcal{T}_{\pi'}$ together with the distributions P_v and Q_v, for every vertex v in the tree $\mathcal{T}_{\pi'}$. In [8], Ganor, Kol and Raz showed that $IC_\mu(\pi') \leq \mathbf{E}[\mathbf{D}(\mathcal{T}_{\pi'})]$, where $\mathbf{D}(\mathcal{T}_{\pi'})$ is the divergence cost of the tree and the expectation is over the sampling of the inputs according to μ and over the randomness. Together with the following claim, we get that $IC_\mu(\pi') \leq O(k^2)$.

▶ **Claim 11.** *Let π' be the protocol that starts as π but ends when the bit $a_v = 1$ is sent. Let $\mathcal{T}_{\pi'}$ be the binary tree associated with π', together with the distributions P_v and Q_v for every vertex v in the tree $\mathcal{T}_{\pi'}$, as defined above. Then,*

$$\mathbf{E}[\mathbf{D}(\mathcal{T}_{\pi'})] = O(k^2),$$

where the expectation is over the inputs and over the randomness.

Proof. We bound the divergence cost separately for vertices in odd layers and for vertices in even layers. First, we sum over vertices in even layers. For every vertex v in an even layer of $\mathcal{T}_{\pi'}$, if $P_v = (1, 0)$ then $\mathbf{D}(P_v \| Q_v) = \log\left(\frac{1}{1-\frac{1}{c}}\right) = \log\left(1 + \frac{1}{c-1}\right) < \frac{2}{c}$. Since there are at most c such vertices on every path and the probability of reaching each vertex is at most 1, the sum in Equation (2) taken over vertices in even layers with $P_v = (1, 0)$ is at most $c \cdot \frac{2}{c} = 2$. If $P_v = (0, 1)$ then $\mathbf{D}(P_v \| Q_v) = \log\left(\frac{1}{\frac{1}{c}}\right) = \log(c) \leq O(k)$. Along each path, there is only one vertex v for which $P_v = (0, 1)$, the last vertex reached by the protocol π'.

Next, we sum over vertices in odd layers along an average path. Recall that each such vertex v corresponds to a vertex $\tilde{v}$ in $\mathcal{T}$. Let v be a vertex in an odd layer of $\mathcal{T}_{\pi'}$. It holds

that $|p_v - q_v| \leq \frac{1}{4}$ and $p_v \geq \frac{5}{16}$, and therefore, $\left|\frac{p_v - q_v}{p_v}\right| \leq \frac{4}{5}$. By Taylor's expansion,

$$\begin{aligned}
-p_v \ln \frac{q_v}{p_v} &= -p_v \ln\left(1 - \frac{p_v - q_v}{p_v}\right) \\
&\leq (p_v - q_v) + \sum_{i=2}^{\infty} \frac{|p_v - q_v|^i}{i \cdot p_v^{i-1}} \\
&\leq (p_v - q_v) + \sum_{i=2}^{\infty} \frac{|p_v - q_v|^i}{2p_v^{i-1}} \\
&= (p_v - q_v) + \frac{(p_v - q_v)^2}{2\left(p_v - |p_v - q_v|\right)} \\
&\leq (p_v - q_v) + O\left((p_v - q_v)^2\right).
\end{aligned}$$

Similarly, $-(1 - p_v) \ln \frac{1-q_v}{1-p_v} \leq (q_v - p_v) + O\left((p_v - q_v)^2\right)$. We get that

$$\mathbf{D}(P_v||Q_v) = -p_v \log \frac{q_v}{p_v} - (1 - p_v) \log \frac{1 - q_v}{1 - p_v} \leq O\left((p_v - q_v)^2\right).$$

For each non-excited vertex v it holds that $|p_v - q_v| \leq \frac{2k}{\sqrt{c}}$ and therefore, the non-excited vertices add at most $O(k^2)$ to the divergence cost along any path. For each excited vertex v it holds that $|p_v - q_v| \leq \frac{2}{\sqrt[8]{c}}$. By Claim 7, the probability that there are more than 3ℓ excited vertices on a path is at most $2^{-k/2} \leq 1/c$. Therefore, along an average path, the expected number of excited vertices is at most $O(\ell) = O(k\sqrt[4]{c})$ and they add at most

$$O\left(\ell \cdot \left(\frac{2}{\sqrt[8]{c}}\right)^2\right) = O(k)$$

to the expected divergence cost. Put together, the expected divergence cost of π' is $O(k^2)$. ◀

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.

2 Boaz Barak, Mark Braverman, Xi Chen, and Anup Rao. How to compress interactive communication. In *STOC*, pages 67–76, 2010.

3 Mark Braverman. Interactive information complexity. In *STOC*, pages 505–524, 2012.

4 Mark Braverman. A hard-to-compress interactive task? *In 51th Annual Allerton Conference on Communication, Control, and Computing*, 2013.

5 Mark Braverman and Anup Rao. Information equals amortized communication. In *FOCS*, pages 748–757, 2011.

6 Mark Braverman and Omri Weinstein. A discrepancy lower bound for information complexity. In *APPROX-RANDOM*, pages 459–470, 2012.

7 Amit Chakrabarti, Yaoyun Shi, Anthony Wirth, and Andrew Chi-Chih Yao. Informational complexity and the direct sum problem for simultaneous message complexity. In *FOCS*, pages 270–278, 2001.

8 Anat Ganor, Gillat Kol, and Ran Raz. Exponential separation of information and communication. In *FOCS*, 2014.

9 Anat Ganor, Gillat Kol, and Ran Raz. Exponential separation of communication and external information. In *STOC*, pages 977–986, 2016.

10 Anat Ganor, Gillat Kol, and Ran Raz. Exponential separation of information and communication for boolean functions. *Journal of the ACM*, 63(5):46:1–46:31, 2016.

11 Amiram H. Kaspi. Two-way source coding with a fidelity criterion. *IEEE Transactions on Information Theory*, 31(6):735–740, 1985.

12 Iordanis Kerenidis, Sophie Laplante, Virginie Lerays, Jérémie Roland, and David Xiao. Lower bounds on information complexity via zero-communication protocols and applications. In *FOCS*, pages 500–509, 2012.

13 Gillat Kol. Interactive compression for product distributions. In *STOC*, pages 987–998, 2016.

14 Eyal Kushilevitz and Noam Nisan. Communication complexity. *Cambridge University Press*, 1997.

15 Troy Lee and Adi Shraibman. Lower bounds in communication complexity. *Foundations and Trends in Theoretical Computer Science*, 3(4):263–398, 2009.

16 Alon Orlitsky and James R. Roche. Coding for computing. *IEEE Transactions on Information Theory*, 47(3):903–917, 2001 (Preliminary version at the IEEE International Symposium on Information Theory (ISIT) 1995, FOCS 1995).

17 Anup Rao and Makrand Sinha. Simplified separation of information and communication. *Electronic Colloquium on Computational Complexity (ECCC)*, 2015.

18 Alexander A. Sherstov. Compressing interactive communication under product distributions. In *FOCS*, pages 535–544, 2016.

Information Value of Two-Prover Games*

Mark Braverman[1] and Young Kun Ko[2]

1 **Princeton University, 35 Olden St. Princeton NJ 08540, USA**
`mbraverm@cs.princeton.edu`
2 **Princeton University, 35 Olden St. Princeton NJ 08540, USA**
`yko@cs.princeton.edu`

Abstract

We introduce a generalization of the standard framework for studying the difficulty of two-prover games. Specifically, we study the model where Alice and Bob are allowed to communicate (with information constraints) – in contrast to the usual two-prover game where they are not allowed to communicate after receiving their respective input. We study the trade-off between the information cost of the protocol and the achieved value of the game after the protocol. In particular, we show the connection of this trade-off and the amortized behavior of the game (i.e. repeated value of the game). We show that if one can win the game with at least $(1-\epsilon)$-probability by communicating at most ϵ bits of information, then one can win n copies with probability at least $2^{-O(\epsilon n)}$. This gives an intuitive explanation why Raz's counter-example to strong parallel repetition [16] (the odd cycle game) is a counter-example to strong parallel repetition – one can win the odd-cycle game on a cycle of length m by communicating $O(m^{-2})$-bits where m is the number of vertices.

Conversely, for projection games, we show that if one can win n copies with probability larger than $(1-\epsilon)^n$, then one can win one copy with at least $(1-O(\epsilon))$-probability by communicating $O(\epsilon)$ bits of information. By showing the equivalence between information value and amortized value, we give an alternative direction for further works in studying amortized behavior of the two-prover games.

The main technical tool is the "Chi-Squared Lemma" which bounds the information cost of the protocol in terms of Chi-Squared distance, instead of usual divergence. This avoids the square loss from using Pinsker's Inequality.

1998 ACM Subject Classification F.1.3 Complexity Measures and Classes

Keywords and phrases Two Prover Game, Parallel Repetition, Odd-Cycle Game, Amortized Value of the Game

Digital Object Identifier 10.4230/LIPIcs.ITCS.2018.12

1 Introduction

A *two-prover one-round game* $\mathcal{G}$ on a bipartite graph with (U, V, E) with distribution $\mathcal{D}$ on E is defined as a following process. Referee picks $(u, v) \in E$ according to $\mathcal{D}$, then sends u to Alice and v to Bob. Then Alice gives an assignment to u and Bob to v from alphabet Σ. Referee checks if they are "valid" assignment for the edge (u, v). In this setting, we are interested in what the best response strategy is for Alice and Bob. In particular, we want to find $f : U \to \Sigma$ which denotes Alice's strategy and $g : V \to \Sigma$ which denotes Bob's strategy

* A full version of the paper is available at `https://eccc.weizmann.ac.il/report/2017/182/`.

that maximize the fraction of satisfied edges. In particular, we want to compute the strategy that achieves the *value* of the game which is defined as

$$\mathrm{val}(\mathcal{G}) := \max_{f,g} \Pr_{(u,v)\sim\mathcal{D}} \left[\pi_{(u,v)}(f(u), g(v)) = 1\right],$$

that is the probability of satisfying a randomly chosen edge according to the best response strategy where $\pi_{(u,v)}$ is the verification function by the referee for the edge (u, v).

In the above setup, it is crucial that Alice's assignment only depends on the input u and Bob's assignment only depends on the input v. In other words, Alice and Bob are assumed to be in separate rooms, leaking zero bits of information about their respective input. Having introduced the amount of information communicated between Alice and Bob into the picture, we could then reformulate the *value* of the game as:

> *Given a game $\mathcal{G}$, Alice and Bob communicate zero bits of information. Then what is the best chance of winning the game?*

Then it is natural to extend to following question: 'If Alice and Bob are allowed to communicate limited information, what is the value of the game?' In particular, we could explicitly ask the following question.

> *If Alice and Bob are allowed to communicate ε bits of information, then what is the value of the game? (in terms of ε)*

First, note that this is a well-defined quantity in a sense that there is a following explicitly bounded curve. Observe that if $|U| = |V| = n$, and they are allowed to communicate $O(\log n)$-bits, the value of the game becomes 1 (if all the edges indeed have at least one satisfying assignment, which can be assumed without loss of generality) due to the following naive strategy. Alice simply sends the hash of her input to Bob, which requires at most $\log n$-bits to do so and vice-versa. Since Alice and Bob both know (u, v), they can simply pick a satisfying assignment (using shared randomness) for (u, v) then answer accordingly. We can further tighten the upper bound (for the amount of information) if we know the structure of the graph. In particular, if the graph were d-regular, given $O(\log d)$-bits, the value of the game again becomes 1, since the entropy (of Alice's input given Bob's input and vice-versa) is at most $\log d$. (We will show this explicitly)

We would like to further investigate *the trade-off between the information vs. the value of the game*. In particular, we initiate the study of *information value of the game*, that is *how much information is necessary* to win the game with say probability $> 1 - \delta$ in terms of δ, which we define more explicitly in Section 2.

Note that this notion can classify "how intrinsically hard" a given two-prover one-round game is. In particular, one could view a game as being "hard" if the value of the game is "resistant" to added information, easy otherwise, providing a better spectrum for analyzing the intrinsic hardness of the game – which is indeed related to the amortized value of the game.

1.1 Our Contribution

We connect the information value of the game with the amortized value of the game – i.e. the value of the repeated game. We note that previous parallel repetition literature can be "translated" as Alice and Bob sharing a common hint provided by the referee. Then we can "remove" the referee from the picture and instead let Alice and Bob sample a common hint. This translation is what we call the "Chi-squared lemma," which is the main technical contribution of this paper.

▶ **Lemma 1** (Chi-squared lemma). *Suppose Alice has access to P and Bob has access to Q which are probability distributions over the universe $\mathcal{U}$. Suppose further that there exists a common distribution R such that $D(R||P) < \varepsilon$ and $D(R||Q) < \varepsilon$. Then there exists a protocol Π that outputs a sample from the distribution $\tilde{P}$ with information cost $\gamma\varepsilon$ with $D(R||\tilde{P}) < O(\varepsilon)$ for some constant $\gamma > 0$.*

where "information cost" refers to the amount of information revealed to each other as in [4].

We also remark that the idea of referring to Chi-Squared distance to bound information cost was also used in proving sharp round complexity of pointer chasing problem [20], avoiding square-loss in the parameter due to the application of Pinsker's inequality.

Via this lemma, we can "translate" previous parallel repetition literature (in particular [3]) as a blackbox to obtain following main theorem.

▶ **Theorem 2** (informal). *If $\mathrm{val}(\mathcal{G}^n) \geq (1-\varepsilon)^n = 2^{-\Theta(\varepsilon n)}$ and $\mathcal{G}$ is a projection game, that is for any valid pair of assignments, Bob's assignment is a projection of Alice's assignment, then one can win $\mathcal{G}$ with probability $1 - O(\varepsilon)$ by communicating $O(\varepsilon)$-bits of information.*

To show the converse, we need to show that a low-information protocol can be translated to a zero-communication protocol with insignificant loss in success probability. We use results from correlated sampling (in particular we use a lemma from [3]) which yields the converse to the main theorem.

▶ **Theorem 3** (informal). *If one can win $\mathcal{G}$ with probability $1 - \varepsilon$ by communicating ε-bits of information, then $\mathrm{val}(\mathcal{G}^n) \geq 2^{-O(\varepsilon n)}$.*

We also remark that Theorem 3 **does not assume** that $\mathcal{G}$ is a projection game.

1.2 Proof Overview

Proof of Theorem 3

It suffices to show that a low-information cost protocol can be translated to a zero-communication protocol with success parameter depending on the information cost. In particular, we want to show that an $O(I)$ information cost protocol can be simulated by two non-communicating parties with $2^{-O(I)}$ success probability. For our purpose $I = n\varepsilon$, since information cost tensorizes with many copies. Note that if Alice and Bob managed to sample a correct transcript together, the transcript is correct with $(1-\varepsilon)^{O(n)}$ probability since each coordinates are chosen independently. We show that the zero-communication sampling lemma from [3] indeed gives the range of parameters that we need.

Proof of Theorem 2

In [3], one could view the "common hint" (which actually comes from the multiple copies of the game) as given by the referee to Alice and Bob via sampling some random coordinates with their answers then sending them to Alice and Bob. However, this does not fit in our framework since the referee samples the hint, not Alice and Bob. Using the Chi-squared lemma, we allow Alice and Bob to jointly sample such a hint with $O(\varepsilon)$ information cost. [3] shows that one can win a single copy with probability $1 - O(\varepsilon)$ after running the joint sampling protocol (under some distribution $\mathcal{D}'$ which is $O(\varepsilon)$-away in terms of divergence from the real distribution). We further show that having a strategy for such $\mathcal{D}'$ suffices to win the original game with probability $1 - O(\varepsilon)$ as well.

Chi-squared lemma

Suppose Alice has access to P_1 and Bob has access to P_2 with a guarantee that there exists some common distribution R such that $D(R||P_1), D(R||P_2) < \varepsilon$. In such a setting, if Alice samples from P_1 and tries to transmit the sample to Bob, the naive information cost would be $D(P_1||P_2)$. This could be unbounded however due to the case where P_1 contains an element in the support that is not in the support of P_2. i.e. $D(P_1||P_2) = \infty$. To rule out such scenario, we give Bob an ability to reject. Instead of receiving the full description or index of the sample, Bob receives a stream of hash values and rejects the stream when he "cannot understand" or in other words expects the divergence from the sample to be high. Via this rejection, Bob will not be "too surprised" about Alice's sample. But the main problem with the above simple rejection protocol is that it allows Alice to learn too much about P_2 from Bob's response. For instance, if Bob rejects a sample, then Alice learns that this sample occurs "infrequently" in P_2. In order to "confuse" Alice and prevent her from learning too much about P_2, Bob rejects a valid stream with some constant probability. This suffices to confuse Alice and learn only $O(\varepsilon)$-information about P_2.

2 Preliminaries

2.1 Information Theory

In this section, we provide background on information theory that will be used to prove main results. We remark that throughout the paper, log is of base 2 and ln is of base e. For further references, we refer the reader to [7].

▶ **Definition 4** (Entropy). The *entropy* of a random variable A, denoted by $H(A)$ is defined as

$$\sum_{a \in \mathsf{Supp}(A)} \Pr[A = a] \log \frac{1}{\Pr[A = a]}$$

Intuitively, this quantifies how much uncertainty we have about variable A. With the definition of entropy, we can further define the relation between various variables. For conditional entropy we have $H(A|B) := H(AB) - H(B)$. Then we are ready to define the relation between different random variables.

▶ **Definition 5** (Mutual Information). The *mutual information* between two random variable A and B, denoted by $I(A;B)$ is defined as

$$I(A;B) := H(A) - H(A|B) = H(B) - H(B|A).$$

The *conditional mutual information* between A and B given C, denoted by $I(A;B|C)$, is defined as

$$I(A;B|C) := H(A|C) - H(A|BC) = H(B|C) - H(B|AC).$$

This gives a measure of how much information does B reveal about A and vice-versa. (when one knows C) Mutual Information is further related to the following distance measure, which will be used throughout the proof.

▶ **Definition 6** (Kullback-Leiber Divergence). Given two probability distributions μ_1 and μ_2 on the same sample space Ω such that $(\forall \omega \in \Omega)(\mu_2(\omega) = 0 \Rightarrow \mu_1(\omega) = 0)$, the *Kullback-Leibler*

Divergence or KL-Divergence in short between μ_1 and μ_2 is defined as (also known as relative entropy)

$$D(\mu_1||\mu_2) = \sum_{\omega\in\Omega} \mu_1(\omega) \log \frac{\mu_1(\omega)}{\mu_2(\omega)}.$$

In particular, the following equality holds between KL-divergence and mutual information.

▶ **Fact 7.** *For random variables A, B and C we have*

$$I(A;B|C) = \mathbb{E}_{b,c}\left[D(A_{bc}||A_c)\right].$$

where A_{bc} is the distribution of random variable A conditioned on $B = b, C = c$ and similarly for A_c.

With the definitions in place, we provide useful properties that will be used in the proof. For mutual information, following facts hold.

▶ **Fact 8** (Chain-rule)**.** *If $I(B;D|C) = 0$, then $I(A;B|C) \leq I(A;B|C,D)$.*

▶ **Fact 9** (Super-Additivity of Mutual Information)**.** *Let C_1, C_2, D, B be random variables such that for every fixing of D, C_1 and C_2 are independent. Then*

$$I(C_1;B|D) + I(C_2;B|D) \leq I(C_1C_2;B|D).$$

For KL-divergence, we use following properties.

▶ **Fact 10.** *Let P and Q be distributions over a universe $\mathcal{U}$. Suppose $\mathcal{V} \subseteq \mathcal{U}$ is such that $P(\mathcal{V}) = 1$. Then $Q(\mathcal{V}) \geq 2^{-D(P||Q)}$.*

Note that Fact 10 immediately implies non-negativity of KL-divergence between any two distributions. For KL-divergence under conditioning, the following property holds.

▶ **Fact 11** (Additivity of KL-Divergence)**.** *Consider two distributions $P(x,y)$ and $Q(x,y)$. Then*

$$D(P(x,y)||Q(x,y)) = D(P(x)||Q(x)) + \mathbb{E}_{x\sim P} D(P(y|x)||Q(y|x))$$

2.2 Previous Work

In this section, we elaborate previous results on parallel repetition and how the state-of-the-art proof technique for parallel repetition is related to the amount of information necessary for Alice and Bob to win the game with probability greater than $(1-\varepsilon)$.

Recall that r-times parallel repetition of a two-prover game $\mathcal{G}$ denoted as $\mathcal{G}^r$ is defined as: the referee first samples r-tuple of edges i.e. $\vec{x} = (x_1, \ldots x_r) \in U^r$ and $\vec{y} = (y_1, \ldots, y_r) \in V^r$ with $(x_i, y_i) \in E$ for all $i \in [r]$; Alice and Bob give assignments to all r-coordinates say $f : U^r \to \Sigma^r$ for Alice and $g : V^r \to \Sigma^r$ for Bob; then the referee checks all r-coordinates i.e. return $\bigwedge_{i\in[r]} \pi_{x_i,y_i}(f_i(\vec{x}), g_i(\vec{y}))$.

The first parallel repetition theorem (with exponential decay in value) was proved by Raz [15]:

▶ **Theorem 12** ([15])**.** *Let $\mathcal{G}$ be a game with $\mathbf{val}(\mathcal{G}) = 1-\varepsilon$ and let s be the size of the alphabet ($|\Sigma|$) of the game. Then $\mathbf{val}(\mathcal{G}^n) \leq (1-\varepsilon^{32}/2)^{\Omega(n/\log(s))}$.*

This was improved (and simplified) by Holenstein [10]:

▶ **Theorem 13** ([10]). *Let $\mathcal{G}$ be a game with $\mathbf{val}(\mathcal{G}) = 1 - \varepsilon$ and let $\log(s)$ be the answer size of the game. Then $\mathbf{val}(\mathcal{G}^n) \leq (1 - \varepsilon^3/2)^{\Omega(n/\log(s))}$.*

For projection games, $\log(s)$ no longer appears in the exponent. In particular, [14] showed improved bound for projection games.

▶ **Theorem 14** ([14]). *Let $\mathcal{G}$ be a projection game with $\mathbf{val}(\mathcal{G}) = 1 - \varepsilon$. Then $\mathbf{val}(\mathcal{G}^n) \leq (1 - \varepsilon^2/2)^{\Omega(n)}$.*

There is an initiation of work from [1], [18] where they study (though not explicitly stated) a quantity $\mathbf{val}_+(\mathcal{G})$ for unique game $\mathcal{G}$ which is an analytic relaxation of $\mathbf{val}(\mathcal{G})$ that tensorizes exactly, that is $\mathbf{val}_+(\mathcal{G}_1 \otimes \mathcal{G}_2) = \mathbf{val}_+(\mathcal{G}_1) \cdot \mathbf{val}_+(\mathcal{G}_2)$ and captures the amortized value of the game. The analysis of $\mathbf{val}_+$ then resulted in analytic proof of parallel repetition for projection games. [8], [19]. In particular, [8] extended [14] to low-value regime using $\mathbf{val}_+$.

▶ **Theorem 15** ([8]). *Let $\mathcal{G}$ be a projection game with $\mathbf{val}(\mathcal{G}) = \beta$. Then $\mathbf{val}(\mathcal{G}^n) \leq (4\beta)^{n/4}$.*

There has been more work on parallel repetition for various special settings: [2] [17] [19] In particular, there also has been a series of works around parallel repetition of games with entanglement [6, 12, 9, 11, 5]. A recent breakthrough in [21] settled a longstanding open problem on whether the value of "any" games with entanglement actually decays to zero as the number of repetition goes to infinity (at a polynomial rate). It would be interesting to see how our framework relates to games with entanglement.

Throughout this paper, we will use the machinery in the latest parallel repetition proof by [3] where they handled the low-value regime. It should be noted that the same proof works in the high-value regime as well, giving an alternate proof for [10], [14] and [8].

▶ **Theorem 16** ([3]). *Let $\mathcal{G}$ be a game with $\mathbf{val}(\mathcal{G}) = \delta$. Then $\mathbf{val}(\mathcal{G}^n) \leq \delta^{\Omega(n \log(1/\delta)/\log(s))}$. Further if $\mathcal{G}$ is a projection game with $\mathbf{val}(\mathcal{G}) = 1 - \varepsilon$, then $\mathbf{val}(\mathcal{G}^n) \leq (1 - \varepsilon^2)^{\Omega(n)}$.*

The main proof technique used to prove parallel repetition in [3] follows the following general roadmap. First one assumes that the value of the repeated game is higher than the desired bound, and focuses on the event where Alice and Bob win the whole copy. Conditioned on winning, one sets up R, the common hint between Alice and Bob, as a subset of question and answer pairs from other coordinates which can be individually sampled by Alice and Bob (approximately), which is the main technical innovation of [3]. Conditioned on successfully sampling R, Alice and Bob's strategy becomes a "too good to be true" strategy for some coordinate, contradicting the original assumption on the value of the game. For the purpose of having a "hint" between Alice and Bob, we mainly focus on sampling R, avoiding technical issues of constructing R correctly via using [3] as a blackbox. One could view the framework in [3] as following explicit model in Protocol 1.

Protocol 1 Non-Protocol Hint.

1. Referee picks a random edge $(x, y) \in E$ as a challenge. Referee then samples r from R_{xy} then transmits r to Alice and Bob
2. Alice, depending on r and x provides an assignment a. Bob analogously answers b depending on r and y.
3. Referee accepts if a and b forms a satisfying assignment for (x, y).

Protocol 1 is indeed not a protocol between Alice and Bob, since the referee samples the hint. Converting this to a protocol between Alice and Bob is our main technical contribution.

For the sake of completeness, we describe how the common hint R is constructed below.

Precise construction of R in [3]

They explicitly construct a "combinatorial" hint $R_{S,G,H,I}$ in a following manner which we restate for completeness. Let n be the number of repetitions, that is the number of coordinates for the game. Then let S, G, H be random subsets of $[n]$ distributed as follows: Let s_h and s_g be random numbers from $\{3n/4+1, \ldots, n\}$. Let $\sigma : [n] \to [n]$ be a uniformly random permutation. Set $H = \sigma([s_h])$, $G = \sigma(\{n - s_g + 1, \ldots, n\})$. Let I be a uniformly random element of $G \cap H$. Let l be a random number from $[T]$, where $T < n/2$ is a parameter. Let S be a uniformly random subset of $G \cap H \backslash \{I\}$ of size l. Then define $R_{S,G,H,I}$ to denote the random variable $X_{G\backslash\{I\}} Y_{H\backslash\{I\}} A_S B_S$ where s, g, h, i denote instantiations of the random variables S, G, H, I respectively with A_S denoted Alice's assignment on S-coordinates and respectively for B_S. Then Alice can be thought of getting $X_{[n]}$ and $R_{S,G,H,I}$, while Bob gets $Y_{[n]}$ and $R_{S,G,H,I}$ where the input (x, y) is set to (X_I, Y_I).

2.3 Definitions

Recall that the **Information Cost** of a protocol is defined as $I(\Pi; X|Y) + I(\Pi; Y|X)$ where Π is the transcript of the protocol, X and Y are inputs for Alice and Bob respectively. **Information Complexity** of computing f is then defined as infimum over Π that computes f. Inspired by the definitions from information complexity literature, we define information value of the game as following.

▶ **Definition 17.** The **information value** of the game $\mathcal{G} = (X, Y, E)$ with distribution $\mathcal{D}$ over E is

$$\mathbf{IV}^{\varepsilon}_{\mathcal{D}}(\mathcal{G}) := \inf_{\Pi} [I(\Pi; X|Y) + I(\Pi; Y|X)]$$

where the infimum is taken over the set of transcripts Π between Alice and Bob which wins $\mathcal{G}$ with probability at least $(1 - \varepsilon)$ with $\varepsilon < 1/2$. X and Y represents Alice and Bob's input respectively.

We remark that if $\mathcal{D}$ is not specified, we assume the distribution to be the uniform distribution over the challenges/edges.

As a straightforward exercise, note that $I(\Pi; X|Y) \leq H(X|Y) \leq H(X) \leq \log n$ where n is the number of vertices in the graph, similarly for $I(\Pi; Y|X)$. Thus for any game $\mathcal{G}$ and $\varepsilon \geq 0$, $\mathbf{IV}^{\varepsilon}_{\mathcal{D}}(\mathcal{G}) \leq O(\log n)$. Thereby, this quantity is strictly bounded. Better bound holds for d-regular graphs since $H(X|Y) \leq \log d$ for regular graphs, similarly for $H(Y|X)$. Any better bounds however, requires lower bound on $H(X|\Pi, Y)$.

3 Main Result

First we state the main technical lemma (Chi-Squared lemma) used in proving the main theorem.

▶ **Lemma 18** (Chi-Squared lemma). *Suppose Alice has access to a distribution P and Bob has access to a distribution Q over $\mathcal{U}$. Suppose further that there exists a common distribution R such that $D(R||P) < \varepsilon$ and $D(R||Q) < \varepsilon$. If $\varepsilon < 1/50$, then there exists a protocol Π that outputs a sample from $\tilde{P}$ with information cost $\gamma\varepsilon$ with $D(R||\tilde{P}) < O(\varepsilon)$ for some constant $\gamma > 0$.*

We append the full proof in the full version of the paper. To see why this lemma is interesting, note that the naive information cost is $D(P||Q)$ which could be infinite in some cases.

Technically speaking, the triangle inequality does not hold for divergence. Applying Pinsker's Inequality to have triangle inequality (in total variation distance) leads to a square loss, resulting in information cost $O(\sqrt{\varepsilon})$, instead of $O(\varepsilon)$. We suspect that there are more applications to this technical lemma. The lemma leads to the following theorem.

▶ **Theorem 19.** *Let $\varepsilon < 1/2$. If $\mathrm{val}(\mathcal{G}^n) \geq (1-\varepsilon)^n = 2^{-\Omega(\varepsilon n)}$ and $\mathcal{G}$ is a projection game, then there exists constants $\alpha_1, \alpha_2 > 0$ such that $\mathbf{IV}^{\alpha_1 \varepsilon}(\mathcal{G}) < \alpha_2 \varepsilon$.*

The main intuition of the proof of Theorem 19 is to use the common hint used for dependency breaking step of the parallel repetition, which are answers and questions in the other coordinates, as hints between Alice and Bob. However, the hints are not exactly adequate for our application, since they are sampled by the referee. We use Lemma 18 to convert it to a low information cost protocol.

We also show that the converse of Theorem 19.

▶ **Theorem 20.** *If $\mathbf{IV}^{\varepsilon}(\mathcal{G}) < \varepsilon$ and $\varepsilon < 1/2$, then $\mathrm{val}(\mathcal{G}^n) \geq 2^{-\Omega(\varepsilon n)} = (1-\varepsilon)^{\Omega(n)}$ with $n > 1/\varepsilon$.*

The main intuition to Theorem 20 is converting a low information cost protocol (for our application $O(n\varepsilon)$) to a zero-communication protocol as seen in [13]. However, the main theorem from [13] does not suffice for our application. Instead, we use a lemma from [3].

As a corollary, we get a complete description of projection games that obey strong parallel repetition in terms of information value of the game.

▶ **Corollary 21.** *If $\mathbf{IV}^{\varepsilon}(\mathcal{G}) > \varepsilon$, then $\mathrm{val}(\mathcal{G}^n) < (1-\varepsilon)^{O(n)}$ and vice versa where $\mathcal{G}$ is a projection game.*

Applying previous parallel repetition result, we indeed get a non-trivial lower bound on the information value of any projection game via [8] and [14].

▶ **Corollary 22.** *For any projection game $\mathcal{G}$ with $\mathrm{val}(\mathcal{G}) \leq 1-\varepsilon$, $\mathbf{IV}^{O(\varepsilon^2)}(\mathcal{G}) > \Omega(\varepsilon^2)$.*

3.1 Proof of Theorem 19

In this section, we prove Theorem 19 via Chi-squared Lemma (Lemma 18). Recall that $R_{s,g,h,i}$ defined in Section 2.2 is a set of challenges and answers on a random set of coordinates. Set $T = n/4$ as the parameter for $R_{s,g,h,i}$. Then we get the following key lemma from [3].

▶ **Lemma 23** (Lemma 5.6 of [3])**.** *Suppose $2^{-20} \geq \Pr[W] \geq (1-\varepsilon)^n$. Then there exists a fixing of s, g, h, i such that:*

1. $\mathbb{E}_{x,y\sim\mu} D\left(P_{R_{s,g,h,i}|X_i=x,Y_i=y,W} || P_{R_{s,g,h,i}|X_i=x,W}\right) \leq O(\varepsilon)$.
2. $\mathbb{E}_{x,y\sim\mu} D\left(P_{R_{s,g,h,i}|X_i=x,Y_i=y,W} || P_{R_{s,g,h,i}|Y_i=y,W}\right) \leq O(\varepsilon)$.
3. $D\left(\mu || P_{X_iY_i}\right) \leq O(\varepsilon)$.
4. $\mathbb{E}_{x,y\sim\mu}\mathbb{E}_{r\sim R_{s,g,h,i}|X_i=x,Y_i=y,W}$
   $D\left(P_{A_i,B_i|X_i=x,Y_i=y,R_{s,g,h,i}=r,W} || P_{A_i|X_i=x,R_{s,g,h,i}=r,W} \otimes P_{B_i|Y_i=y,R_{s,g,h,i}=r}\right) \leq O(\varepsilon)$.

where μ denotes the distribution $P_{X_i,Y_i|W}$ and P_X stands for the probability distribution of random variable X.

We omit the proof

▶ Remark. The last property does not suffice for our application, since we do not get $r \sim R_{s,g,h,i}|X_i = x, Y_i = y, W$ but a distribution that is $O(\varepsilon)$-away from it in divergence at the end of the protocol given by the Chi-squared lemma which we denote as $\tilde{R}_{s,g,h,i}|X_i =$

$x, Y_i = y, W$. However, we remark that the same proof in [3] indeed gives the property that we want. That is

$$\mathbb{E}_{x,y\sim P_{X_i,Y_i|W}} \mathbb{E}_{r\sim \tilde{R}_{s,g,h,i}|X_i=x,Y_i=y,W}$$
$$D\left(P_{A_i,B_i|X_i=x,Y_i=y,R_{s,g,h,i}=r,W} || P_{A_i|X_i=x,R_{s,g,h,i}=r,W} \otimes P_{B_i|Y_i=y,R_{s,g,h,i}=r}\right) \leq O(\varepsilon). \quad (1)$$

In particular, note that the distribution of s, g, h, i and the permutation remain the same since Alice and Bob can agree (via public randomness prior to running the protocol) on them prior to sampling the actual question and answer sets (by Alice) This suffices for the proof in [3], specifically Lemma 5.2 and Lemma 5.5.

We also need the following lemma to translate a strategy on $X_i, Y_i|W$ to a strategy on actual distribution X_i, Y_i. Due to space constraints, we attach the proof in Section A.

▶ **Lemma 24.** *Suppose $\mathcal{G}$ with μ as the distribution over the edges achieves $val(\mathcal{G}) = 1 - \varepsilon$. Then consider $\tilde{\mu}$ such that $D(\mu||\tilde{\mu}) < \varepsilon$. Then $\mathcal{G}$ with $\tilde{\mu}$ as distribution over the edges has value $> 1 - O(\varepsilon)$.*

Now we have all the necessary lemmas to prove Theorem 19.

Proof of Theorem 19. First we construct a low information protocol that wins $\mathcal{G}$ with probability $1 - O(\varepsilon)$ under μ.

We write $P_{R_{s,g,h,i}|X_i=x,Y_i=y,W}$ in the above as $R_{x,y}$ and $P_{R_{s,g,h,i}|X_i=x,W}$, $P_{R_{s,g,h,i}|Y_i=y,W}$ respectively as P_x, Q_y. Consider $\mathcal{S} \subset E$ that satisfies all

- $D\left(R_{x,y}||P_x\right) \leq \frac{1}{10\gamma}$.
- $D\left(R_{x,y}||Q_y\right) \leq \frac{1}{10\gamma}$.

where γ is the constant from the Chi-Squared Lemma. Then note that $\mu(\mathcal{S}) > 1 - O(\gamma\varepsilon) = 1 - O(\varepsilon)$ by Markov's inequality. Focus pairs in $\mathcal{S}$. Now applying the protocol given by the Chi-Squared Lemma to pairs in $\mathcal{S}$, we obtain a protocol that samples $r \sim \tilde{R}_{s,g,h,i}|X_i = x, Y_i = y, W$ with information cost at most

$$\mathbb{E}_{x,y\sim\mathcal{S}_{X_i,Y_i|W}}\left[D\left(R_{x,y}||P_x\right) + D\left(R_{x,y}||Q_y\right)\right] < O(\varepsilon)$$

where $\mathcal{S}_{X_i,Y_i|W}$ is the distribution over the edges further conditioned on $\mathcal{S}$.

Since $\mathcal{S}$ contributes $1 - O(\varepsilon)$-fraction, (1) implies

$$\mathbb{E}_{x,y\sim\mathcal{S}_{X_i,Y_i|W}} \mathbb{E}_{P_{\tilde{R}_{s,g,h,i},X_i=x,Y_i=y|W}}$$
$$D\left(P_{A_i,B_i|X_i=x,Y_i=y,R_{s,g,h,i}=r,W} || P_{A_i|X_i=x,R_{s,g,h,i}=r,W} \otimes P_{B_i|Y_i=y,R_{s,g,h,i}=r}\right) \leq O(\varepsilon) \quad (2)$$

At the end of the protocol, Alice and Bob obtain same $r \sim \tilde{R}_{s,g,h,i}|X_i = x, Y_i = y, W$. We now construct an explicit answering strategy for Alice and Bob dependent on r when they get edges distributed according to $\mathcal{S}_{X_i,Y_i|W}$. Ideally Alice and Bob would like to answer according to $P_{A_i,B_i|X_i=x,Y_i=y,\tilde{R}_{s,g,h,i}=r,W}$. This would indeed succeed with probability 1. In other words, if we define $\mathcal{G}_{x,y} = \{(a,b)|V(x,y,a,b) = 1\}$, $P_{A_i,B_i|X_i=x,Y_i=y,\tilde{R}_{s,g,h,i}=r,W}(\mathcal{G}_{x,y}) = 1$ for all $(x,y) \in \mathcal{S}$. However, this is not a valid strategy. There is correlation between A_i and B_i, while for any valid strategy they should be independent given respective input.

Instead, they answer according to $P_{A_i|X_i=x,\tilde{R}_{s,g,h,i}=r,W} \otimes P_{B_i|Y_i=y,\tilde{R}_{s,g,h,i}=r}$. Now, we analyze $P_{A_i|X_i=x,\tilde{R}_{s,g,h,i}=r,W} \otimes P_{B_i|Y_i=y,\tilde{R}_{s,g,h,i}=r}(\mathcal{G}_{x,y})$ i.e. the value of such strategy. Applying Fact 10,

$$P_{A_i|X_i=x,\tilde{R}_{s,g,h,i}=r,W} \otimes P_{B_i|Y_i=y,\tilde{R}_{s,g,h,i}=r}(\mathcal{G}_{x,y})$$

$$\geq 2^{-D\left(P_{A_i,B_i|X_i=x,Y_i=y,R_{s,g,h,i}=r,W}||P_{A_i|X_i=x,R_{s,g,h,i}=r,W}\otimes P_{B_i|Y_i=y,R_{s,g,h,i}=r}\right)}$$

By the convexity of 2^{-x} along with (2), we get the desired bound

$$\mathbb{E}_{x,y\sim\mathcal{S}_{X_i,Y_i|W}}\left[P_{A_i|X_i=x,\tilde{R}_{s,g,h,i}=r,W}\otimes P_{B_i|Y_i=y,\tilde{R}_{s,g,h,i}=r}(\mathcal{G}_{x,y})\right]\geq 2^{-O(\varepsilon)}=1-O(\varepsilon)$$

Since $\mathcal{S}$ contributes $1-O(\varepsilon)$-fraction on μ, this strategy wins with $1-O(\varepsilon)$ probability when the edges are distributed according to $\mu=P_{X_i,Y_i|W}$ as well. Finally, applying Lemma 24 to this strategy with $D\left(P_{X_iY_i|W}||P_{X_iY_i}\right)\leq O(\varepsilon)$, we get the desired claim. ◀

3.2 Proof of Theorem 20

In this section, we give a formal proof of Theorem 20. This involves converting a protocol (between Alice and Bob) with $O(n\varepsilon)$-information cost to a zero-communication protocol with success probability $2^{-O(n\varepsilon)}$. We start by stating the following lemma.

▶ **Lemma 25.** *Suppose Alice has access to distribution P and Bob has access to distribution Q over the universe $\mathcal{U}$. They wish to jointly sample from R where $D(R||P)<\delta$ and $D(R||Q)<\delta$. If $\delta>1$, then there exists a zero-communication protocol such that*

1. *There exists an event E such that $\Pr[E]>2^{-\Omega(\delta)}$ and $\Pr[\pi_a=\pi_b|E]=1$, where π_a and π_b refers to the final output of Alice and Bob respectively. Furthermore, E only depends on the public randomness.*
2. *Given E, consider the set of outputs of π, denoted as $\mathcal{S}$. Then $\mathcal{S}\subseteq\mathsf{Supp}(R)$*

▶ **Claim 26.** *Let W be a subset of universe $\mathcal{U}$. Let A and B be a distribution and A_W be a distribution of A conditioned on picking an element from W. Then if $A(W)>\Omega(D(A||B))$, then*

$$D(A_W||B)<\log(1/A(W))+\frac{D(A||B)}{A(W)}+O\left(\frac{1-A(W)}{A(W)}\right)$$

where $A(W)$ corresponds to the probability of picking an element from W under A.

Proof of Lemma 25 and Claim 26 are appended in Section A. Now, we are ready to prove the main lemma of this section which implies Theorem 20.

▶ **Lemma 27.** *If $\mathbf{IV}^{\varepsilon}(\mathcal{G})<\varepsilon$, then there exists a zero-communication protocol that achieves $\mathsf{val}(\mathcal{G}^n)>2^{-\Omega(\varepsilon n)}$ where $n>1/\varepsilon$.*

Proof. Note that under this model, Alice's strategy and Bob's strategy are dependent on the transcript $\pi_{x,y}$ as well, instead of just their input in zero-communication model. We denote $\Pi_{x,y}$ as the distribution over the transcript that Alice and Bob will have when they are given input x and y respectively. Now Alice and Bob will try to imitate each other by simulating the other party in zero-communication setting. Let Π_x,Π_y denote the simulated transcript with input x and y respectively. More precisely, $\Pi_x:=\mathbb{E}_{y\sim\mu|x}\Pi_{x,y}$ and $\Pi_y:=\mathbb{E}_{x\sim\mu|y}\Pi_{x,y}$. Further, we introduce the notation $\Pi^W_{x,y}:=\Pi_{x,y}|W$, the distribution of $\Pi_{x,y}$ conditioned on referee accepting what Alice and Bob returns as their answer at the end of the protocol.

Note that from our assumption on the information cost of the protocol, we get the following

$$\mathbb{E}_{(x_i,y_i)\sim\mu}\left[D(\Pi_{x_iy_i}||\Pi_{x_i})\right]=\mathbb{E}_{(x_i,y_i)\sim\mu}\left[\mathbb{E}_{\Pi_{x_iy_i}}\left[\log\frac{\Pr_{\Pi_{x_iy_i}}[\pi]}{\Pr_{\Pi_{x_i}}[\pi]}\right]\right]<\varepsilon. \tag{3}$$

The same inequality holds for Bob's side ($D(\Pi_{x_iy_i}||\Pi_{y_i})$) as well. First we define "good" edges. We say π is a good transcript if the referee accepts what Alice and Bob return after following the transcript π. [1] Then edge (x_i, y_i) is good if it satisfies both

- $A_{x_iy_i}(W) := \Pr_{\pi\sim\Pi_{x_iy_i}}[\pi \text{ is a good transcript}] > 1/2$ (i.e. most sampled transcripts are good);
- $D(\Pi_{x_iy_i}||\Pi_{x_i}) < 1/2$.

We argue that most of the edges are good. Due to our assumption on the value of the game that is,

$$\mathsf{val}(\mathcal{G}) = \mathbb{E}_{(x_i,y_i)\sim\mu}\left[A_{x_iy_i}(W)\right] > 1-\varepsilon,$$

at most 2ε-fraction of (x_i, y_i)'s does not satisfy the first condition. Also due to our divergence condition that is,

$$\mathbb{E}_{(x_i,y_i)\sim\mu}\left[D(\Pi_{x_iy_i}||\Pi_{x_i})\right] < \varepsilon$$

at most 2ε-fraction of the edges violate the second condition. Thus all but at most 4ε-fraction of the edges are good. Then we can write

$$\mathbb{E}_{x_iy_i\sim\tilde{\mu}}\left[D(\Pi_{x_iy_i}||\Pi_{x_i})\right] < O(\varepsilon) \tag{4}$$

where $\tilde{\mu}$ corresponds to μ conditioned on picking an edge that is good. Also note that without loss of generality, in such regime, one can assume that $1 - A_{x_iy_i}(W) > \Omega(D(\Pi_{x_iy_i}||\Pi_{x_i}))$ for all the edges. For edges that do not satisfy such condition, i.e. $1 - A_{x_iy_i}(W) < O(D(\Pi_{x_iy_i}||\Pi_{x_i}))$, the referee can randomly reject with probability $O(D(\Pi_{x_iy_i}||\Pi_{x_i}))$ to satisfy $1 - A_{x_iy_i}(W) > \Omega(D(\Pi_{x_iy_i}||\Pi_{x_i}))$. Indeed it will add up the rejection probability, but by at most $D(\Pi_{x_iy_i}||\Pi_{x_i})$ which indeed is good enough for application in our regime, since it is in expectation at most $O(\varepsilon)$. If (x_i, y_i) is indeed a good edge, applying Claim 26,

$$\begin{aligned} D(\Pi^W_{x_iy_i}||\Pi_{x_i}) &< \log(1/A_{x_iy_i}(W)) + \frac{D(\Pi_{x_iy_i}||\Pi_{x_i})}{A_{x_iy_i}(W)} + O\left(\frac{1-A_{x_iy_i}(W)}{A_{x_iy_i}(W)}\right) \\ &< \log(1/A_{x_iy_i}(W)) + 2D(\Pi_{x_iy_i}||\Pi_{x_i}) + O\left(\frac{1-A_{x_iy_i}(W)}{A_{x_iy_i}(W)}\right) \end{aligned}$$

where the second inequality holds since $A_{x_iy_i}(W) > 1/2$, $D(\Pi_{x_iy_i}||\Pi_{x_i}) < 1$, and our assumption that $1 - A_{x_iy_i}(W) > \Omega(D(\Pi_{x_iy_i}||\Pi_{x_i}))$ for all edges. Then

$$\begin{aligned} \mathbb{E}_{\tilde{\mu}}[D(\Pi^W_{x_iy_i}||\Pi_{x_i})] &< \mathbb{E}_{\tilde{\mu}}[\log(1/A_{x_iy_i}(W))] + 2\mathbb{E}_{\tilde{\mu}}[D(\Pi_{x_iy_i}||\Pi_{x_i})] + \mathbb{E}_{\tilde{\mu}}\left[O\left(\frac{1-A_{x_iy_i}(W)}{A_{x_iy_i}(W)}\right)\right] \\ &< O(\varepsilon) + O(\varepsilon) + O(\varepsilon) < O(\varepsilon) \end{aligned}$$

where the second inequality holds by Jensen's inequality on log and since ε is small enough.

Now consider taking n-copies of the game. In particular, we focus on $(\vec{x}, \vec{y}) \sim \tilde{\mu}^{\otimes n}$, that is all edges are "good" edges. Then observe that

$$\mathbb{E}_{(\vec{x},\vec{y})\sim\tilde{\mu}^{\otimes n}}\left[D(\bigotimes_{i\in[n]}\Pi^W_{x_iy_i}||\bigotimes_{i\in[n]}\Pi_{x_i})\right] = \mathbb{E}_{(\vec{x},\vec{y})\sim\tilde{\mu}^{\otimes n}}\left[\sum_{i\in[n]}D(\Pi^W_{x_iy_i}||\Pi_{x_i})\right] < O(n\varepsilon)$$

[1] π does not necessarily depend just on the input. It can depend on private randomness as well. But this is not crucial to the proof as we argue on sampling the transcript conditioned on the edges.

For a randomly picked $(\vec{x}, \vec{y}) \sim \tilde{\mu}^{\otimes n}$, the divergence in consideration is indeed low with high probability by Markov. That is

$$\Pr_{(\vec{x},\vec{y})}\left[\sum_{i\in[n]} D(\Pi^W_{x_i y_i}||\Pi_{x_i}) > \alpha n\varepsilon\right] \leq O(1/\alpha) \tag{5}$$

Similarly, we get

$$\Pr_{(\vec{x},\vec{y})}\left[\sum_{i\in[n]} D(\Pi^W_{x_i y_i}||\Pi_{y_i}) > \alpha n\varepsilon\right] \leq O(1/\alpha) \tag{6}$$

Now, we consider the particular set of vectors $(\vec{x}, \vec{y})$ that satisfy

- $\forall i \in [n]$, (x_i, y_i) is a "good" edge.
- $D(\Pi^W_{\vec{x},\vec{y}}||\Pi_{\vec{x}}) = \sum_{i\in[n]} D(\Pi^W_{x_i y_i}||\Pi_{x_i}) \leq Kn\varepsilon$ and $D(\Pi^W_{\vec{x},\vec{y}}||\Pi_{\vec{y}}) = \sum_{i\in[n]} D(\Pi^W_{x_i y_i}||\Pi_{y_i}) \leq Kn\varepsilon$

which we denote as "good" vectors.

If $(\vec{x}, \vec{y}) \sim \mathcal{U}^{\otimes n}$, since (x_i, y_i) is good with probability at least $(1 - 2\varepsilon)$, all the coordinates are good with at least $(1 - 2\varepsilon)^n$ probability. If all the coordinates are good, by (5) and (6) and picking appropriately large K, $\Omega(1)$-fraction of such edges satisfy the second condition as well. In total, $2^{-\Omega(\varepsilon n)}$-fraction of edges satisfies both conditions, since we assume $n > 1/\varepsilon$.

Now we apply Lemma 25 to "good" vectors to complete the proof. In particular, Lemma 25 gives a zero-communication sampling protocol for transcript where Alice and Bob gets a matching transcript from $\mathsf{Supp}(\Pi^W_{\vec{x},\vec{y}})$ with probability at least $2^{-O(\varepsilon n)}$. Thus for $2^{-O(\varepsilon n)}$-fraction of the edges, we get a zero-communication strategy that wins with probability at least $2^{-O(\varepsilon n)}$, thus $\mathsf{val}(\mathcal{G}^n) > 2^{-O(\varepsilon n)}$. ◀

Acknowledgment. We thank Ankit Garg for many helpful discussions and comments on earlier version of this paper.

References

1. Boaz Barak, Ishay Haviv, Moritz Hardt, Anup Rao, Oded Regev, and David Steurer. Rounding parallel repetitions of unique games. In *Proceedings of the 49th Annual IEEE Symposium on Foundations of Computer Science*. IEEE Computer Society, 2008.
2. Boaz Barak, Anup Rao, Ran Raz, Ricky Rosen, and Ronen Shaltiel. Strong parallel repetition theorem for free projection games. *RANDOM*, 2009.
3. Mark Braverman and Ankit Garg. Small value parallel repetition for general games. 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 335–340. ACM, 2015. doi:10.1145/2746539.2746565.
4. 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.
5. André Chailloux and Giannicola Scarpa. Parallel repetition of entangled games with exponential decay via the superposed information cost. *41st International Colloquium on Automata, Languages and Programming*, 2014.
6. Richard Cleve, William Slofstra, Falk Unger, and Sarvagya Upadhyay. Perfect parallel repetition theorem for quantum xor proof systems. *Journal of Computational Complexity*, 17(2):282–299, May 2008.

7 Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory*. Wiley series in telecommunications. J. Wiley and Sons, New York, 1991.

8 Irit Dinur and David Steurer. Analytical approach to parallel repetition. *46th Annual Symposium on the Theory of Computing*, 2014.

9 Irit Dinur, David Steurer, and Thomas Vidick. A parallel repetition theorem for entangled projection games. *IEEE Conference on Computational Complexity*, 2014.

10 Thomas Holenstein. Parallel repetition: Simplifications and the no-signaling case. In *Proceedings of the 39th Annual ACM Symposium on Theory of Computing*, 2007.

11 Rahul Jain, Attila Pereszlényi, and Penghui Yao. A parallel repetition theorem for entangled two-player one-round games under product distributions. *IEEE Conference on Computational Complexity*, 2014.

12 Julia Kempe and Thomas Vidick. Parallel repetition of entangled games. *43rd annual ACM symposium on Theory of computing*, 2011.

13 Iordanis Kerenidis, Sophie Laplante, Virginie Lerays, Jérémie Roland, and David Xiao. Lower bounds on information complexity via zero-communication protocols and applications. *CoRR*, abs/1204.1505, 2012. URL: `http://arxiv.org/abs/1204.1505`.

14 Anup Rao. Parallel repetition in projection games and a concentration bound. In *Proceedings of the 40th Annual ACM Symposium on Theory of Computing*, 2008.

15 Ran Raz. A parallel repetition theorem. *SIAM Journal on Computing*, 27(3):763–803, jun 1998. Prelim version in STOC'95.

16 Ran Raz. A counterexample to strong parallel repetition. In *Proceedings of the 49th Annual IEEE Symposium on Foundations of Computer Science*. IEEE Computer Society, 2008.

17 Ran Raz and Ricky Rosen. A strong parallel repetition theorem for projection games on ex- panders. *IEEE Conference on Computational Complexity*, pages 247–257, 2012.

18 David Steurer. Improved rounding for parallel repeated unique games. In *Proceedings of the 13th International Conference on Approximation, and 14 the International Conference on Randomization, and Combinatorial Optimization: Algorithms and Techniques*, APPROX/RANDOM'10, pages 724–737, Berlin, Heidelberg, 2010. Springer-Verlag. URL: `http://dl.acm.org/citation.cfm?id=1886521.1886577`.

19 Madhur Tulsiani, John Wright, and Yuan Zhou. Optimal strong parallel repetition for projection games on low threshold rank graphs. *ICALP*, 2014.

20 Amir Yehudayoff. Pointer chasing via triangular discrimination. *TR16-151*, 2016.

21 Henry Yuen. A parallel repetition theorem for all entangled games. *CoRR*, abs/1604.04340, 2016. `arXiv:1604.04340`.

A Omitted Proof from Section 3

Proof of Lemma 24. Consider the set of edges $\mathcal{S}$ that are satisfied under μ. Denote μ conditioned on being inside $\mathcal{S}$ as ν. We show that $D(\nu||\tilde{\mu}) < O(\varepsilon)$, which indeed implies $\tilde{\mu}(\mathcal{S}) < 1 - O(\varepsilon)$ by Fact 10. But indeed Claim 26 implies $D(\nu||\tilde{\mu}) < O(\varepsilon)$. ◀

Proof of Lemma 25. The proof follows from Lemma 4.5 of [3]. To argue that the second condition is indeed met, recall that the protocol in [3] is the following:

Protocol 2 Protocol for sampling a transcript (π, q)

- Using shared randomness, get uniformly random samples from $\Pi \times [0,1]$, which we denote by $\{(\pi_i, q_i)\}_{i=0}^{\infty}$.
- Alice outputs the first π_a that satisfies $q_a < \frac{P(\pi_a)}{\delta}$.
- Bob outputs the first π_b that satisfies $q_b < \frac{Q(\pi_b)}{\delta}$.

Let $\mathcal{A} := \{(\pi, q) | q < P(\pi)/\delta\}$, $\mathcal{B} := \{(\pi, q) | q < Q(\pi)/\delta\}$ and $\mathcal{C} := \{(\pi, q) | q < R(\pi)\}$. We define event E as first dart in $\mathcal{A} \cup \mathcal{B}$ being inside $\mathcal{A} \cap \mathcal{B} \cap \mathcal{C}$. Note that Lemma 4.5. of [3] exactly gives that $\Pr[E] \geq 2^{-\Omega(\delta)}$, therefore the first condition must hold. Furthermore, $\mathcal{S}$ is indeed included in $\mathsf{Supp}(R)$ since for such event it must be the case that $R(\pi) > 0$. Therefore, E satisfies the second condition as well. ◀

Proof of Claim 26. For brevity denote W as the set of x with $W(x) = 1$, $D(A||B) = \delta_0$ and $A(\overline{W}) = \delta_1$. First, we show that $B(\overline{W}) < O(D(A||B)) = O(\delta_0)$. Note that $D(A||B)$ can be written as

$$D(A||B) = \sum_{x \in W} A(x) \log \frac{A(x)}{B(x)} + \sum_{x \notin W} A(x) \log \frac{A(x)}{B(x)} = \delta_0 \tag{7}$$

Applying log-sum, we get

$$A(W) \log \frac{A(W)}{B(W)} + A(\overline{W}) \log \frac{A(\overline{W})}{B(\overline{W})} \leq \delta_0$$

Assume for contradiction that $B(\overline{W}) > KA(\overline{W})$, where K is some parameter that we will setup later for contradiction. And put $B(\overline{W}) = \alpha A(\overline{W})$ for $\alpha > K$. Since $A(W) = 1 - \delta_1$ by our assumption, substituting the terms we get

$$(1 - \delta_1) \log \frac{1 - \delta_1}{1 - \alpha\delta_1} + \delta_1 \log \frac{1}{\alpha} = (1 - \delta_1) \log \frac{1}{1 - \frac{(\alpha-1)\delta_1}{1-\delta_1}} + \delta_1 \log \frac{1}{\alpha}$$

Suppose for now that $\alpha < \frac{1}{2\delta_1} + \frac{1}{2}$. Then note that $\log(1/(1-x)) = \Omega(x)$. Applying this fact to the first term, we get

$$\delta_0 > (1 - \delta_1) \log \frac{1 - \delta_1}{1 - \alpha\delta_1} + \delta_1 \log \frac{1}{\alpha} > \Omega((\alpha - 1 - \log \alpha)\delta_1) > \Omega(\alpha\delta_0).$$

where the last inequality holds due to our assumption that $\delta_1 > \delta_0$. Picking an appropriately large constant $\alpha = O(1)$, we get a contradiction. Instead if $\alpha > \frac{1}{2\delta_1} + \frac{1}{2}$, then

$$(1 - \delta_1) \log \frac{1 - \delta_1}{1 - \alpha\delta_1} + \delta_1 \log \frac{1}{\alpha} > (1 - \delta_1) \log 2(1 - \delta_1) + \delta_1 \log 2\delta_1 = H(\delta_1) + 1 > \delta_0$$

which indeed is a contradiction since $\delta_0 < 1$ and $H(\delta_1) \geq 0$. Thus $B(\overline{W}) < O(\delta_1) = K_0\delta_1$.

To bound $D(A_W||B)$, we first bound $\sum_{x \in W} A(x) \log \frac{A(x)}{B(x)}$. Note that via our bound on $B(\overline{W})$ implies

$$\sum_{x \notin W} A(x) \log \frac{A(x)}{B(x)} > A(\overline{W}) \log \frac{A(\overline{W})}{B(\overline{W})} > \delta_1 \log \frac{1}{K_0}$$

Applying this to (7), we have

$$\sum_{x \in W} A(x) \log \frac{A(x)}{B(x)} = \delta_0 - \sum_{x \notin W} A(x) \log \frac{A(x)}{B(x)} < \delta_0 - \delta_1 \log \frac{1}{K_0}$$

Now we use the above fact to bound $D(A_W||B)$.

$$D(A_W||B) = \sum_x a_W(x) \log \frac{a_W(x)}{b(x)} = \sum_{x \in W} a_W(x) \log \frac{a_W(x)}{a(x)} \frac{a(x)}{b(x)}$$

$$= D(A_W||A) + \sum_{x\in W} a_W(x)\log\frac{a(x)}{b(x)} \le \log(1/A(W)) + \sum_{x\in W}\frac{a(x)}{A(W)}\log\frac{a(x)}{b(x)}$$
$$< \log(1/A(W)) + \frac{D(A||B)}{A(W)} + \frac{1-A(W)}{A(W)}\log K_0$$
$$< \log(1/A(W)) + \frac{D(A||B)}{A(W)} + O\left(\frac{1-A(W)}{A(W)}\right)$$

which is indeed the statement of the claim. ◀

Equilibrium Selection in Information Elicitation without Verification via Information Monotonicity*

Yuqing Kong[†1] **and Grant Schoenebeck**[‡2]

1 **University of Michigan, Ann Arbor, USA**
yuqkong@umich.edu
2 **University of Michigan, Ann Arbor, USA**
schoeneb@umich.edu

Abstract

In this paper, we propose a new mechanism – the Disagreement Mechanism – which elicits privately-held, non-variable information from self-interested agents in the single question (peer-prediction) setting.

To the best of our knowledge, our Disagreement Mechanism is the first strictly truthful mechanism in the single-question setting that is simultaneously:

- **Detail-Free:** does not need to know the common prior;
- **Focal:** truth-telling pays strictly higher than any other symmetric equilibria excluding some unnatural permutation equilibria;
- **Small group:** the properties of the mechanism hold even for a small number of agents, even in binary signal setting. Our mechanism only asks each agent her signal as well as a forecast of the other agents' signals.

Additionally, we show that the focal result is both tight and robust, and we extend it to the case of asymmetric equilibria when the number of agents is sufficiently large.

1998 ACM Subject Classification J.4 Social and Behavioral Sciences

Keywords and phrases peer prediction, equilibrium selection, information theory

Digital Object Identifier 10.4230/LIPIcs.ITCS.2018.13

1 Introduction

User feedback requests (e.g Ebay's reputation system and the innumerable survey requests in one's email inbox) are increasingly prominent and important. However, the overwhelming number of requests can lead to low participation rates, which in turn may yield unrepresentative samples. To encourage participation, a system can reward people for answering requests. But this may cause perverse incentives: some people may answer a large of number of questions simply for the reward and without making any attempt to answer accurately. In this case, the reviews the system obtains may be inaccurate and meaningless. Moreover, people may be motivated to lie when they face a potential loss of privacy or can benefit in the future by lying now.

It is thus important to develop systems that motivate honesty. If we can verify the information people provide in the future (e.g via prediction markets), we can motivate

* A full version of the paper is available at [15], https://arxiv.org/abs/1603.07751

† The author was supported by National Science Foundation Career Award 1452915 and CCF Award 1618187.

‡ The author was supported by National Science Foundation Career Award 1452915 and Algorithms in the Field Award 1535912.

honesty via this future verification. However, sometimes we need to elicit information without verification since the objective truth is hard to access or even does not exist (e.g. a self-report survey for involvement in crime). In our paper, we focus on the situation where the objective truth is not observable.

One important framework for designing incentive systems without verification is *peer prediction* [19]. *Peer prediction* uses each person's information to predict other people's information and pays according to how good the prediction is. *Peer prediction* assumes people's information is related and the systems and the people share a common prior. In the peer prediction mechanism, if an agent believes everyone else tells the truth, the best strategy to maximize her expected payment is telling the truth as well. In other words, peer prediction is *truthful* in the sense it has truth-telling as an equilibrium.

A series of work [20, 27, 21, 22, 30, 24, 7, 28, 29, 26, 11, 12, 16] extends peer prediction to incorporate some other desired properties in addition to *truthful* property, which we highlight here.

Detail-Free: A detail free mechanism does not need to know the common prior which is a clear advantage if deploying these mechanisms in the real world.

Focal: We hope the mechanism can pay the truth-telling equilibrium strictly more than other equilibria in expectation. However, we show this is not technically possible in the detail-free setting, and so instead we say *a mechanism is focal if truth-telling pays strictly higher than any other symmetric equilibria excluding some unnatural permutation equilibria* (see definition for permutation equilibria in Section 2). We emphasize that the *focal* property is very important, since in a non-focal mechanism, non-informative and effortless equilibria like everyone reporting the a priori most likely answer may pay equally or even much better than the truth-telling equilibrium, possibly incentivizing agents to coordinate on effortless and non-informative equilibria. Recent research [9] indicates that individuals in lab experiments do not always truth-tell when faced with peer prediction mechanisms; this may in part be related to the issue of equilibrium multiplicity.

Small group: The properties of the mechanism hold even when the number of agents is small.

General Informative Symmetric Common Prior: The mechanism should make minimal assumptions on the common prior. It is required that the prior be "informative" (that is each agents' signals contain stochastic information about the other agent's signals).

Finite Signals: The mechanism should only assume that the number of signals is finite, not binary.

Another desirable mechanism design property is the minimal property. A mechanism is minimal if it only requires agents to report their information rather than forecasts for other agents' reports. However, no detail free, strictly truthful mechanism can be minimal[1] [21]. We will introduce several strictly truthful, minimal mechanisms (not detail free) in the related work. Our mechanism is not minimal since it is detail free and strictly truthful.

Our Contributions

To the best of our knowledge, our Disagreement Mechanism is the first mechanism in the single-question setting that simultaneously has the *small group*, *focal*, *strictly truthful* and

[1] There exists detail free, strictly truthful mechanism that is "almost" minimal. For example, Riley [24] designs strictly truthful, detail free mechanism with optional prediction report.

Table 1 Detail Free Multi-Signal Single-Question Mechanisms Comparison

	Strictly Truthful	Small group	Focal	General Symmetric Prior
Bayesian Truth Serum [20]	×		×	⧸[a]
Logarithmic PTS [23]	×		×	⧸[a]
Multi-Signal SM [26]	×	×		
Multi-Valued RBTS [21]	×	×		
Minimal Truth Serum [24]	×	×		×
Divergence-Based BTS [22]	×	×		×
Disagreement Mechanism	×	×	×	×

[a] BTS requires an additional assumption – conditioning on the state, the signals are independently assigned to agents. But this conditional independence assumption is very natural in this literature, thus we still put ⧸ here.

detail free properties (see Table 1), even in binary signal setting. Moreover, our Disagreement Mechanism can be applied to a general family of symmetric prior, and

1. we show the *focal* property is **tight** in the sense that for the set of symmetric equilibria (what we call permutation equilibria) that have the same expected payment with truth-telling in our Disagreement Mechanism, no detail free, truthful mechanism can pay truth-telling strictly higher than permutation equilibria.
2. we show the *focal* property is **robust** in the sense that any symmetric equilibrium that has expected payoff close to truth-telling must be "close" to a permutation equilibrium;
3. we extend this *focal* property to asymmetric equilibria when the number of agents is sufficiently large in the sense that any asymmetric equilibrium which is "close" to a permutation equilibrium has expected payoff "close" to that of the truth-telling equilibrium, and any equilibrium that is not "close" to a permuation equilibrium pays strictly less then the truth-telling equilibrium.

The permutation equilibria are intuitively unnatural and risky as they require extreme coordination amongst the agents and as much effort as truth-telling. For *symmetric* equilibria that are not permutation equilibrium, **every agent's** expected payment is strictly less than the expected payment she obtains when everyone tells the truth. Thus our results about symmetric equilibrium are quite strong, despite the impossibility result. Asymmetric equilibrium require more coordination between agents than symmetric equilibrium. Additionally, the possible *total* gains from doing so are limited and go to zero as the number of agents increase.

High Level Techniques

Our *Disagreement Mechanism* pays agents individually (locally) for "agreement" and globally for "disagreement". When agents collude, they share information about their strategies. Since they are paid individually for "agreement", they will use their information about other people's strategies and "agree to agree" which reduces their global payoff which depends on "disagreement".

Essentially, when agents collude, our *Disagreement Mechanism* encourages each agent to implicitly admit their collusion by unilaterally increasing their individual payoff for doing so, but the mechanism then simultaneously decreases the total payoffs to all agents. Only when agents do not choose to collude and lack information about other people's strategies,

can they "agree to disagree." In this case, even when they maximize their individual payoff, globally they still have a lot of disagreement.

Technical Contributions

In addition to the above results, our works has several contributions in the techniques employed:

1. Our *Disagreement Mechanism* encourages not only agents with the same private information to agree, but also agents with different private information to disagree. We present a novel way to measure the amount of "information" by casting the reports as multiple labelled points in a space and measuring the quality of the "classification".
2. To show that the "classification" quality always decreases with non-truthful equilibria, we exploit tools from information theory, namely ***Information Monotonicity***. Despite their natural and powerful application, to our knowledge, this is the first time such tools have been explicitly employed in the peer prediction literature.

1.1 Related Work

After Miller et al. [19] introducing peer prediction, a host of results (see, e.g., [20, 27, 21, 22, 30, 24, 7, 28, 29, 26, 11, 12, 16]) have followed. In this section, we will introduce them and classifies them into several categories according to the properties they (do not) have.

(1) Single-question, detail free, focal (not small group): *Bayesian Truth Serum (BTS)* [20] first successfully weakened the known common prior assumption (detail free) and solves the equilibrium multiplicity issue (focal). Prelec [20] also provides an important framework for mechanisms without known common prior. BTS requires the agents report – in addition to their reported signal – a forecast (prediction) of the other agents' reported signals, and uses this predictions in lieu of the common prior. BTS incentives agents to report accurate forecasts by rewarding forecasts that have the ability to predict the other agents' reported signal. However, BTS has two weakness: (1) BTS requires that the number of agents goes to infinity (or is large enough in a modified version) since the mechanism needs agents to believe it has access to the true distribution from which agents' signals are drawn. (2) The analysis of non-truthful equilibria provided in [20] requires that the number of agents goes to infinity and only proves that truth-telling has total expected payment at least as high as other equilibrium. Specifically, it does not rule out the existence of many other equilibrium which are all paid the same as the truth-telling equilibrium.

Logarithmic Peer Truth Serum (PTS) [23] extends BTS to a slightly different setting involving sensors, but still requires a large number of agents.

(2) Single-question, small group, detail free (not focal): Several mechanisms [27, 21, 22, 24, 28, 29, 26] are based on the BTS framework and address the first weakness of BTS. *Robust Bayesian Truth Serum (RBTS)* [27] is a mechanism which can only be applied to binary signals. *Multi-Valued RBTS* [21] and *Multi-Signal Shadowing Method (Multi-Signal SM)* [26] can be applied to non-binary signals while they require an *additional assumption* that an agent will think the probability that other agents receive signal σ higher if he himself also receives σ. *Divergence-based BTS* [22] can be applied to non-binary signals and does not require additional assumptions on the prior. All of those works do not solve the equilibrium multiplicity issue, but do work for a small number of agents. *Minimal Truth Serum (MTS)* [24] is a mechanism where agents have the option to report or not report their predictions, and also lacks analysis of non-truthful equilibria. MTS uses a typical zero-sum technique such that all equilibria are paid equally. In contrast, we show that

in our *Disagreement Mechanism* any equilibrium that is even close to paying more than the truth-telling equilibrium must be close to a small set of permutation equilibrium. The *Divergence based BTS* only requires the common prior assumption to be truthful. Because of its generality, we use it as a building block in our Disagreement Mechanism. However, the *Divergence based BTS* contains effortless equilibrium that pay significantly more than truth-telling. Moreover, analyzing the set of equilibria in *Divergence-based BTS* is very complicated and becomes a main technical obstacle in our paper. Thus, while the above work addresses the first weakness of BTS, it exacerbates the second.

(3) Single-question, small group, minimal (not detail free): Jurca and Faltings [11, 12] use algorithmic mechanism design to build their own peer prediction style mechanism where truth-telling is paid strictly better than non-truthful pure strategies but leave the analysis of mixed strategies as a open question. Frongillo and Witkowski [8] consider the design for robust, truthful and minimal peer prediction mechanisms with the prior knowledge and lack the analysis of non-truthful equilibria. Kong et al. [16] modify the peer prediction mechanism such that truth-telling is paid strictly better than any other non-truthful equilibrium. Thus, the mechanism designed in Kong et al. [16] is focal. Additionally, they optimize the cost their mechanism needs over a natural space. The assumption that the mechanism knows the prior, allows these mechanisms to not ask for a prediction report. However, unlike the current work, the mechanism still needs to know the prior and the analysis only works for the case of binary signals.

(4) Different Settings: Several works are not in the traditional one-question setting. Some of these works make the additional assumption of multiple a priori similar questions so that the mechanism need not explicitly ask the agents for a prediction. The mechanism in Dasgupta and Ghosh [6] uses the presence of multiple questions to elicit agent strategies with high effort, addressing the equilibrium multiplicity issue for binary signals in their setting. Recently Kong and Schoenebeck [14] and Shnayder et al. [25] independently extend the mechanism in [6] to non-binary signal setting but still require the presence of multiple questions. Our setting is different since the agents only have one question (and thus we do not have to assume relations between questions) and our results for equilibrium multiplicity issue are robust to non-binary signals.

In addition to the multiple questions setting, there are many other works in the settings that are different from our results. For example, Cai et al. [4] and Liu and Chen [17] consider the machine learning setting. Kamble et al. [13], Mandal et al. [18], and Agarwal et al. [1] consider the heterogeneous setting in the multiple questions setting. Zhang and Chen [30] consider a sequential game. Faltings et al. [7] consider a setting where they have an estimation of the public distribution of previous answers on other a priori similar questions.

2 Preliminaries, Background, and Notation

We recommend that the eager reader skip to the end of Section 2.4 where Hellinger Divergence is discussed and then refer back to the earlier preliminaries only as needed. Section 3 states the main theorem and outlines the technical contributions.

See notation table in Appendix A.

2.1 Prior Definitions and Assumptions

We consider a setting with n agents and a set of signals Σ, and define a *setting* as a tuple (n, Σ). Each agent i has a private signal $\sigma_i \in \Sigma$ chosen from a joint distribution Q over Σ^n called the prior. Given a prior Q, for $\sigma \in \Sigma$, let $q_i(\sigma) = \Pr_Q[\sigma_i = \sigma]$ be the *a priori*

probability that agent i receives signal σ. Let $q_{j,i}(\sigma'|\sigma) = \Pr_Q[\sigma_j = \sigma'|\sigma_i = \sigma]$ be the probability that agent j receives signal σ given that agent i received signal σ'.

We say that a prior Q over Σ is *symmetric* if for all σ, $\sigma' \in \Sigma$ and for all pairs of agents $i \neq j$ and $i' \neq j'$ we have $q_i(\sigma) = q_{i'}(\sigma)$ and $q_{i,j}(\sigma|\sigma') = q_{i',j'}(\sigma|\sigma')$. That is, the first two moments of the prior do not depend on the agent identities. Because we will assume that the prior is symmetric, we denote $q_i(\sigma)$ by $q(\sigma)$ and $q_{i,j}(\sigma|\sigma')$ (where $i \neq j$) by $q(\sigma|\sigma')$. We also define $\mathbf{q}_\sigma = q(\cdot|\sigma)$. We assume the common prior shared by agents is

symmetric: we assume throughout that the agents' signals $\boldsymbol{\sigma}$ are drawn from some joint **symmetric prior** Q;

non-zero: for any $\sigma, \sigma' \in \Sigma$, $q(\sigma) > 0, q(\sigma|\sigma') > 0$;

informative: we assume if agents have different private signals, they will have different expectations for the fraction of at least one signal. That is for any $\sigma \neq \sigma'$, there exists σ'' such that $q(\sigma''|\sigma) \neq q(\sigma''|\sigma')$;

fine-grained: this assumption conceptually states that one state is not just a more likely version of another state. While we defer the exact definition to the full version, a slightly stronger version of this assumption is that $q(\sigma|\cdot)$ are linearly independent. This additional fine-grained assumption is required only when we need to show truth-telling is *strictly* better than other symmetric equilibria that are not permutation equilibria;

ensemble: the first two moments of the prior are fixed as the number of agents increases,

We sometimes will denote the class of priors that satisfy all five of these assumptions as SNIFE priors (see formal definitions of the above SNIFE prior assumptions in Appendix B.).

2.2 Game Setting and Equilibrium Concepts

Given a setting (n, Σ) with prior Q, we consider a game in which each agent i is asked to report his private signal $\sigma_i \in \Sigma$ and his prediction $\mathbf{p}_i \in \Delta_\Sigma$, a distribution over Σ, where $\mathbf{p}_i = \mathbf{q}_{\sigma_i}$. For any $\sigma \in \Sigma$, $\mathbf{p_i}(\sigma)$ is agent i's (reported) expectation for the fraction of other agents who has received σ given he has received σ_i. However, agents may not tell the truth. We denote the spaces of reports as $\Sigma \times \Delta_\Sigma$ by $\mathcal{R}$. We define a report profile of agent i as $r_i = (\hat{\sigma}_i, \hat{\mathbf{p}}_i) \in \mathcal{R}$ where $\hat{\sigma}_i$ is agent i's reported signal and $\hat{\mathbf{p}}_i$ is agent i's reported prediction.

We would like to encourage **truth-telling**, namely that agent i reports $\hat{\sigma}_i = \sigma_i, \hat{\mathbf{p}}_i = \mathbf{q}_{\sigma_i}$. To this end, agent i will receive some payment $\nu_i(\hat{\sigma}_i, \hat{\mathbf{p}}_i, \hat{\sigma}_{-i}, \hat{\mathbf{p}}_{-i})$ from our mechanism where $(\hat{\sigma}_{-i}, \hat{\mathbf{p}}_{-i})$ are all agents' report profiles excluding agent i.

We define **strategy** as a mapping from each possible signal σ and prior Q to a distribution over the report profile space.

We define a **strategy profile s** as a profile of all agents' strategies $\{s_1, s_2, ...s_n\}$ and we say agents play **s** if for any i, agent i plays strategy s_i. We say a strategy profile is *symmetric* if each agent plays the same strategy. Assuming a fixed prior Q, for any strategy profile $\boldsymbol{s} = (s_1, s_2, ..., s_n)$, we will represent the marginal distribution of an agent i's strategy for her signal report as a matrix θ_i where $\theta_i(\hat{\sigma}, \sigma)$ is the probability that agent will report signal $\hat{\sigma}$ when his private signal is σ. Note that θ_i is a *transition matrix*, since its entries are all non-negative and the sum of its every column is 1. We call θ_i the **signal strategy** of agent i. We also call $(\theta_1, \theta_2, ..., \theta_n)$ the signal strategy of $\boldsymbol{s}$. In the symmetric case, we call θ the signal strategy of $\boldsymbol{s}$. We say a signal strategy θ is τ**-close** to a permutation matrix if for any row of θ, there is at most one entry that is greater than τ.

We informally define a **permutation strategy profile** as a strategy profile where agents "collude" to relabel the signals and then tell the truth with respect to the relabelled signals. When agents play a permutation strategy profile, they play the same signal strategy which is

a permutation matrix θ_π. We defer the formal definition of the permutation strategy profile to the full version. We will show that if truth-telling is an equilibrium, then all permutation strategy profiles are equilibria as well. Thus we sometimes call these permutation strategy profiles **permutation equilibria**. We define the **agent welfare** of a strategy profile $\mathbf{s}$ and a mechanism $\mathcal{M}$ for setting (n, Σ) with prior Q to be the expectation of the sum of payments to each agent and we write it as $AW_{\mathcal{M}}(n, \Sigma, Q, \mathbf{s})$. Note that for symmetric strategy profile, the **agent welfare** is proportional to each agent's expected payment since everyone plays the same strategy. A **Bayesian Nash equilibrium** consists of a strategy profile $\boldsymbol{s} = (s_1, \ldots, s_n)$ such that no player wishes to change her strategy, given the strategies of the other players and the information contained in the prior and her signal. See formal definitions in the full version of this paper.

2.3 Mechanism Design Tools: f-divergence and Proper Scoring Rules

Now we introduce f-divergence and strictly proper scoring rules, which are two of the main tools we will use in our mechanism design. f-divergence ([2, 5]) is always used in measuring the "difference" between distributions. One important property of the f-divergence family is information monotonicity: for any two distributions, if we use the same way to post-process each distribution, the two distributions will become "closer" because of potential information loses.

f-divergence

f-divergence [2, 5] $D_f : \Delta_\Sigma \times \Delta_\Sigma \to \mathbb{R}$ is a non-symmetric measure of difference between distribution $\mathbf{p} \in \Delta_\Sigma$ and distribution $\mathbf{q} \in \Delta_\Sigma$ and is defined to be

$$D_f(\mathbf{p}, \mathbf{q}) = \sum_{\sigma \in \Sigma} \mathbf{p}(\sigma) f\left(\frac{\mathbf{p}(\sigma)}{\mathbf{q}(\sigma)}\right)$$

where $f(\cdot)$ is a convex function.

We introduce three properties of f-divergence:

(1) Information Monotonicity [5, 3]: For any $\mathbf{p}, \mathbf{q}$, and transition matrix $\theta \in \mathbb{R}^{|\Sigma| \times |\Sigma|}$ where $\theta(\sigma, \sigma')$ is the probability that we map σ' to σ, we have $D(\mathbf{p}, \mathbf{q}) \geq D^*(\theta\mathbf{p}, \theta\mathbf{q})$. When θ is a permutation matrix θ_π, $D(\mathbf{p}, \mathbf{q}) = D(\theta_\pi\mathbf{p}, \theta_\pi\mathbf{q})$. When θ is not a permutation, the inequality is strict if $\mathbf{p}$ and $\mathbf{q}$ satisfy some weak conditions. The weak conditions are closely related to the definition of fine-grained prior (see Appendix C for more details.).

(2) Non-negative [5] For any $\mathbf{p}, \mathbf{q}$, $D_f(\mathbf{p}, \mathbf{q}) \geq 0$ and $D_f(\mathbf{p}, \mathbf{q}) = 0$ if and only if $\mathbf{p} = \mathbf{q}$.

(3) Convexity [5]: Both $D_f(\cdot, \mathbf{q})$ and $D_f(\mathbf{p}, \cdot)$ are convex functions for any $\mathbf{p}, \mathbf{q}$.

Hellinger-divergence

If we pick the convex function $f(\cdot)$ as $(\sqrt{x} - 1)^2$, we will obtain Hellinger-divergerce [5]

$$D^*(\mathbf{p}, \mathbf{q}) = \sum_{\sigma} (\sqrt{\mathbf{p}(\sigma)} - \sqrt{\mathbf{q}(\sigma)})^2$$

Thus Hellinger-divergence is a type of f-divergence.

In addition to the above three properties, we highlight two important properties Hellinger-divergence has:

(4) Square root triangle inequality [5]: $|\sqrt{D^*(\mathbf{p}, \mathbf{q})} - \sqrt{D^*(\mathbf{p}, \mathbf{q}')}| < \sqrt{D^*(\mathbf{q}', \mathbf{q})}$ for any $\mathbf{p}, \mathbf{q}, \mathbf{q}'$

(5) Bounded divergence [5]: $0 \leq D^*(\mathbf{p}, \mathbf{q}) \leq 1$

Proper Scoring Rules

Now we introduce strictly proper scoring rules, another key tool we will use in our mechanism design. Starting with [19], proper scoring rules have become a common ingredient in mechanisms for unverifiable information elicitation (e.g. [20, 27]).

A scoring rule $PS : \Sigma \times \Delta_\Sigma \to \mathbb{R}$ takes in a signal $\sigma \in \Sigma$ and a distribution over signals $\delta_\Sigma \in \Delta_\Sigma$ and outputs a real number. A scoring rule is *proper* if, whenever the first input is drawn from a distribution δ_Σ, then the expectation of PS is maximized by δ_Σ. A scoring rule is called *strictly proper* if this maximum is unique. We will assume throughout that the scoring rules we use are strictly proper. By slightly abusing notation, we can extend a scoring rule to be $PS : \Delta_\Sigma \times \Delta_\Sigma \to \mathbb{R}$ by simply taking $PS(\delta_\Sigma, \delta'_\Sigma) = \mathbb{E}_{\sigma \leftarrow \delta_\Sigma}(\sigma, \delta'_\Sigma)$. We note that this means that any proper scoring rule is linear in the first term.

▶ **Example 1** (Example of Proper Scoring Rule)**.** Fix an outcome space Σ for a signal σ. Let $\mathbf{q} \in \Delta_\Sigma$ be a reported distribution. The Logarithmic Scoring Rule maps a signal and reported distribution to a payoff as follows:

$L(\sigma, \mathbf{q}) = \log(\mathbf{q}(\sigma))$.

Let the signal σ be drawn from some random process with distribution $\mathbf{p} \in \Delta_\Sigma$.
Then the expected payoff of the Logarithmic Scoring Rule

$$\mathbb{E}_{\sigma \leftarrow \mathbf{p}}[L(\sigma, \mathbf{q})] = \sum_\sigma \mathbf{q}(\sigma) \log \mathbf{q}(\sigma) = L(\mathbf{p}, \mathbf{q})$$

According to [10], this value will be maximized if and only if $\mathbf{q} = \mathbf{p}$.

Proper scoring rules are a key tool in the design of mechanisms [20] in the BTS framework. In such mechanism, agents are asked to report their private information and forecast for other agents and paid based on a "prediction score" and an "information score". The prediction score is usually calculated by a proper scoring rule and the information score is customized.

Prediction Score via Proper Scoring Rules. Agents will receive a prediction score based on how well their prediction predicts a randomly chosen agent's reported signal. Say an agent i reports prediction $\hat{\mathbf{p}}_i$ then a random agent, call him agent j, is chosen, agent i will receive a prediction score $PS(\hat{\sigma}_j, \hat{\mathbf{p}}_i)$ where PS is a proper scoring rule. Note that any proper scoring rule works. $PS(\hat{\sigma}_j, \hat{\mathbf{p}}_i)$ is maximized if and only if agent i's reported prediction $\hat{\mathbf{p}}_i$ is his expected likelihood for $\hat{\sigma}_j$. Agent i cannot pretend to have a different expected likelihood without reducing his expectation for his prediction score.

3 The Disagreement Mechanism

3.1 Buiding Block – Divergence-Based BTS

In this section, we introduce a building block of our Disagreement Mechanism – Divergence-Based BTS [22]. It follows the BTS framework and still pays agents an "information score" and a "prediction score". The main idea of Divergence-Based BTS is that the mechanism punishes the **Inconsistency** of agents – the "difference" between two random agents' predictions when they report the same signal. The common prior assumption tells us agents cannot agree to disagree. That is, if two agents receive the same private information, they must have the same "belief" about the world. In our setting, if agents tell the truth, whenever two agents report the same signal, they will report the same prediction as well. Thus, everyone telling the truth is a consistent strategy. Since Divergence-Based BTS punishes inconsistency, the truth-telling strategy will be encouraged in Divergence-Based BTS.

Divergence-Based BTS [22] $\mathcal{M}$:

Let $\alpha, \beta > 0$ be parameters and let PS be a strictly proper scoring rule, then we define $\mathcal{M}(\alpha, \beta, PS)$[2] as follows:

1. Each agent i reports a signal and a prediction $r_i = (\hat{\sigma}_i, \hat{\mathbf{p}}_i)$
2. For each agent i and agent j, we define a prediction score that depends on agent i's prediction and agent j's report signal

   $$score_P(r_i, r_j) = PS(\hat{\sigma}_j, \hat{\mathbf{p}}_i),$$

   and an information score

   $$score_I(r_i, r_j) = \begin{cases} 0 & \hat{\sigma}_i \neq \hat{\sigma}_j \\ -(PS(\hat{\mathbf{p}}_j, \hat{\mathbf{p}}_j) - PS(\hat{\mathbf{p}}_j, \hat{\mathbf{p}}_i)) & \hat{\sigma}_i = \hat{\sigma}_j \end{cases}$$

3. Each agent i is matched with a random agent j. The payment for agent i is

   $$payment_{\mathcal{M}(\alpha,\beta,\mathcal{PS})}(i, \mathbf{r}) = \alpha score_P(r_i, r_j) + \beta score_I(r_i, r_j).$$

▶ **Theorem 2.** *[22] For any $\alpha, \beta > 0$ and any strictly proper scoring rule PS, $\mathcal{M}(\alpha, \beta, PS)$ has truth-telling as a strict Bayesian-Nash equilibrium whenever the prior Q is informative and symmetric.*

Main Drawback of Divergence-Based BTS

The main drawback is that there may be many other equilibria that have the same payoff with truth-telling. Agents can simply report the a priori most popular signal and predict that everyone does the same. This strategy is a consistent equilibrium and gives agents the maximum possible payoff since their predictions are perfect. In particular, for any non-trivial prior, this strategy pays *strictly more* than the truth-telling equilibrium – so that it Pareto dominates truth-telling.

The above extreme example provides a effortless and meaningless equilibrium but is preferred by agents in Divergence-Based BTS. To deal with this problem, one key observation is that in the meaningless equilibrium mentioned above, the unitary predictions implies their report profiles have little information. At a high level, the "disagreement" between agents represents the amount of information their report profiles have. Motivated by this extreme example, we design a new mechanism – the Disagreement Mechanism – that encourages "disagreement".

3.2 The Disagreement Mechanism and Main Theorem

In this section, we will describe our Disagreement Mechanism and state our main theorem. To design our mechanism, we start with the Divergence-Based BTS and (a) first use a typical trick to create a zero-sum game which has the same equilibria as the Divergence-Based BTS; (b) pay each agent an extra score that only depends on other agents which will not change the structure of the equilibria. We want this extra score to represent "classification score" (See Figure 1).

[2] This mechanism is a little bit different from Divergence-Based BTS mechanism [22]. Divergence-Based BTS uses specific proper scoring rule (log scoring rule). But it is easy to see using general proper scoring rules still keeps the strictly truthful property of Divergence-Based BTS. We defer the proof to the full version.

Figure 1 Illustration for Classification Score: Each point represents an agent's report profile – the *color* represents the *signal* the agent reports; the *position* represents the *prediction* the agent reports. We informally define **Inconsistency** as the *average* disagreement between every two agents' predictions when they report the *same* signal and **Diversity** as the *average* disagreement between every two agents' predictions when they report *different* signals. We informally define **Classification Score** as Diversity minus Inconsistency. Note that the report profiles in the right figure will have a much higher classification score than those in the left figure since the right figure has high Diversity and low Inconsistency.

Disagreement Mechanism $\mathcal{M}^+(\alpha, \beta, PS(\cdot,\cdot))$

$\mathbf{r} = \{r_1, r_2, ..., r_n\}$ is all agents' report profiles where for any r, $r_i = (\hat{\sigma}_i, \hat{\mathbf{p}}_i)$.

1. *Zero-sum Trick*: Divide the agents arbitrarily into two groups – group A and group B – such that both A and B have at least 3 agents. Each group of agents plays the game (mechanism) $\mathcal{M}$ that is restricted in their own group. For group A, each agent i_A receives a

$$score_{\mathcal{M}}(i_A, \mathbf{r}) = payment_{\mathcal{M}(\alpha,\beta,PS(\cdot,\cdot))}(i_A, \mathbf{r}_A) - \frac{1}{|A|}\sum_{j_B \in B} payment_{\mathcal{M}(\alpha,\beta,PS(\cdot,\cdot))}(j_B, \mathbf{r}_B)$$

   Where $payment_{\mathcal{M}(\alpha,\beta,PS(\cdot,\cdot))}(i_A, \mathbf{r}_A)$ is agent i_A's payment when he is paid by mechanism $\mathcal{M}(\alpha, \beta, PS(\cdot,\cdot))$ given group A's report profiles $\mathbf{r}_A$ and that he can only be paired with a random peer from group A (we have similar explanation for $payment_{\mathcal{M}(\alpha,\beta,PS(\cdot,\cdot))}(j_B, \mathbf{r}_B)$). For agents in group B, we use the analogous way to score them.
2. *Additional Classification Reward*: Each agent i is matched with two random agents $j, k \neq i$ chosen from all agents (including group A and group B), the payment for agent i is

$$payment_{\mathcal{M}^+(\alpha,\beta,PS(\cdot,\cdot)}(i, \mathbf{r}) = score_{\mathcal{M}}(i, \mathbf{r}) + score_C(r_j, r_k)$$

   where

$$score_C(r_j, r_k) = \begin{cases} D^*(\hat{\mathbf{p}}_j, \hat{\mathbf{p}}_k) & \hat{\sigma}_j \neq \hat{\sigma}_k \\ -\sqrt{D^*(\hat{\mathbf{p}}_j, \hat{\mathbf{p}}_k)} & \hat{\sigma}_j = \hat{\sigma}_k \end{cases}$$

   recall that D^* denotes the Hellinger Divergence.

▶ **Theorem 3** (Main Theorem). *For any number of signals m, given any SNIFE prior, if the number of agents $n \geq 6$, then in $\mathcal{M}^+(\alpha, \beta, PS(\cdot,\cdot))$ with $\frac{\alpha}{\beta} < \frac{1}{4m}$,*

1. *(*Strictly truthful*) truth-telling is a strict Bayesian Nash equilibrium;*
2. *(*Focal*) in any permutation equilibrium,* every agent *has equal expected payment with truth-telling; and in any* symmetric *equilibrium that is not a permutation equilibrium,* every agent*'s expected payment is strictly less than that of truth-telling.*
3. *(*Robust Focal*) any* symmetric *equilibrium that pays within γ_1 of truth-telling* [3] *must be $\tau_1(\gamma_1)$ close to a permutation strategy profile; and moreover*
4. *(*Tight*) no detail free mechanism can have truth-telling as an equilibrium that has strictly higher agent-welfare than all other permutation equilibria.*

where $\tau_1(\gamma_1) = O(\sqrt[3]{\gamma_1})$, (the constants we omit only depend on the first two moments of prior Q)[4]*.*

We extend our results to asymmetric equilibria when the number of agents is sufficiently large in the full version.

3.3 Proof Highlights

In this section we give a few proof highlights.

First note that to show each agent's expected payment in a symmetric equilibrium is less than that of truth-telling, we only need to show the sum of all agents' expected payments – agent welfare – is less than that of truth-telling since everyone plays the same strategy in a symmetric equilibrium.

We first show that the agent welfare of our *Disagreement Mechanism* is the *Classification Score*, which follows by a straightforward computation. It remains to show that *Classification Score* has the aforementioned properties.

Best Prediction Strategy Profiles: We call a strategy profile a *best prediction strategy profile* if for any i, agent i reports a prediction that maximizes his prediction score. By some calculations, we know agent i's *best prediction* is $\theta_{-i}\mathbf{q}_{\sigma_i}$ given σ_i is his private signal and recall that $\theta_{-i} = \frac{\sum_{j\neq i}\theta_i}{n-1}$ where $(\theta_1, \theta_2, ..., \theta_n)$ is the signal strategy. We call this strategy profile a *symmetric* best prediction strategy profile if there exists a signal strategy θ such that $\theta_i = \theta$ for any i. Based on the definition of permutation strategy profile, it is clear that any permutation strategy profile is a symmetric *best prediction* strategy profile.

Consider two agents who report different signals. If they use a permutation strategy profile π whose signal strategy is θ_π, then their predictions will be $\theta_\pi\mathbf{q}_\sigma, \theta_\pi\mathbf{q}_{\sigma'}$ given their private signals are $\sigma \neq \sigma'$. If they use a symmetric best prediction strategy, then their reported predictions will be $\theta\mathbf{q}_\sigma, \theta\mathbf{q}_{\sigma'}$. In the first case, the Hellinger divergence between the two agents' reported predictions is $D^*(\theta_\pi\mathbf{q}_\sigma, \theta_\pi\mathbf{q}_{\sigma'}) = D^*(\mathbf{q}_\sigma, \mathbf{q}_{\sigma'})$ while in the second case, the Hellinger divergence between the two agents' reported predictions is $D^*(\theta\mathbf{q}_\sigma, \theta\mathbf{q}_{\sigma'}) \leq D^*(\mathbf{q}_\sigma, \mathbf{q}_{\sigma'}) = D^*(\theta_\pi\mathbf{q}_\sigma, \theta_\pi\mathbf{q}_{\sigma'})$. The inequality follows from the information monotonicity of Hellinger divergence. Thus, the two agents' predictions in the second case is "closer" than those in the first case. So a permutation strategy profile is more diverse than any other symmetric best prediction strategy, and additionally has no inconsistency. To make permutation strategy profiles beat symmetric best prediction strategy profiles, it is enough to just pay agents the additional diversity reward.

[3] Note that it is an additive gap.
[4] Actually $\tau_1(\gamma_1) = \frac{1}{c_1}\sqrt[3]{\frac{\gamma_1}{c_2c_3c_4}}$

General Equilibria: However, **the biggest challenge** is that there exists equilibria that are not best prediction strategy profiles. Thus, *it is not enough to just pay agents an additional diversity reward.* To deal with this challenge, we replace diversity by *classification score.* To show that classification score works, we map each equilibrium $\boldsymbol{s}^*$ to a strategy profile $\boldsymbol{s}^*_{BP}$ that belongs to *best prediction strategy profiles.* The *technical heart* of the proof bounds the classification score of an equilibrium strategy profile $\boldsymbol{s}^*$ by the diversity of its corresponding best prediction strategy profile $\boldsymbol{s}^*_{BP}$. Once we finish this, we can bound the classification score of any equilibrium strategy profile by the classification score of permutation strategy profiles (note that for permutation strategy profiles, the classification score is equal to the diversity since they are consistent strategy profiles) and complete the proof.

Arriving at this bound requires a non-trivial understanding of the structure of the equilibria, and especially the relation between the different agents' prediction reports in any equilibria. Given a strategy profile for the reports, we obtain a system of linear equations relating the prediction reports. Achieving this bound also requires the delicate use of the triangle inequality applied to $\sqrt{D^*}$. That is why we pick the Hellinger divergence rather than any other f-divergence.

Basically, considering that agents lie for the signal reports, we will show that it's better for them to report their best predictions. Moreover, the information monotonicity shows that even when the agents report their best predictions, it is still worse than truth-telling. Thus, our mechanism is focal.

Asymmetric Equilibria: In the more complicated asymmetric case, the difficulty is that even if agents play best prediction strategy profiles, we cannot use information monotonicity to prove permutation strategy profiles gain the strictly highest classification score. However, if the number of agents is large enough, we will see any strategy profile that belongs to *best prediction strategy profiles* family is "almost symmetric". Using "almost symmetric" result, we can generalize the above framework to approximate work for asymmetric case.

Finally, we show that equilibrium that having the classification score close to that of truth-telling, must be close to a permutation equilibrium.

Tightness Result: The intuitive explanation for this tightness result is that the agents can collude to relabel the signals and the mechanism has no way to defend against this relabelling without knowing some information about agents' common prior. The key idea to prove that result is what we refer to as **Indistinguishable Scenarios**, that is, for the scenario A where agents collude to relabel the signals, there always exists another scenario B where agents tell the truth such that no detail free and truthful mechanism can distinguish A and B.

3.4 Proof Outline for Main Theorem

In this section, we give the proof outline for our main theorem. Recall that we informally defined *Inconsistency* as the average disagreement between every two agents' predictions when they report the same signal and *Diversity* as the average disagreement between every two agents' predictions when they report different signals. We also defined *Classification Score* as Diversity minus Inconsistency. Here we give technical definitions of those concepts.

We first introduce a short hand which will simplify the formula for *Diversity* and *Inconsistency.*

$$\int_{\hat{j},\hat{k}} Pr(\hat{j},\hat{k}) \triangleq \int_{\hat{\sigma}_j,\hat{\mathbf{p}}_j,\hat{\sigma}_k,\hat{\mathbf{p}}_k} Pr_{(\hat{\sigma}_j,\hat{\mathbf{p}}_j)\leftarrow s_j(\sigma_j)}(\hat{\sigma}_j,\hat{\mathbf{p}}_j) Pr_{(\hat{\sigma}_k,\hat{\mathbf{p}}_k)\leftarrow s_k(\sigma_k)}(\hat{\sigma}_k,\hat{\mathbf{p}}_k)$$

where s_j is the strategy of agent j and $s_j(\sigma_j)$ is a distribution over agent j's report profile $(\hat{\sigma}_j, \hat{\mathbf{p}}_j)$ given agent j receives private signal σ_j and uses strategy s_j, and similarly for agent k. This defines the natural measure on the reports of agents j and k given that they play strategies s_j and s_k and a fixed prior Q (which is implicit), and allows us to succinctly describe probabilities of events in this space.

We define *Diversity* as the expected Hellinger divergence D^* between two random agents when they report different signals, so

$$\textit{Diversity} = \sum_{\substack{j \\ k \neq j}} \sum_{\sigma_j, \sigma_k} Pr(j,k) Pr(\sigma_j, \sigma_k) \int_{\hat{j},\hat{k}} Pr(\hat{j},\hat{k}) \delta(\hat{\sigma}_j \neq \hat{\sigma}_k) D^*(\hat{\mathbf{p}}_j, \hat{\mathbf{p}}_k)$$

where $Pr(j,k)$ is the probability agents j,k are picked, and $Pr(\sigma_j, \sigma_k)$ is the probability that agent j receives private signal σ_j and agent k receives private signal σ_k.

Similarly, we can write down the technical definition for *Inconsistency*. But here we do not use Hellinger divergence as the "difference" function in $\sum_{u,v \in U, \mathcal{C}_r(u)=\mathcal{C}_r(v)} D(u,v)$, we use square root of the Hellinger divergence which is the Hellinger distance as the "difference" function. The reason is we want to use the convexity of the Hellinger divergence and the triangle inequality of the Hellinger distance. We will describe the details in the future. For now we give a technical definition for *Inconsistency*:

$$\textit{Inconsistency} = -\sum_{\substack{j \\ k \neq j}} \sum_{\sigma_j, \sigma_k} Pr(j,k) Pr(\sigma_j, \sigma_k) \int_{\hat{j},\hat{k}} Pr(\hat{j},\hat{k}) \delta(\hat{\sigma}_j = \hat{\sigma}_k) \sqrt{D^*(\hat{\mathbf{p}}_j, \hat{\mathbf{p}}_k)}$$

In addition to the *Diversity* and *Inconsistency*, we also introduce a new concept – *TotalDivergence* – and then we use this value as a bridge. Recall that we defined $\textit{Diversity} = \sum_{j,k \neq j, \sigma_j, \sigma_k} Pr(j,k) Pr(\sigma_j, \sigma_k) \int_{\hat{j},\hat{k}} Pr(\hat{j},\hat{k}) \delta(\hat{\sigma}_j \neq \hat{\sigma}_k) D^*(\hat{\mathbf{p}}_j, \hat{\mathbf{p}}_k)$, now we define a similar concept

$$\textit{TotalDivergence} = \sum_{j,k,\sigma_j,\sigma_k} Pr(j,k) Pr(\sigma_j, \sigma_k) \int_{\hat{j},\hat{k}} Pr(\hat{j},\hat{k}) D^*(\hat{\mathbf{p}}_j, \hat{\mathbf{p}}_k)$$

First note that total divergence is robust to summing over j,k or $j \neq k$ since when $j = k$, $D^*(\hat{\mathbf{p}}_j, \hat{\mathbf{p}}_k) = 0$.

We can see $\textit{TotalDivergence} \geq \textit{Diversity}$ since *TotalDivergence* also includes the divergence between the agents who report the same signals. We show that the equality holds if and only if $\textit{Inconsistency} = 0$:

▶ **Claim 4.** *For any strategy profile* $\mathbf{s}$, $\textit{Diversity}(\mathbf{s}) = \textit{TotalDivergence}(\mathbf{s}) \Leftrightarrow \textit{Inconsistency}(\mathbf{s}) = 0$

The above claim implies the below claim. We defer the proofs to the full version.

▶ **Claim 5.**

$$\begin{aligned} \textit{ClassificationScore(truthtelling)} &= \textit{Diversity(truthtelling)} \\ &= \textit{TotalDivergence(truthtelling)} \end{aligned}$$

Now we begin to state our proof outline: For any equilibrium $\boldsymbol{s}$, we define a modified strategy for $\boldsymbol{s}$:

We define $\boldsymbol{s}_{BP}$ what we call a *best prediction strategy* of $\boldsymbol{s}$ as a strategy where each agent uses the same signal strategy which he uses in $\boldsymbol{s}$ but plays his *best prediction* which maximizes

the prediction score. In this case, by some calculations (see full version for a detailed proof), for any i, agent i plays $\theta_{-i}\mathbf{q}_{\sigma_i}$. In the symmetric case, agent i play $\theta\mathbf{q}_{\sigma_i}$.

The results or our main theorem follows from two technical lemmas:

(1) ***ClassificationScore***$(s) \leq$ ***TotalDivergence***(s_{BP})**.** [Main Lemma: Lemma 7] This is our main lemma and the main technical ingredient we use to show our main lemma is the triangle inequality of the square root of Hellinger divergence. Once we show it, we can directly prove the focal property of the Disagreement Mechanism: considering that agents lie for the signal reports, our main technical lemma shows that it's better for them to report their best predictions. Moreover, the information monotonicity shows that even when the agents report their best predictions, it is still worse than truth-telling.

Note that this result is valid for any equilibrium $\boldsymbol{s}$ – symmetric or asymmetric – and is still a main ingredient when we extend the focal property to the asymmetric case.

(2) ***TotalDivergence***$(truthtelling) \approx$ ***TotalDivergence***$(s_{BP}) \Rightarrow \theta \approx \pi$ when $\boldsymbol{s}$ is a symmetric equilibrium with signal strategy θ. This, informally, means that if a symmetric equilibrium pays close to truth-telling, it must be close to a permutation equilibrium, and thus pays about the same as truth-telling.

3.5 Proof for Main Theorem

In this section, we are going to show the first two parts of Theorem 3. We defer the proof for other parts and the asymmetric extension to the full version.

Proof of Theorem 3 Part 1: $\mathcal{M}^+(\alpha, \beta, PS(\cdot,\cdot))$ is strictly truthful.

▶ **Lemma 6.** *The* Disagreement Mechanism *has the same equilibria as the Divergence-based BTS.*

We defer the proof of this lemma to the full version. Radanovic and Faltings [22] have already show $\mathcal{M}(\alpha, \beta, PS(\cdot,\cdot))$ has truth-telling as a strict equilibrium for any SNIFE prior in Theorem 2. Since $\mathcal{M}^+(\alpha, \beta, PS(\cdot,\cdot))$ does not change the equilibrium structure of $\mathcal{M}(\alpha, \beta, PS(\cdot,\cdot))$ according to Claim 6, we have $\mathcal{M}^+(\alpha, \beta, PS(\cdot,\cdot))$ has truth-telling as a strict equilibrium for any SNIFE prior as well. ◀

Proof of Theorem 3 Part 2: $\mathcal{M}^+(\alpha, \beta, PS(\cdot,\cdot))$ has truth-telling as a focal equilibrium. We use our main lemma *ClassificationScore*$(\boldsymbol{s}) <$ *TotalDivergence*$(\boldsymbol{s}_{BP})$ directly to prove: **any *symmetric* non-permutation equilibrium's agent welfare (*ClassificationScore*) must be strictly less than truth-telling**

▶ **Lemma 7** (Main Lemma)**.** *For any equilibrium $\boldsymbol{s}$, if $\boldsymbol{s}_{BP}$ is a best prediction strategy of $\boldsymbol{s}$, we have*

$$\textit{ClassificationScore}(\boldsymbol{s}) \leq \textit{TotalDivergence}(\boldsymbol{s}_{BP})$$

If the equality holds, then we have Inconsistency$(\boldsymbol{s}) = 0$ and $s = \boldsymbol{s}_{BP}$.

We defer the proof of our main lemma to the full version. Notice that if all agents play a symmetric signal strategy θ, then for any j, k, $\theta_{-j} = \theta_{-k} = \theta$. For any symmetric non-permutation equilibrium $\boldsymbol{s}$, it is possible that the signal strategy of $\boldsymbol{s}$ is not a permutation or it is a permutation θ_π but agents do not report $\theta_\pi\mathbf{q}_\sigma$ given σ is their private signal. So we consider two cases:

(a) We first consider the case that the signal strategy θ of $\boldsymbol{s}$ is a permutation matrix θ_π, but agents do not report $\theta_\pi \mathbf{q}_\sigma$. That is, agents collude to relabel the signal but do not report the best predictions based on the collusion.

$$\begin{aligned}
\textit{ClassificationScore}(\boldsymbol{s}) <& \textit{TotalDivergence}(\boldsymbol{s}_{BP}) \\
=& \sum_{j,k,\sigma_j,\sigma_k} Pr(j,k)Pr(\sigma_j,\sigma_k)D^*(\theta_{-j}\mathbf{q}_{\sigma_j}, \theta_{-k}\mathbf{q}_{\sigma_k}) \\
& \qquad \text{(Definition of best prediction strategy)} \\
=& \sum_{j,k,\sigma_j,\sigma_k} Pr(j,k)Pr(\sigma_j,\sigma_k)D^*(\theta_\pi\mathbf{q}_{\sigma_j}, \theta_\pi\mathbf{q}_{\sigma_k}) \\
& \text{(In Case (a), the signal strategy } \theta \text{ of } \boldsymbol{s} \text{ is a permutation matrix } \theta_\pi) \\
=& \sum_{j,k,\sigma_j,\sigma_k} Pr(j,k)Pr(\sigma_j,\sigma_k)D^*(\mathbf{q}_{\sigma_j}, \mathbf{q}_{\sigma_k}) \\
=& \textit{TotalDivergence}(\textit{truthtelling}) \\
=& \textit{ClassificationScore}(\textit{truthtelling})
\end{aligned}$$

The first inequality follows from our main lemma. The inequality is strict for the following reason: when the signal strategy θ of $\boldsymbol{s}$ is a permutation matrix, $\boldsymbol{s}_{BP}$ is a permutation strategy profile. Based on our main lemma if $\textit{ClassificationScore}(\boldsymbol{s}) = \textit{TotalDivergence}(\boldsymbol{s}_{BP})$, we have $\boldsymbol{s} = \boldsymbol{s}_{BP}$ which implies that $\boldsymbol{s}$ is a permutation strategy profile which is a contradiction to the fact $\boldsymbol{s}$ is a non-permutation strategy profile.
The last equality follows from Corollary 5.

(b) We consider the case that the signal strategy θ of $\boldsymbol{s}$ is not a permutation matrix. In this case, agents may or may not report their best predictions. Our main technical lemma shows that it's better for them to report their best predictions. The information monotonicity shows that even when the agents report the best predictions, it is still worse than truth-telling. We still use case (a)'s proof. Even though the inequality in the first line may not be strict, the equality in the fourth line must be a strict inequality:

$$\begin{aligned}
& \sum_{j,k,\sigma_j,\sigma_k} Pr(j,k)Pr(\sigma_j,\sigma_k)D^*(\theta\mathbf{q}_{\sigma_j}, \theta\mathbf{q}_{\sigma_k}) \\
<& \sum_{j,k,\sigma_j,\sigma_k} Pr(j,k)Pr(\sigma_j,\sigma_k)D^*(\mathbf{q}_{\sigma_j}, \mathbf{q}_{\sigma_k}) \qquad \text{(Information Monotonicity)}
\end{aligned}$$

since based on Corollary 16, when θ is not a permutation, and Q is fine-grained and non-zero, the information monotonicity is strict. So in both of the above two cases, we have

$$\textit{ClassificationScore}(\boldsymbol{s}) < \textit{ClassificationScore}(\textit{truthtelling})$$

if $\boldsymbol{s}$ is not a permutation equilibrium. Therefore, our mechanism is focal. ◀

Acknowledgements. We thank David Parkes for helpful conversations and suggestions.

References

1 Arpit Agarwal, Debmalya Mandal, David C. Parkes, and Nisarg Shah. Peer prediction with heterogeneous users. In Constantinos Daskalakis, Moshe Babaioff, and Hervé Moulin,

editors, *Proceedings of the 2017 ACM Conference on Economics and Computation, EC '17, Cambridge, MA, USA, June 26-30, 2017*, pages 81–98. ACM, 2017. doi:10.1145/3033274.3085127.

2 Syed Mumtaz Ali and Samuel D Silvey. A general class of coefficients of divergence of one distribution from another. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 131–142, 1966.

3 S-I Amari and A Cichocki. Information geometry of divergence functions. *Bulletin of the Polish Academy of Sciences: Technical Sciences*, 58(1):183–195, 2010.

4 Yang Cai, Constantinos Daskalakis, and Christos H Papadimitriou. Optimum statistical estimation with strategic data sources. *arXiv preprint arXiv:1408.2539*, 2014.

5 Imre Csiszár, Paul C Shields, et al. Information theory and statistics: A tutorial. *Foundations and Trends® in Communications and Information Theory*, 1(4):417–528, 2004.

6 Anirban Dasgupta and Arpita Ghosh. Crowdsourced judgement elicitation with endogenous proficiency. In *Proceedings of the 22nd international conference on World Wide Web*, pages 319–330. International World Wide Web Conferences Steering Committee, 2013.

7 Boi Faltings, Radu Jurca, Pearl Pu, and Bao Duy Tran. Incentives to counter bias in human computation. In *Second AAAI Conference on Human Computation and Crowdsourcing*, 2014.

8 Rafael M Frongillo and Jens Witkowski. A geometric method to construct minimal peer prediction mechanisms. In *AAAI*, pages 502–508, 2016.

9 Xi Alice Gao, Andrew Mao, Yiling Chen, and Ryan Prescott Adams. Trick or treat: putting peer prediction to the test. In *Proceedings of the fifteenth ACM conference on Economics and computation*, pages 507–524. ACM, 2014.

10 Tilmann Gneiting and Adrian E Raftery. Strictly proper scoring rules, prediction, and estimation. *Journal of the American Statistical Association*, 102(477):359–378, 2007.

11 Radu Jurca and Boi Faltings. Collusion-resistant, incentive-compatible feedback payments. In *Proceedings of the 8th ACM conference on Electronic commerce*, pages 200–209. ACM, 2007.

12 Radu Jurca and Boi Faltings. Mechanisms for making crowds truthful. *J. Artif. Int. Res.*, 34(1), 2009.

13 Vijay Kamble, Nihar Shah, David Marn, Abhay Parekh, and Kannan Ramachandran. Truth serums for massively crowdsourced evaluation tasks. *arXiv preprint arXiv:1507.07045*, 2015.

14 Y. Kong and G. Schoenebeck. A Framework For Designing Information Elicitation Mechanisms That Reward Truth-telling. *ArXiv e-prints*, 2016. arXiv:1605.01021.

15 Y. Kong and G. Schoenebeck. Equilibrium Selection in Information Elicitation without Verification via Information Monotonicity. *ArXiv e-prints*, mar 2016. arXiv:1603.07751.

16 Y. Kong, G. Schoenebeck, and K. Ligett. Putting Peer Prediction Under the Micro(economic)scope and Making Truth-telling Focal. *ArXiv e-prints*, mar 2016. arXiv:1603.07319.

17 Yang Liu and Yiling Chen. Machine-learning aided peer prediction. In Constantinos Daskalakis, Moshe Babaioff, and Hervé Moulin, editors, *Proceedings of the 2017 ACM Conference on Economics and Computation, EC '17, Cambridge, MA, USA, June 26-30, 2017*, pages 63–80. ACM, 2017. doi:10.1145/3033274.3085126.

18 Debmalya Mandal, Matthew Leifer, David C Parkes, Galen Pickard, and Victor Shnayder. Peer prediction with heterogeneous tasks. *arXiv preprint arXiv:1612.00928*, 2016.

19 N. Miller, P. Resnick, and R. Zeckhauser. Eliciting informative feedback: The peer-prediction method. *Management Science*, pages 1359–1373, 2005.

20 D. Prelec. A Bayesian Truth Serum for subjective data. *Science*, 306(5695):462–466, 2004.

21 Goran Radanovic and Boi Faltings. A robust bayesian truth serum for non-binary signals. In Marie desJardins and Michael L. Littman, editors, *Proceedings of the Twenty-Seventh AAAI Conference on Artificial Intelligence, July 14-18, 2013, Bellevue, Washington, USA*. AAAI Press, 2013. URL: `http://www.aaai.org/ocs/index.php/AAAI/AAAI13/paper/view/6451`.

22 Goran Radanovic and Boi Faltings. Incentives for truthful information elicitation of continuous signals. In *Twenty-Eighth AAAI Conference on Artificial Intelligence*, 2014.

23 Goran Radanovic and Boi Faltings. Incentive schemes for participatory sensing. In *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems*, pages 1081–1089. International Foundation for Autonomous Agents and Multiagent Systems, 2015.

24 Blake Riley. Minimum truth serums with optional predictions. In *Proceedings of the 4th Workshop on Social Computing and User Generated Content (SC14)*, 2014.

25 V. Shnayder, A. Agarwal, R. Frongillo, and D. C. Parkes. Informed Truthfulness in Multi-Task Peer Prediction. *ArXiv e-prints*, mar 2016. `arXiv:1603.03151`.

26 Jens Witkowski, Bernhard Nebel, and David C Parkes. *Robust Peer Prediction Mechanisms*. PhD thesis, Ph. D. Dissertation, Albert-Ludwigs-Universitat Freiburg: Institut fur Informatik, 2014.

27 Jens Witkowski and David C. Parkes. A Robust Bayesian Truth Serum for Small Populations. In *Proceedings of the 26th AAAI Conference on Artificial Intelligence (AAAI 2012)*, 2011.

28 Jens Witkowski and David C Parkes. Peer prediction without a common prior. In *Proceedings of the 13th ACM Conference on Electronic Commerce*, pages 964–981. ACM, 2012.

29 Jens Witkowski and David C Parkes. Learning the prior in minimal peer prediction. In *Proceedings of the 3rd Workshop on Social Computing and User Generated Content at the ACM Conference on Electronic Commerce*, page 14. Citeseer, 2013.

30 Peter Zhang and Yiling Chen. Elicitability and knowledge-free elicitation with peer prediction. In *Proceedings of the 2014 international conference on Autonomous agents and multi-agent systems*, pages 245–252. International Foundation for Autonomous Agents and Multiagent Systems, 2014.

A Notation Table

B Formal Definitions of the SNIFE Priors

▶ **Assumption 8** (Symmetric Prior). *We assume throughout that the agents' signals* $\boldsymbol{\sigma}$ *are drawn from some joint* **symmetric prior** Q.

Because we will assume that the prior is symmetric, we denote $q_i(\sigma)$ by $q(\sigma)$ and $q_{i,j}(\sigma|\sigma')$ (where $i \neq j$) by $q(\sigma|\sigma')$. We also define $\mathbf{q}_\sigma = q(\cdot|\sigma)$.

▶ **Assumption 9** (Non-zero Prior). *We assume that for any* $\sigma, \sigma' \in \Sigma$, $q(\sigma) > 0, q(\sigma|\sigma') > 0$.

▶ **Assumption 10** (Informative Prior). *We assume if agents have different private signals, they will have different expectations for the fraction of at least one signal. That is for any* $\sigma \neq \sigma'$, *there exists* σ'' *such that* $q(\sigma''|\sigma) \neq q(\sigma''|\sigma')$.

The following assumption conceptually states that one state is not just a more likely version of another state, and can be thought of as a weaker version of assuming $q(\sigma|\cdot)$ are linearly independent.

Table 2 Basic Notations.

Σ	$\triangleq$	the space of all possible signals
Δ_Σ	$\triangleq$	the space of all probability distributions over Σ
σ_i	$\triangleq$	agent i's private signal
$\hat{\sigma}_i$	$\triangleq$	agent i's reported signal
$\mathbf{p}_i$ or $\mathbf{q}_{\sigma_i}$	$\triangleq$	agent i's prediction for other agents' private signals conditioning on she receives σ_i
$\hat{\mathbf{p}}_i$	$\triangleq$	agent i's reported prediction
$r_i = (\hat{\sigma}_i, \hat{\mathbf{p}}_i)$	$\triangleq$	agent i's report
$\mathbf{r}$	$\triangleq$	all agents' reports
s_i	$\triangleq$	agent i's strategy
θ_i	$\triangleq$	the signal strategy (marginal distribution) of s_i
θ_{-i}	$\triangleq$	$\frac{\sum_{j\neq i}\theta_j}{n-1}$ the average signal strategy excluding θ_i
$\bar{\theta}_n$	$\triangleq$	$\frac{\sum_i \theta_i}{n}$ the average signal strategy
$\mathbf{s}$	$\triangleq$	the strategy profile agents have
Q	$\triangleq$	the common symmetric prior agents share
$q(\sigma'\|\sigma)$	$\triangleq$	an agent's expected fraction of other agents who receives σ' when this agent receives σ
$q(\sigma)$	$\triangleq$	the priori expected fraction of agents who will receive σ

▶ **Assumption 11** (Fine-grained Prior). *We assume that for any $\sigma \neq \sigma' \in \Sigma$, there exists σ'', σ''' such that*

$$\frac{q(\sigma|\sigma'')}{q(\sigma'|\sigma'')} \neq \frac{q(\sigma|\sigma''')}{q(\sigma'|\sigma''')}$$

If this assumption does not hold, then in some sense σ and σ' are the same signal since the agents who receive σ have the same posterior with the agents who receive σ'. Therefore, We can replace σ and σ' with a new signal $\sigma_0 := \sigma$ or σ', and not lose any information.

We illustrate this in the below example:

▶ **Example 12.** $Q = \begin{pmatrix} q(s_1|s_1) & q(s_1|s_2) & q(s_1|s_3) \\ q(s_2|s_1) & q(s_2|s_2) & q(s_2|s_3) \\ q(s_3|s_1) & q(s_3|s_2) & q(s_3|s_3) \end{pmatrix} = \begin{pmatrix} 0.1 & 0.2 & 0.3 \\ 0.2 & 0.4 & 0.6 \\ 0.7 & 0.4 & 0.1 \end{pmatrix}$ is not a fine-grained prior since

$$\frac{q(s_1|s_1)}{q(s_2|s_1)} = \frac{q(s_1|s_2)}{q(s_2|s_2)} = \frac{q(s_1|s_3)}{q(s_2|s_3)}$$

Note that in this example, even we combine s_1 and s_2 to be a single signal s_0 which is defined as $s_0 := s_1$ or s_2, we do not lose any information: if an agent knows that the fraction of agents who report s_0 is x, we know his belief for the expectation of the fraction of s_1 must be $\frac{x}{3}$ no matter what private signal he receives.

We only require the fine-grained prior assumption to show that truth-telling is *strictly* "better" than any other symmetric equilibrium (excluding permutation equilibrium). In the above example where the prior is not fine-grained, if agents always report s_1 when they receive s_1 or s_2, this does not lose information (is not "worse") comparing with the case agents always tell the truth. So we cannot say truth-telling is strictly "better" than any other

equilibrium when the prior is not fine-grained. However, this assumption is not necessary to show that truth-telling is a strict Bayesian equilibrium of our mechanism, nor to show that the agent welfare of truth-telling is at least as high as other symmetric equilibrium.

▶ **Assumption 13** (Ensemble Prior). *Although we talk of a single prior, in fact we have an ensemble $Q = \{Q_n\}_{n\in N, n\geq 3}$ of priors; one for each possible number of agents greater than 3. We assume that all Q_n are over the same signal set Σ have identical $q(\sigma)$ and $q(\sigma'|\sigma)$.*

When the number of agents n changes, the joint prior actually changes as well, but the first two moments of the prior are fixed. This allows us to make meaningful statements about n going to infinity.

C Information Monotonicity

▶ **Lemma 14** (Information Monotonicity ([3])). *For any strictly convex function f, f-divergence $D_f(\mathbf{p}, \mathbf{q})$ satisfies information monotonicity so that for any transition matrix $\theta \in \mathbb{R}^{\Sigma\times\Sigma}$, $D_f(\mathbf{p}, \mathbf{q}) \geq D_f(\theta\mathbf{p}, \theta\mathbf{q})$.*

Moreover, the inequality is strict if and only if there exists $\sigma, \sigma', \sigma''$ such that $\theta(\sigma, \sigma')\mathbf{p}(\sigma') > 0$, $\theta(\sigma, \sigma'')\mathbf{p}(\sigma'') > 0$ and $\frac{\mathbf{p}(\sigma'')}{\mathbf{p}(\sigma')} \neq \frac{\mathbf{q}(\sigma'')}{\mathbf{q}(\sigma')}$.

We introduce the proof and give an example where the strictness condition is not satisfied here.

Proof. The proof follows from algebraic manipulation and one application of convexity.

$$D_f(\theta\mathbf{p}, \theta\mathbf{q}) = \sum_\sigma (\theta\mathbf{p})(\sigma) f\left(\frac{(\theta\mathbf{q})(\sigma)}{(\theta\mathbf{p})(\sigma)}\right) \tag{1}$$

$$= \sum_\sigma \theta(\sigma, \cdot)\mathbf{p} f\left(\frac{\theta(\sigma, \cdot)\mathbf{q}}{\theta(\sigma, \cdot)\mathbf{p}}\right) \tag{2}$$

$$= \sum_\sigma \theta(\sigma, \cdot)\mathbf{p} f\left(\frac{1}{\theta(\sigma, \cdot)\mathbf{p}} \sum_{\sigma'} \theta(\sigma, \sigma')\mathbf{p}(\sigma')\frac{\mathbf{q}(\sigma')}{\mathbf{p}(\sigma')}\right) \tag{3}$$

$$\leq \sum_\sigma \theta(\sigma, \cdot)\mathbf{p} \frac{1}{\theta(\sigma, \cdot)\mathbf{p}} \sum_{\sigma'} \theta(\sigma, \sigma')\mathbf{p}(\sigma') f\left(\frac{\mathbf{q}(\sigma')}{\mathbf{p}(\sigma')}\right) \tag{4}$$

$$= \sum_\sigma \mathbf{p}(\sigma) f\left(\frac{\mathbf{q}(\sigma)}{\mathbf{p}(\sigma)}\right) = D_f(\mathbf{p}, \mathbf{q}) \tag{5}$$

The second equality holds since $(\theta\mathbf{p})(\sigma)$ is dot product of the σ^{th} row of θ and $\mathbf{p}$.

The third equality holds since $\sum_{\sigma'} \theta(\sigma, \sigma')\mathbf{p}(\sigma')\frac{\mathbf{q}(\sigma')}{\mathbf{p}(\sigma')} = \theta(\sigma, \cdot)\mathbf{q}$.

The fourth inequality follows from the convexity of $f(\cdot)$.

The last equality holds since $\sum_\sigma \theta(\sigma, \sigma') = 1$.

We now examine under what conditions the inequality in Equation 4 is strict. Note that for any strictly convex function g, if $\forall u, \lambda_u > 0$, $g(\sum_u \lambda_u x_u) = \sum_u \lambda_u g(x_u)$ if and only if there exists x such that $\forall u, x_u = x$. By this property, the inequality is strict if and only if there exists $\sigma, \sigma', \sigma''$ such that $\frac{\mathbf{p}(\sigma'')}{\mathbf{p}(\sigma')} \neq \frac{\mathbf{q}(\sigma'')}{\mathbf{q}(\sigma')}$ and $\theta(\sigma, \sigma')\mathbf{p}(\sigma') > 0$, $\theta(\sigma, \sigma'')\mathbf{p}(\sigma'') > 0$. ◀

To understand the strictness condition more in Lemma 14, we give an example where the strictness condition is not satisfied:

▶ **Example 15.** $\mathbf{p} = (0.1\ 0.2\ 0.7)$, $\mathbf{q} = (0.2\ 0.4\ 0.4)$, $\theta = \begin{pmatrix} 0.3 & 0.6 & 0 \\ 0.7 & 0.4 & 0 \\ 0 & 0 & 1 \end{pmatrix}$.

We show by case analysis that we cannot find $\sigma, \sigma', \sigma''$ such that $\theta(\sigma, \sigma')\mathbf{p}(\sigma') > 0$, $\theta(\sigma, \sigma'')\mathbf{p}(\sigma'') > 0$ and $\frac{\mathbf{p}(\sigma'')}{\mathbf{p}(\sigma')} \neq \frac{\mathbf{q}(\sigma'')}{\mathbf{q}(\sigma')}$.

First note that because $\frac{\mathbf{p}(\sigma'')}{\mathbf{p}(\sigma')} \neq \frac{\mathbf{q}(\sigma'')}{\mathbf{q}(\sigma')}$, it cannot be that $\sigma' = \sigma''$, nor can it be the case that $\sigma', \sigma'' \in \{1, 2\}$ because $\frac{\mathbf{p}(1)}{\mathbf{p}(2)} = \frac{\mathbf{q}(1)}{\mathbf{q}(2)}$ and $\frac{\mathbf{p}(2)}{\mathbf{p}(1)} = \frac{\mathbf{q}(2)}{\mathbf{q}(1)}$. Thus it must be that either $\sigma' \in \{1, 2\}$ and $\sigma'' = 3$ or $\sigma' = 3$ and $\sigma'' \in \{1, 2\}$. Because these are symmetric, we consider the first case.

Because $\theta(\sigma, \sigma')\mathbf{p}(\sigma') > 0$ it must be that $\sigma \in \{1, 2\}$, but because $\theta(\sigma, \sigma'')\mathbf{p}(\sigma'') > 0$, it must be that $\sigma = 3$. So no assignment of $\sigma, \sigma', \sigma''$ is possible.

Thus, the strictness condition is not satisfied. By simple calculations, we have $\theta\mathbf{p} = (0.15\ 0.15\ 0.7)$, $\theta\mathbf{q} = (0.3\ 0.3\ 0.4)$. By some algebraic calculations, we have $D_f(\mathbf{p}, \mathbf{q}) = D_f(\theta\mathbf{p}, \theta\mathbf{q})$ for any function f.

▶ **Corollary 16.** *Given SNIFE prior Q, for any θ that is not a permutation, there exists two private signals $\sigma_1 \neq \sigma_2$ such that $D_f(\theta\mathbf{q}_{\sigma_1}, \theta\mathbf{q}_{\sigma_2}) < D_f(\mathbf{q}_{\sigma_1}, \mathbf{q}_{\sigma_2})$*

Proof. First notice that when θ is not a permutation, there exists a row of θ such that the row has at least two positive entries, in other words, there exists $\sigma, \sigma', \sigma''$ such that $\theta(\sigma, \sigma'), \theta(\sigma, \sigma'') > 0$. Based on the non-zero and fine-grained assumptions of Q, there exists $\sigma_1 \neq \sigma_2$ such that
$\theta(\sigma, \sigma')\mathbf{p}(\sigma'), \theta(\sigma, \sigma'')\mathbf{p}(\sigma'') > 0$ and $\frac{\mathbf{p}(\sigma')}{\mathbf{p}(\sigma'')} \neq \frac{\mathbf{q}(\sigma')}{\mathbf{q}(\sigma'')}$ where $\mathbf{p} = \mathbf{q}_{\sigma_1}, \mathbf{q} = \mathbf{q}_{\sigma_2}$. When $\theta(\sigma, \sigma')\mathbf{p}(\sigma'), \theta(\sigma, \sigma'')\mathbf{p}(\sigma'') > 0$, we have $\theta(\sigma, \cdot)\mathbf{p} > 0$. By Lemma 14, we have

$$D_f(\theta\mathbf{q}_{\sigma_1}, \theta\mathbf{q}_{\sigma_2}) < D_f(\mathbf{q}_{\sigma_1}, \mathbf{q}_{\sigma_2}).$$ ◀

Optimizing Bayesian Information Revelation Strategy in Prediction Markets: the Alice Bob Alice Case

Yuqing Kong[*1] **and Grant Schoenebeck**[†2]

1 **University of Michigan, Ann Arbor, USA**
yuqkong@umich.edu
2 **University of Michigan, Ann Arbor, USA**
schoeneb@umich.edu

Abstract

Prediction markets provide a unique and compelling way to sell and aggregate information, yet a good understanding of optimal strategies for agents participating in such markets remains elusive. To model this complex setting, prior work proposes a three stages game called the Alice Bob Alice (A-B-A) game – Alice participates in the market first, then Bob joins, and then Alice has a chance to participate again. While prior work has made progress in classifying the optimal strategy for certain interesting edge cases, it remained an open question to calculate Alice's best strategy in the A-B-A game for a general information structure.

In this paper, we analyze the A-B-A game for a general information structure and (1) show a "revelation-principle" style result: it is enough for Alice to use her private signal space as her announced signal space, that is, Alice cannot gain more by revealing her information more "finely"; (2) provide a FPTAS to compute the optimal information revelation strategy with additive error when Alice's information is a signal from a constant-sized set; (3) show that sometimes it is better for Alice to reveal partial information in the first stage even if Alice's information is a single binary bit.

1998 ACM Subject Classification J.4 Social and Behavioral Sciences

Keywords and phrases prediction market, information revelation, optimization

Digital Object Identifier 10.4230/LIPIcs.ITCS.2018.14

1 Introduction

Prediction markets aggregate information from diverse resources in a compelling manner. However, despite their powerful function in real-life deployments, large holes remain in the theory of prediction markets. For example, a basic information revelation question – how should an agent reveal her information in a prediction market to maximize her expected payoff – is still not fully answered. To model the complex setting of prediction market and deal with the information revelation question, Chen et al. [4, 3], Chen and Waggoner [5] propose and study a three stages game called the Alice Bob Alice (A-B-A) game – Alice participates in the market first, then Bob joins, and then Alice has an opportunity to participate again. They also define two special information structures – "substitutes" and "compliments" – and

* The author was supported by National Science Foundation Career Award 1452915 and CCF Award 1618187.
† The author was supported by National Science Foundation Career Award 1452915 and Algorithms in the Field Award 1535912.

show that when traders' information are substitutes (compliments), Alice should reveal her information as soon (late) as possible. However, apart from those extreme cases, it remained an open question to calculate Alice's optimal information revelation strategy in the A-B-A game for a general information structure which is the main focus of the current paper.

Computing optimal information revelation strategy is also a key problem in the Bayesian persuasion literature ([11, 7, 2]). Bayesian persuasion, proposed by Kamenica and Gentzkow [11], models a situation where a informed sender (partially) reveals her information to persuade an uninformed receiver to adopt an action. In the Bayesian persuasion model, the informed sender first commits to an information revelation strategy and then announces a signal based upon the committed strategy. Borrowing this idea of commitment from Bayesian persuasion, we consider the A-B-A game where Alice makes a commitment to her information revelation strategy before the game. We call the strategy in the A-B-A game with commitment a *Bayesian* information revelation strategy. Like in the Bayesian persuasion case, this power of commitment makes sense if we expect the game to be repeated [11]. People cannot lie to others in the market and make money forever. After many rounds of market activities, people will identify rules to effectively translate the leaked information.

Our Results

The current paper analyzes A-B-A for the general information structures and

1) proves a general revelation principle style theorem (Section 5.2.2) for the A-B-A game by showing it is enough for Alice to use her private signal space as her announced signal space, that is, Alice cannot gain more by revealing her information more "finely".
2) leverages the intuition of the aforementioned result to give a fully polynomial time approximation scheme (FPTAS) to compute the optimal information revelation strategy with additive error when Alice's information is a signal from a constant-sized set (Section 5.2.2);
3) shows that sometimes it is better for Alice to reveal partial information in the first stage, even when Alice's information is a binary bit (Section 4, Appendix B).

Before our result, it was not known how to compute (regardless of time complexity) an optimal strategy for Alice even when her signal was binary, or that such a strategy existed for a general information structure.

1.1 Related Work

Bayesian Persuasion (BP) Model

Conceptually, the Bayesian Persuasion model is different than the A-B-A with commitment. In the A-B-A game, both Alice and Bob sell information to the market and Bob is informed as well and is actually a competitor of Alice. In contrast, in BP, the sender sells the information to the uninformed receiver and has different kind of utility function with the receiver. Technically, in BP, the goal function is a linear function of the revelation strategy while the A-B-A is much more complicated.

Despite those differences, Kamenica and Gentzkow [11] also show a "revelation principle" style statement which is similar with the statement in the current paper – it is enough for the sender to draw her announced signal from the receiver's action space. That is, the sender cannot gain more by making her announced signal space more complicated. More formally, if the receiver's action space is A_r and the sender's private information space is X_s, the sender can obtain the optimal utility by just optimizing over the space of all "simple" strategies

$M : X_s \times A_r \mapsto [0, 1]$ such that $M(x, a)$ represents the probability the sender who has private information x and announces action a regardless of other "complicated" strategies. However, the proof of our A-B-A revelation principle is much more complicated than that in the BP case. In the A-B-A game, Bob's possible actions (best responses) are infinite, while in the BP case they are finite. Moreover, Alice's utility depends non-linearly on Bob's action which means even though Bob's action space has a good structure, any simple proof is unlikely to work. To prove the A-B-A revelation principle, we use a totally different method involving linear programming.

Information Revelation Problem

As mentioned in the introduction, Chen et al. [4, 3], Chen and Waggoner [5] propose and study the A-B-A game. When Alice's information and Bob's information are independent with each other, their information is defined as "compliments". On the other hand, when Alice's information and Bob's information are conditionally independent with each other (conditioning on the event they want to forecast), their information is defined as "substitutes". Chen et al. [4, 3], Chen and Waggoner [5] show that Alice should reveal her information as late (early) as possible when their information is "compliments" ("substitutes"). In those extreme cases, Alice cannot obtain better utility by partially revealing her information which is not true in the general case which is studied in the current paper.

Azar et al. [1] consider a model where the market price is a reverse Gaussian random walk and the expert who has a less noisy signal should decide a time to announce her signal. However, in their model, the expert only has a chance to participate in the market once while in the A-B-A game, Alice has multiple chances to participate the market and can partially reveal her information at first to obtain better utility. Moreover, the information structure considered in Azar et al. [1] is limited by their assumptions while the current paper considers the general information structure.

2 Preliminaries

2.1 Prediction Markets

In this section, we introduce the market scoring rule (MSR) model [9, 10]. We first introduce the main technical tools in the MSR model – proper scoring rules [13], which are used to measure the *score* (accuracy) of the forecast.

Proper Scoring Rules [8, 13]

A scoring rule which we denote $PS : \Sigma \times \Delta_\Sigma \mapsto \mathbb{R}$ takes in a signal $\sigma \in \Sigma$ and a distribution over signals $\mathbf{p} \in \Delta_\Sigma$ and outputs a real number. A scoring rule is *proper* if, whenever the first input is drawn from a distribution $\mathbf{p}$, the expectation of PS is maximized if the second input is $\mathbf{p}$. That is, $\mathbf{p} \in \arg\max_{\mathbf{p}'} \mathbb{E}_{\sigma\sim\mathbf{p}}[PS(\sigma, \mathbf{p}')]$. A scoring rule is called *strictly proper* if $\mathbf{p}$ uniquely maximizes $\mathbb{E}_{\sigma\sim\mathbf{p}}[PS(\sigma, \mathbf{p}')]$. We will assume throughout that the scoring rules we use are strictly proper. By slightly abusing notation, we can extend a scoring rule to be $PS : \Delta_\Sigma \times \Delta_\Sigma \mapsto \mathbb{R}$ by simply taking $PS(\mathbf{p}, \mathbf{p}') = \mathbb{E}_{\sigma\leftarrow\mathbf{p}}(\sigma, \mathbf{p}')$. Any proper scoring rule is linear in the first term.

Fix an outcome space Σ for a signal σ. Let $\mathbf{q} \in \Delta_\Sigma$ be a reported distribution.

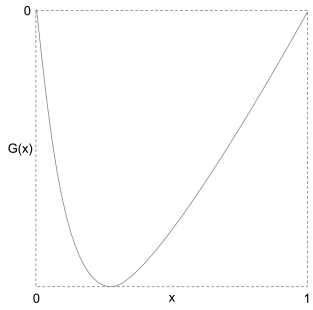


Figure 1 An example of nice convex function $G : [0, 1] \mapsto \mathbb{R}$, $G(0) = G(1) = 0$.

▶ **Example 1.** (Logarithmic Scoring Rule) A *logarithmic scoring rule* maps a signal σ and reported distribution $\mathbf{q}$ to a payoff as follows:

$$LSR(\sigma, \mathbf{q}) = \log(\mathbf{q}(\sigma)).$$

Let the signal σ be drawn from some random process with distribution $\mathbf{p} \in \Delta_\Sigma$.
Then the expected payoff of the logarithmic scoring rule is

$$\mathbb{E}_{\sigma \leftarrow \mathbf{p}}[LSR(\sigma, \mathbf{q})] = LSR(\mathbf{p}, \mathbf{q}) = \sum_\sigma \mathbf{p}(\sigma) \log \mathbf{q}(\sigma)$$

This value will be maximized if and only if $\mathbf{q} = \mathbf{p}$

▶ **Example 2.** (Quadratic Scoring Rule / Brier scoring rule) A *quadratic scoring rule* which maps a signal σ and reported distribution $\mathbf{q}$ to a payoff as follows:

$$QSR(\sigma, \mathbf{q}) = 2\mathbf{q}(\sigma) - \sum_{\sigma'} \mathbf{q}(\sigma')^2 - 1.$$

Let the signal σ be drawn from some random process with distribution $\mathbf{p} \in \Delta_\Sigma$.
Then the expected payoff of the logarithmic scoring rule is

$$\mathbb{E}_{\sigma \leftarrow \mathbf{p}}[QSR(\sigma, \mathbf{q})] = QSR(\mathbf{p}, \mathbf{q}) = 2\langle \mathbf{p}, \mathbf{q} \rangle - \langle \mathbf{q}, \mathbf{q} \rangle - 1$$

This value will be maximized if and only if $\mathbf{q} = \mathbf{p}$.

In general, proper scoring rules can be constructed from convex functions. Given a bounded convex function $H : \Delta_\Sigma \mapsto \mathbb{R}$, we define $PS_H : \Sigma \times \Delta_\Sigma \mapsto \mathbb{R}$ such that

$$PS_H(\sigma, \mathbf{p}) = H(\mathbf{p}) - \langle H'(\mathbf{p}), \mathbf{p} \rangle + H'_\sigma(\mathbf{p})$$

where $\langle , \rangle$ denotes the inner product of two vectors and H'_σ denotes the partial derivative of H with respect to the σ^{th} entry.

▶ **Fact 3.** *[8] When $H : \Delta_\Sigma \mapsto \mathbb{R}$ is (strictly) convex, $PS_H : \Sigma \times \Delta_\Sigma \mapsto \mathbb{R}$ is (strictly) proper and $\forall \mathbf{p}$, $PS_H(\mathbf{p}, \mathbf{p}) = H(\mathbf{p})$.*

To control the convergence rate analysis in the future, we consider a special class of proper scoring rules.

▶ **Definition 4** (Nice convex functions)**.** We say a convex real function $G : [0, 1] \mapsto \mathbb{R}$ is *nice* if (i) $G(0) = G(1) = 0$ and (ii) there exists a constant $\lambda > 0$ such that when ϵ is sufficiently small, $\max\{|G(\epsilon)|, |G(1 - \epsilon)|\} \leq \epsilon^\lambda$. We denote the set of all such nice functions as $\mathcal{G}$.

▶ **Definition 5** (PS^G, H_G, $\mathcal{PSG}$)**.** Given a bounded strictly convex real function $G : [0, 1] \mapsto \mathbb{R}$, (Figure 1), we define $H_G : \Delta_\Sigma \mapsto \mathbb{R}$ as a function such that $H_G(\mathbf{p}) := \sum_\sigma G(\mathbf{p}(\sigma))$ for any $\mathbf{p} \in \Delta_\Sigma$. We define $PS^G(\sigma, \mathbf{p}) := PS_{H_G}(\sigma, \mathbf{p})$. We call such a proper scoring rule *good* if G is a nice convex function, and define the set of all nice proper scoring scoring rules as $\mathcal{PSG}$.

Now we explain the restrictions of nice convex functions $\mathcal{G}$. If we pick $G(x) = x \log x$ which is a nice convex function (Example 1), the proper scoring rule is the common used log scoring rule and $|PS^G(\mathbf{p}, \mathbf{p})| = -H_G(\mathbf{p}) = -\sum_\sigma \mathbf{p}(\sigma) \log \mathbf{p}(\sigma)$ which is the Shannon entropy of distribution $\mathbf{p}$ [6].

Note that entropy can be interpreted as the uncertainty of distribution $\mathbf{p}$. For example, when $\mathbf{p} = (0, 1, 0, 0, 0)$, there is no uncertainty, the entropy is 0. When we use other $G(x)$, we still want $|PS^G(\mathbf{p}, \mathbf{p})| = -H_G(\mathbf{p})$ to be interpreted as the uncertainty of distribution $\mathbf{p}$. Therefore, we put the restriction $G(0) = G(1) = 0$. Then if when there is no uncertainty, $\mathbf{p}$ must be an extreme point of Δ_Σ, $H_G(\mathbf{p}) = G(1) + G(0) + ... + G(0) = 0$.

The second restriction is needed when we analyze the convergence rate in the future. We hope $G(x)$ never changes too fast. Note that it's a weaker condition than lipschiz condition since $G(x) = x \log x$ does not satisfy the lipschiz condition but satisfy our restriction since $|G(\epsilon)| = \epsilon \log \epsilon \leq \epsilon^\lambda, \forall 0 < \lambda < 1$ and $|G(1 - \epsilon)| = (1 - \epsilon) \log(1 - \epsilon) \leq \log(1 - \epsilon) \leq \epsilon$.

Note this special class is still rich and several commonly used proper scoring rules, including the aforemetioned examples, belong to this class.

▶ Remark. Setting $G(x) = x \log x$, the nice proper scoring rule PS^G is the logarithmic scoring rule. Setting $G(x) = (x - \frac{1}{2})^2 - \frac{1}{4}$, the nice proper scoring rule PS^G is the quadratic scoring rule.

▶ Remark. Note that affine transformations preserve convexity. Without loss of generality, we assume for any $PS^G \in \mathcal{PSG}$, $|PS^G(\mathbf{p}, \mathbf{p})| = |H_G(\mathbf{p})| \leq 1, \forall \mathbf{p}$. This also means our results apply to the Brier Scoring rule, which is a shift of the quadradic scoring rule.

Market Scoring Rule Model

The theoretical prediction market model used in the current paper is the market scoring rule (MSR) model which is proposed by Hanson [9, 10]. In this model, market price corresponds to people's beliefs for the event. When a trader changes the market belief from p_1 to p_2, the automated market maker market scoring rule (MSR) will pay the trader the "accuracy" of forecast p_2 minus the "accuracy" of forecast p_1. The "accuracy" of the forecast is measured by the proper scoring rules [13]. We provide a formal definition in the below paragraph.

▶ **Definition 6** (Prediction market $PM(PS, X_E)$ [9, 10])**.** Let X_E be a random variable that people want to forecast. The market maker sets up an initial belief for X_E. Every agent can modify the market belief. When an agent changes the market belief from p_1 to p_2, her payment will be the score of belief p_2 minus the score of belief p_1, that is, $PS(X_E, p_2) - PS(X_E, p_1)$ after X_E is revealed.

2.2 Notation

For two random variables X, Y which are drawn from space $[n] \times [m]$, we define

$$\Pr[\mathbf{Y}|X = i] := (\Pr[Y = 1|X = i], \Pr[Y = 2|X = i], \cdots, \Pr[Y = m|X = i]).$$

For any function $f : [n] \mapsto \mathbb{R}$, $\mathbb{E}_X f(X) = \sum_i \Pr[X = i] f(i)$.

For a matrix M, we use $M_{i,\cdot}$ to denote the i^{th} row, $M_{\cdot,j}$ to denote the j^{th} column, and $M_{i,j} = M(i,j)$ to denote the entry at the i^{th} row and the j^{th} column. In particular, $M_{i,\cdot}$ is a row vector and $M_{\cdot,j}$ is a column vector.

A matrix M is a *transition matrix* if every entry of M is non-negative and the sum of all entries in each *row* is 1. Throughout this paper, we denote by $\mathcal{M}_{n\times m}$ the set of all transition matrices that have dimension $n \times m$, and denote $\mathcal{M}_{n\times *} = \bigcup_{m\geq n} \mathcal{M}_{n\times m}$.

Given a random variable X that has n possible outcomes, an transition matrix $M \in \mathcal{M}_{n\times m}$ defines a **transition probability** that transforms X to $M(X)$ such that $M(X)$ is a new random variable that has m possible outcomes where $\Pr[M(X) = j|X = i] = M_{i,j}$.

If the distribution of X is represented by an $1 \times n$ row vector $\mathbf{p}$, then the distribution over $M(X)$ is $\mathbf{p}M$ and $\Pr[M(x) = j] = \mathbf{p} \cdot M_{\cdot,j}$.

3 Alice Bob Alice Game with Commitment

We analyse the Alice Bob Alice game with commitment in this section, that is, Alice commits a signaling scheme before the game. The random event of interest is X_E. X_E is drawn from a signal space $\Sigma_E, |\Sigma_E| = n_E$. We use a proper scoring rule based prediction market $PM(PS, X_E)$ to pay Alice and Bob. Suppose Alice's private information is X_A and Bob's private information is X_B. X_A is drawn from a signal space $\Sigma_A, |\Sigma_A| = n_A$ and X_B is drawn from a signal space $\Sigma_B, |\Sigma_B| = n_B$.

We assume both Alice and Bob are rational.

▶ **Definition 7** (signaling scheme)**.** Given that a signal space $\Sigma, |\Sigma| = m$, we define Alice's signaling scheme M as an $n_A \times m$ transition matrix such that $M_{x_A,\sigma} = M(x_A, \sigma)$ is the probability Alice announces signal $\sigma \in \Sigma$ given private information $X_A = x$.

A signaling scheme M defines a transition probability. We define X_σ as a random variable such that $X_\sigma := M(X_A)$, that is,

$$X_A \xrightarrow{M} X_\sigma.$$

Alice Bob Alice Game with Commitment (X_A, X_B, X_E, PS)

Stage 0 Alice commits her signaling scheme M.

Stage 1 Alice receives a signal $\sigma_A \in \Sigma_A$, implements her signaling scheme, and announces the result $\sigma \in \Sigma$. Alice changes the market belief for event X_E from the original prior forecast $\mathbf{p}_0 = \Pr[\mathbf{X_E}]$ to

$$\mathbf{p}_1 = \Pr[\mathbf{X_E}|X_\sigma = \sigma, M].$$

Stage 2 Bob changes the market belief to $\mathbf{p}_2$ (which is a function of $M, \mathbf{p}_1, X_B$ and Bob's strategy).

Stage 3 Alice changes the market belief to $\mathbf{p}_3$ (which is a function of $M, \mathbf{p}_1, \mathbf{p}_2, X_A$ and Alice's strategy).

Payment Both Alice and Bob are paid according to proper scoring rule based prediction market $PM(PS, X_E)$. Suppose the initial market belief is $\mathbf{p}_0 = \Pr[X_E|X_\sigma]$. Alice and Bob's payments are

$$\begin{aligned}\mu_A =& (PS(X_E, \mathbf{p}_1) - PS(X_E, \mathbf{p}_0)) + (PS(X_E, \mathbf{p}_3) - PS(X_E, \mathbf{p}_2))\\ \mu_B =& PS(X_E, \mathbf{p}_2) - PS(X_E, \mathbf{p}_1)\end{aligned}$$

correspondingly.

Fix the joint distribution over random variables X_A, X_B, X_E. Consider an A-B-A game with commitment $(X_A, X_B, X_E, PS \in \mathcal{PSG})$. We assume Alice and Bob will optimally respond in stage 2 and stage 3. Actually we will see since the market uses strictly proper scoring rule, both Alice and Bob's optimal responses in stage 2 and stage 3 are unique (Claim 11, 12). In this case, both Alice and Bob's expected payments can be seen as a function of Alice's signaling scheme M. We define Alice's expected payment as $\mu_A^*(M)$; and Bob's expected payment as $\mu_B^*(M)$.

We define $\mu^*(M) := \mu_A^*(M) + \mu_B^*(M)$. We also define $\mu^* := \sup_M \mu^*(M)$, $\mu_B^\dagger := \inf_M \mu_B^*(M)$ (note that it's an infimum here), and $\mu_A^* := \mu^* - \mu_B^\dagger$. Note that $\mu_A^*(M) \le \mu_A^*$.

▶ **Definition 8** (Optimizing Signaling Scheme Problem)**.** Consider an A-B-A game with commitment $(X_A, X_B, X_E, PS \in \mathcal{PSG})$. An optimizing signaling scheme problem is the problem of constructing Alice's optimal signaling scheme M^* such that $\mu_A^*(M^*) = \mu_A^*$ if exists or a series of signaling schemes $\{M^*(\epsilon)\}_\epsilon$ such that

$$\mu_A^*(M^*(\epsilon)) \xrightarrow{\epsilon \to 0} \mu_A^*.$$

4 Summary of the Main Results

▶ **Theorem 9** (Optimizing Signaling Scheme)**.** *Given the joint distribution over random variables X_A, X_B, X_E, consider an A-B-A game with commitment $(X_A, X_B, X_E, PS^G \in \mathcal{PSG})$. When Alice's private signal is from a constant-sized set, that is, n_A is a constant integer T, for all sufficiently small $0 < \epsilon < 1$, there exists an $O\left((LP(\frac{1}{\epsilon}+1)^T) + n_B n_E (\frac{1}{\epsilon}+1)^T) + n_B^2 n_E\right)$ time algorithm that constructs the signaling scheme $M^*(\epsilon) \in \mathcal{M}_{n_A \times n_A}$ such that*

$$\mu_A^*(M^*(\epsilon)) \ge \mu_A^* - \Theta(|\epsilon| + n_E|G(\epsilon)| + n_E|G(1-\epsilon)|)$$

where $LP(k)$ is the time complexity of linear programming with k variables.

Moreover, when Alice commits to signaling scheme $M^*(\epsilon)$, the optimal responses of Bob and Alice in stage 2 and stage 3 are

$$\mathbf{p}_2^* = \Pr[\mathbf{X_E}|X_\sigma, X_B] \qquad \mathbf{p}_3^* = \Pr[\mathbf{X_E}|X_A, X_B]$$

respectively where $X_A \xrightarrow{M^*(\epsilon)} X_\sigma$.

▶ **Corollary 10.** *Given the joint distribution over random variables X_A, X_B, X_E, consider an A-B-A game with commitment $(X_A, X_B, X_E, PS^G \in \mathcal{PSG})$. When Alice's private signal is from a constant-sized set, that is, n_A is a constant integer T, there is a FPTAS for Alice to optimize her signaling scheme with additive error.*

We defer the full proof to the end of Section 5.

Proof Sketch of Theorem 9

We will first give a game theoretic analysis for the optimal strategy for Alice and Bob in stage 2 and stage 3. The definition of the proper scoring rules implies that in stage 2 and stage 3, Alice and Bob should honestly report their best forecast for event X_E at that stage. The best forecast should be the posterior probability of X_E conditioning on all possible information they have at that time.

Step 1 Minimizing Bob's optimal expected payment Fixing the joint distribution over Alice and Bob's private information and the event, Bob's optimal expected payment $\mu_B^*(M)$ is a function of Alice's signal scheme $M \in \mathcal{M}^{n \times *}$. To calculate the signal scheme $M^\dagger$ for Alice to minimize Bob's optimal expected payment, we will prove that

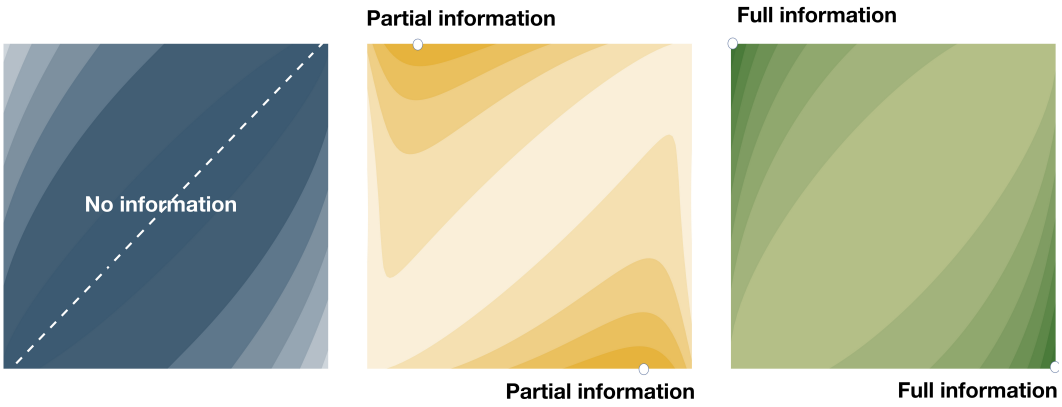


Figure 2 The optimal signal scheme when X_A is a binary random variable given different joint distributions overX_A, X_B, X_E (see Appendix for the numerical values of the three joint distributions). Left: $M^* = \begin{bmatrix} x & 1-x \\ x & 1-x \end{bmatrix}, \forall x$ Alice's optimal strategy is revealing no information. Middle: $M^* = \begin{bmatrix} 0.18 & 0.82 \\ 1 & 0 \end{bmatrix}$ or $\begin{bmatrix} 0.82 & 0.18 \\ 0 & 1 \end{bmatrix}$ Alice's optimal strategy is revealing partial information. Right: $M* = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ or $\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$ Alice's optimal strategy is revealing full information.

Revelation principle it is sufficient to optimize over $M \in \mathcal{M}^{n\times n}$, we prove this by showing a "decomposibility" property of Bob's optimal expected payment $\mu_B^*(M)$;

Continousness when $M \approx M'$, $\mu_B^*(M) \approx \mu_B^*(M')$.

Then we use a linear programming based algorithm to approximate the signal scheme $M^\dagger$ for Alice to minimize Bob's optimal expected payment.

Step 2 Maximizing the total expected payment We will perturb $M^\dagger$ to $M^* \approx M^\dagger$ such that adopting the signaling scheme M^* guarantees that the sum of Alice and Bob's optimal expected payment obtains the upper-bound.

After finishing the above two steps, we obtain M^* which gives Alice the expected payment which is close to optima since M^* both minimizes Bob's optimal expected payment and maximizes the sum of Alice and Bob's expected payment.

$$\begin{aligned}
\mu_A^*(M^*) =& \mu^*(M^*) - \mu_B^*(M^*) \\
=& \mu^* - \mu_B^*(M^*) \\
\approx& \mu^* - \mu_B^*(M^\dagger) && (M^* \approx M^\dagger, \text{ continousness of } \mu_B^*(M)) \\
=& \mu^* - \mu_B^\dagger \\
=& \mu_A^*
\end{aligned}$$

Experimental Results

We show that sometimes it is better for Alice to reveal partial information in the first stage even if Alice's information is a single binary bit by providing the optimal signal scheme of Alice in the A-B-A game in three scenarios (Figure 2). We give the numerical values of the three scenarios in appendix.

5 Optimizing Signaling Scheme

5.1 Game Theoretic Analysis of A-B-A

This section shows that in stage 2 and stage 3, Alice and Bob should honestly report their best forecast for event X_E at that stage. The best forecast is the posterior probability of X_E conditioning on all possible information they have at that time. Moreover, the sum of the expected payments of Alice and Bob obtains its upper-bound if Alice learns the exact value of X_B from Bob's behavior and plays the strategy in stage 3 that fully aggregates the information.

▶ **Claim 11.** *Given the joint distribution over random variables X_A, X_B, X_E, consider an A-B-A game with commitment $(X_A, X_B, X_E, PS^G \in \mathcal{PSG})$. Given that Alice commits signaling scheme M, $X_A \xrightarrow{M} X_\sigma$, Bob's optimal action is changing the market belief to $\mathbf{p}_2^* = \Pr[\mathbf{X_E}|X_\sigma, X_B]$ in stage 2 and his optimal expected payment is*

$$\mu_B^*(M) = \mathbb{E}_{X_\sigma, X_B}[H_G(\Pr[\mathbf{X_E}|X_\sigma, X_B]) - H_G(\Pr[\mathbf{X_E}|X_\sigma])].$$

Recall that $-H_G(\mathbf{p})$ can be interpreted as the uncertainty / "entropy" of distribution $\mathbf{p}$. The uncertainty of event X_E decreases with more information. Therefore, Bob's expected payment can be interpreted as the contribution of Bob's private information X_B to decrease the uncertainty if the event X_E given the existence of the partial information X_σ of Alice.

Proof of Claim 11. Bob only has one chance to participate. Thus, he will always report his truthful forecast which is conditioning on his own information and the information Alice conveys to him. Therefore, $\Pr[\mathbf{X_E}|X_\sigma, X_B]$ is Bob's optimal action and

$$\begin{aligned}
&\mathbb{E}_{X_E, X_\sigma, X_B} PS^G(X_E, \mathbf{p}_2) - PS^G(X_E, \mathbf{p}_1) \\
=&\mathbb{E}_{X_\sigma, X_B}\mathbb{E}_{X_E|X_\sigma, X_B}[PS^G(X_E, \mathbf{p}_2) - PS^G(X_E, \Pr[\mathbf{X_E}|X_\sigma])] && (\mathbf{p}_1 = \Pr[\mathbf{X_E}|X_\sigma]) \\
\leq&\mathbb{E}_{X_E, X_\sigma, X_B}[PS^G(X_E, \Pr[\mathbf{X_E}|X_\sigma, X_B]) - PS^G(X_E, \Pr[\mathbf{X_E}|X_\sigma])] \\
& && \text{(definition of proper scoring rule)} \\
=&\mathbb{E}_{X_E, X_\sigma, X_B} PS^G(X_E, \Pr[\mathbf{X_E}|X_\sigma, X_B]) \\
&- \mathbb{E}_{X_E, X_\sigma} PS^G(X_E, \Pr[\mathbf{X_E}|X_\sigma]) \\
& && \text{(The second part is independent of Bob's information } X_B) \\
=&\mathbb{E}_{X_\sigma, X_B}[H_G(\Pr[\mathbf{X_E}|X_\sigma, X_B]) - H_G(\Pr[\mathbf{X_E}|X_\sigma])] && \text{(Fact 3)}
\end{aligned}$$

◀

▶ **Claim 12.** *Given the joint distribution over random variables X_A, X_B, X_E, consider an A-B-A game with commitment $(X_A, X_B, X_E, PS^G \in \mathcal{PSG})$. When Alice learns the exact value of X_B from Bob's behavior, Alice's optimal action is changing the market belief to $\mathbf{p}_3^* = \Pr[\mathbf{X_E}|X_A, X_B]$ in stage 3 and the optimal sum of expected payment is*

$$\mu^* = \mathbb{E}_{X_E, X_A, X_B}[H_G(\Pr[\mathbf{X_E}|X_A, X_B]) - H_G(\Pr[\mathbf{X_E}])].$$

The optimal total expected payment can be interpreted as the contribution of Alice and Bob's private information to decrease the uncertainty of the event X_E and can be obtained when in stage 3, Alice learns all information.

Proof of Claim 12. According to the definition of the proper scoring rules, the optimal $\mathbf{p}_3$ in stage 3 is the forecast that conditions on all information X_A, X_B. When Alice learns the

exact value of X_B from Bob's behavior, she can play this optimal strategy in stage 3 which makes the sum of expected payments of Alice and Bob optimal.

$$
\begin{aligned}
&\mathbb{E}_{X_E,X_A,X_B} PS^G(X_E,\mathbf{p}_3) - PS^G(X_E,\mathbf{p}_0)\\
=&\mathbb{E}_{X_A,X_B}\mathbb{E}_{X_E|X_A,X_B} PS^G(X_E,\mathbf{p}_3) - \mathbb{E}_{X_E} PS^G(X_E,\mathbf{p}_0)\\
\leq&\mathbb{E}_{X_E,X_A,X_B}[PS^G(X_E,\Pr[\mathbf{X_E}|X_A,X_B]) - PS^G(X_E,\mathbf{p}_0)]\\
&&\text{(definition of proper scoring rule)}\\
=&\mathbb{E}_{X_E,X_A,X_B} PS^G(X_E,\Pr[\mathbf{X_E}|X_A,X_B]) - \mathbb{E}_{X_E} PS^G(X_E,Pr[\mathbf{X_E}]) && (\mathbf{p}_0 = Pr[\mathbf{X_E}])\\
=&\mathbb{E}_{X_A,X_B}[H_G(\Pr[\mathbf{X_E}|X_A,X_B]) - H_G(\Pr[\mathbf{X_E}])] && \text{(Fact 3)}
\end{aligned}
$$

◀

5.2 Minimizing Bob's Expected Payment

5.2.1 Decomposability and Continuousness of Bob's Expected Payment

This section will show two important properties required of $\mu_B^*(M)$ for the optimization step.

▶ **Definition 13** (Decomposability). Recall that $\mathcal{M}_{n\times *}$ is the set of all transition matrices which have n rows. A function $F : \mathcal{M}_{n\times *} \mapsto \mathbb{R}$ is *decomposable* if there exists a function $f : \mathbb{R}^n \mapsto \mathbb{R}$ satisfying $f(\lambda\mathbf{v}) = \lambda f(\mathbf{v})$ for any $\lambda \in \mathbb{R}^+, \mathbf{v} \in \mathbb{R}^n$ such that for any

$$M = [M_{\cdot,1} \quad M_{\cdot,2} \quad \cdots \quad M_{\cdot,m}] \in \mathcal{M},$$

we have

$$F(M) = \sum_{j=1}^{m} f(M_{\cdot,j}).$$

Given the joint distribution over random variables X_A, X_B, X_E, consider an A-B-A game with commitment $(X_A, X_B, X_E, PS^G \in \mathcal{PSR})$. Recall that we define Bob's optimal expected payment as $\mu_B^*(M)$ and random variables X_A, X_B, and X_E have n_A, n_B, and n_E possible outcomes respectively. The signal space of Alice is $\Sigma, |\Sigma| = m$.

▶ **Lemma 14** (Decomposability). *$\mu_B^*(M)$ is a decomposable function of $M \in \mathcal{M}_{n\times *}$.*

▶ **Lemma 15** (Continousness). *For every $M \in \mathcal{M}_{n_A\times m}$, if $\max_{i,j} |M_{i,j} - M'_{i,j}| \leq \epsilon$, then*

$$|\mu_B^*(M') - \mu_B^*(M))| \leq \Theta(n_A m(n_E|G(\epsilon)| + n_E|G(1-\epsilon)| + \epsilon)).$$

We defer the full proofs to the appendix.

Proof sketch

Recall that

$$\mu_B^*(M) = \mathbb{E}_{X_\sigma,X_B}[H_G(\Pr[\mathbf{X_E}|X_\sigma,X_B]) - H_G(\Pr[\mathbf{X_E}|X_\sigma])].$$

For the decomposability,

$$
\begin{aligned}
\mu_B^*(M) =&\mathbb{E}_{X_\sigma,X_B}[H_G(\Pr[\mathbf{X_E}|X_\sigma,X_B]) - H_G(\Pr[\mathbf{X_E}|X_\sigma])] && \text{(Claim 11)}\\
=&\sum_\sigma \Pr[X_\sigma = \sigma]\mathbb{E}_{X_B|X_\sigma=\sigma}[H_G(\Pr[\mathbf{X_E}|X_\sigma = \sigma, X_B]) - H_G(\Pr[\mathbf{X_E}|X_\sigma = \sigma])]
\end{aligned}
$$

We define $\phi_j(M) := \Pr[X_\sigma = j]$, $\psi_j(M) := \mathbb{E}_{X_B|X_\sigma=\sigma} H_G(\Pr[\mathbf{X_E}|X_\sigma = \sigma, X_B]) - H_G(\Pr[\mathbf{X_E}|X_\sigma = \sigma])$.

We will show

(i) $\phi_j(M)$ is a linear function of $M_{\cdot,j}$

(ii) $\psi_j(M)$ only depends on the "shape" of $M_{\cdot,j} - \frac{M_{\cdot,j}}{S(M_{\cdot,j})}$[1] – where $S(M_{\cdot,j})$ is the sum of vector $M_{\cdot,j}$

Combining both (i) and (ii), we can see $\phi_j(M)\psi_j(M) = \Phi(M_{\cdot,j})\Psi(M_{\cdot,j})$ only depends on $M_{\cdot,j}$ and moreover, for any $\lambda \in \mathbb{R}^+$,

$$\Phi(\lambda M_{\cdot,j})\Psi(\frac{\lambda M_{\cdot,j}}{S(\lambda M_{\cdot,j})}) = \lambda\Phi(M_{\cdot,j})\Psi(\frac{M_{\cdot,j}}{S(M_{\cdot,j})})$$

that is, it preserves the scalar multiplication of $M_{\cdot,j}$. Therefore, $\mu_B^*(M) = \sum_j \phi_j(M)\psi_j(M)$ is a decomposable function of M. We defer the proofs of (i) and (ii) to the appendix.

The proof for continousness is a little bit tricky. When we perturb M a little bit, it's possible that $\Pr[\mathbf{X_E}|M(X_A)])$ changes a lot. Consider an extreme case where the prior of X_A is a uniform distribution and we pick transition probability M such that $\Pr[M(X_A) = j] = 0.000001$. We add $\epsilon = 0.01$ to M_{ij} to obtain M'. In this case, $M'_{ij} >> M'_{kj}, k \neq i$. Thus, conditioning on $M'(X_A) = j$ the probability $X_A = i$ is close to 1. Therefore, $\Pr[\mathbf{X_E}|M'(X_A) = j] \approx \Pr[\mathbf{X_E}|X_A = i]$. However, since we can still freely determine the "shape" of $M_{\cdot,j}$, we can make $\Pr[\mathbf{X_E}|M(X_A) = j]$ far away from $\Pr[\mathbf{X_E}|X_A = i] \approx \Pr[\mathbf{X_E}|M'(X_A) = j]$ even if $M \approx M'$. Fortunately, this bad case only happens when $\Pr_M[X_\sigma = j]$ is very small. We will show that the product $\Pr[X_\sigma = j]H_G(\Pr[\mathbf{X_E}|M(X_A) = j])$ is robust with respect to M.

The key property needed in the proof of contiousness is the convexity of function $G(x) : [0,1] \mapsto \mathbb{R}$. For a convex function $G(x)$, its derivative is a monotone function which implies that the absolute value of the derivative is maximized at the endpoints. That is why the values of $|G(\epsilon)|$ and $|G(1-\epsilon)|$ dominate the convergence rate.

5.2.2 Optimizing a Decomposable and Continuous Function

▶ **Definition 16.** $F : \mathcal{M}_{n\times *} \mapsto \mathbb{R}$ is $C(\epsilon, n)$-continuous if for all sufficiently small $0 < \epsilon < 1$, for every $M, M' \in \mathcal{M}_{n\times n}$, if $\max_{i,j} |M_{i,j} - M'_{i,j}| \leq \epsilon$, then

$$|F(M') - F(M))| \leq C(\epsilon, n).$$

▶ **Theorem 17.** *If $F : \mathcal{M}_{n\times *} \mapsto \mathbb{R}$ is decomposable, then*

$$\min_{M\in\mathcal{M}_{n\times *}} F(M) = \min_{M\in\mathcal{M}_{n\times n}} F(M) \qquad \text{and} \qquad \max_{M\in\mathcal{M}_{n\times *}} F(M) = \max_{M\in\mathcal{M}_{n\times n}} F(M).$$

Moreover, when F is $C(\epsilon, n)$-continuous, for all sufficiently small $0 < \epsilon < 1$, there exists an $O\left(LP(\frac{1}{\epsilon}+1)^n\right)$ time algorithm that outputs $M^{-*}(\epsilon), M^{+*}(\epsilon) \in \mathcal{M}_{n\times n}$ such that

$$F(M^{-*}(\epsilon)) \leq \min_{M\in\mathcal{M}_{n\times *}} F(M) + C(\epsilon, n) \qquad \text{and} \qquad F(M^{+*}(\epsilon)) \geq \max_{M\in\mathcal{M}_{n\times *}} F(M) - C(\epsilon, n)$$

where $LP(k)$ is the time complexity of linear programming with k variables.

Proof.

▶ **Claim 18.** *For any $M \in \mathcal{M}_{n\times *}$, there exists $M^-, M^+ \in \mathcal{M}_{n\times n}$ such that*

$$F(M^-) \leq F(M) \leq F(M^+).$$

[1] if $M_{\cdot,j} = (0, 0, ..., 0)^\top$, we define $\frac{M_{\cdot,j}}{S(M_{\cdot,j})}$ as $(0, 0, ..., 0)^\top$.

Part 1: Revelation Principle

The above claim directly implies the first result. It remains to show the claim. Let's construct the below linear program. Fix any $M \in \mathcal{M}_{n\times m}$,

$$\min_{\mathbf{x}} \sum_j x_j f(M_{\cdot,j}) \tag{1}$$

$$s.t. \sum_j x_j M_{\cdot,j} = [1, 1...1]^\top_{1\times n} \tag{2}$$

$$x_j \geq 0, \forall j \tag{3}$$

Note that $\sum_j M_{\cdot,j} = [1, 1...1]^\top_{1\times n}$ since M is a transition matrix. Thus, the above linear program must have a solution which implies that it must have a basic feasible solution (bfs) $\mathbf{x}^*$ [12]. Since $\mathbf{x}^*$ is a bfs, it must have at least $m - n$ zero entries. Therefore, there exists a size n subset $\{j_1, j_2, ..., j_n\}$ such that for any $j \notin \{j_1, j_2, ..., j_n\}$, $x^*_j = 0$.

Let $M^- = [x^*_{j_1} M_{j_1} \quad x^*_{j_2} M_{j_2} \quad ... \quad x_{j_n} M_{j_n}]$.

$\sum_k x^*_{j_k} M_{j_k} = \sum_j x^*_j M_{\cdot,j} = [1, 1...1]^\top_{1\times n}$. Thus, M^- is a transition matrix. Moreover,

$$\begin{aligned} F(M^-) &= F(\sum_k x^*_{j_k} M_{j_k}) \\ &= \sum_k x^*_{j_k} f(M_{j_k}) && \text{(Decomposability of } F) \\ &= \sum_j x^*_j f(M_{\cdot,j}) \leq \sum_j f(M_{\cdot,j}) = F(M) && \text{(For any } j \notin \{j_1, j_2, ..., j_n\},\ x^*_j = 0) \end{aligned}$$

Therefore, we finish our construction of M^-. The construction of M^+ is similar.

Part 2: LP Based Algorithm

The LP based algorithm for $\arg\min_{M\in\mathcal{M}_{n\times *}} F(M)$ is in Algorithm 1. Solving $\arg\max_{M\in\mathcal{M}_{n\times *}} F(M)$ is similar.

We can first enumerate all possible row vectors of $M \in \mathcal{M}_{n\times *}$ which are $(\frac{1}{\epsilon} + 1)^n$ in number. Then we can construct linear programming (1) with the all possible row vectors and solve the linear programming to obtain a basic feasible solution. ◀

5.3 Maximizing the Total Expected Payment

In this section, we will show that for any signaling scheme M, we can always perturb M a little bit to M' such that M' maximizes the optimal sum of expected payments of Alice and Bob.

The A-B-A game is a constant-sum game if Bob is required to reveal (announce) his full information to Alice, and Alice plays rationally in the final round. This is because the market belief will be changed from $\Pr[X_E]$ to $\Pr[X_E|X_A, X_B]$ regardless of Alice's signaling scheme. In this case, information is fully aggregated. Therefore when Bob is required to announce his full information, minimizing Bob's expected payment is equivalent to maximizing Alice's expected payment since ABA is a constant sum game.

However, when Bob is not required to announce his full information, minimizing Bob's expected payment is not equivalent to maximizing Alice's expected payment. For example, we consider the case where both X_A and X_B are i.i.d. binary bits (equal 1 with probability (w.p.) $\frac{1}{2}$, equal 0 w.p. $\frac{1}{2}$), and $X_E = X_A \oplus X_B$. To minimize Bob's expected payment, Alice

Algorithm 1 Minimizing Decomposable Function (MDF): $\arg\min_{M \in \mathcal{M}_{n\times *}} F(M)$

```
function MDF(f(·), n, ε)
N = 1/ε
enumerate all [ℓ_1/N, ℓ_2/N, ..., ℓ_n/N]
where ℓ_1, ℓ_2, ..., ℓ_n ∈ {0, 1, 2, ..., N} and denote them by {M_{·,j}}_{j=1}^{(N+1)^n}

solve the following linear program and return a BFS x*

        min_x Σ_i x_j f(M_{·,j})
s.t.    Σ_i x_j M_{·,j} = [1, 1...1]^⊤_{1×n}
        x_j ≥ 0, ∀j

pick a size n subset {j_1, j_2, ..., j_n} s.t. for any j ∉ {j_1, j_2, ..., j_n}, x*_j = 0

return [x*_{j_1} M_{·,j_1}   x*_{j_2} M_{·,j_2}   ...   x*_{j_n} M_{·,j_n}]
```

should reveal no information in the first stage. However, in this case, rational Bob will not change the market belief and thus leaks no information of Bob's signal. In this case, the total payment for Alice and Bob is 0. On the other hand, if Alice commits a signaling scheme where with some small probability ϵ she announces X_A, and with probability $1-\epsilon$ she flips a coin and announces the result. This signaling scheme entices Bob to move the market thus Alice can identify Bob's full information to guarantee almost optimal expected payment of Alice. Actually via a similar idea, for any signaling scheme M, we will construct signaling schemes $\{M(\epsilon)\}_\epsilon \approx M$ such that for all sufficiently small $\epsilon > 0$,

$$\mu^*(M(\epsilon)) = \mu^*.$$

Recall that n_A is the size of Alice's private information space, n_B is the size of Bob's private information space, n_E is the size the event X_E's outcome space and m is the size of Alice's announced signal space.

▶ **Lemma 19** (Perturbing Signaling Scheme). *Given the joint distribution over random variables X_A, X_B, X_E, consider an A-B-A game with commitment $(X_A, X_B, X_E, PS^G) \in \mathcal{PSR}$. For any signaling scheme M, for all sufficiently small $0 < \epsilon < 1$, we can always use $O(mn_An_B^2n_E)$ time [2] to perturb M to $M(\epsilon)$ such that*

$$\mu^*(M(\epsilon)) = \mu^* \qquad \text{and} \qquad \max_{x_A,\sigma} |M(x_A,\sigma) - M(\epsilon)(x_A,\sigma)| \le \Theta(mn_B^2\epsilon).$$

We defer the construction of the perturbing signaling scheme and the proof of Lemma 19 to Appendix A.

Now we are ready to give a full proof for the main theorem – Theorem 9 and Corollary 10.

Proof of Theorem 9. We construct the optimal signaling scheme of Alice using two steps.

Step 1: Minimizing Bob's expected payment we first use $O\left(\frac{1}{\epsilon}\right)^T n_B n_E)$ time to construct the linear programming in Algorithm 1. We then spend $O\left(L(\frac{1}{\epsilon})^T)\right)$ time to solve the linear programming to obtain signaling scheme $M^\dagger(\epsilon) \in \mathcal{M}^{T\times T}$. Note that $\mu_B^*(M^\dagger(\epsilon)) \ge \mu_B^\dagger - \Theta(\epsilon + n_E|G(\epsilon)| + n_E|G(1-\epsilon)|)$ based on Lemma 15 and the fact that we enumerate an ϵ-net of all possible column vectors in Algorithm 1.

[2] Note that the running time is independent with ϵ

Step 2: Maximizing the total expected payment: Lemma 19 shows for *any* signaling scheme M, there exists an $O(n_B^2 n_E)$ algorithm that constructs $M(\epsilon')$ such that

$$\max_{x_A,\sigma} |M(x_A,\sigma) - M(\epsilon')(x_A,\sigma)| \leq \Theta(n_B^2\epsilon')$$

and $\mu^*(M(\epsilon')) = \mu^*$. Since the running time of the perturbing method is independent with ϵ', we can pick sufficiently small ϵ' such that $n_B^2\epsilon' \leq \epsilon$. Thus, we can still use $O(n_B^2 n_E)$ time to perturb $M^\dagger(\epsilon)$ to $M^*(\epsilon)$ such that

$$\max_{x_A,\sigma} |M^\dagger(\epsilon)(x_A,\sigma) - M^*(\epsilon)(x_A,\sigma)| \leq \epsilon$$

and $\mu^*(M^*(\epsilon)) = \mu^*$.

We can see

$$\begin{aligned}
\mu_A^*(M^*(\epsilon)) =& \mu^*(M^*(\epsilon)) - \mu_B^*(M^*(\epsilon)) \\
=& \mu^* - \mu_B^*(M^*(\epsilon)) \\
\geq& \mu^* - \mu_B^*(M^\dagger(\epsilon)) - \Theta(n_A^2(\epsilon + n_E|G(\epsilon)| + n_E|G(1-\epsilon)|)) && \text{(Lemma 15)} \\
\geq& \mu^* - \mu_B^\dagger - \Theta(n_A^2(\epsilon + n_E|G(\epsilon)| + n_E|G(1-\epsilon)|)) && \text{(step 1)} \\
=& \mu^* - \mu_B^\dagger - \Theta(\epsilon + n_E|G(\epsilon)| + n_E|G(1-\epsilon)|) \\
& && (n_A = T \text{ is a constant integer}) \\
=& \mu_A^* - \Theta(\epsilon + n_E|G(\epsilon)| + n_E|G(1-\epsilon)|)
\end{aligned}$$

When Alice commits to signaling scheme $M^*(\epsilon)$, according to Claim 11 and Claim 12,

$$\mathbf{p}_2^* = \Pr[\mathbf{X_E}|X_\sigma, X_B] \qquad \mathbf{p}_3^* = \Pr[\mathbf{X_E}|X_A, X_B]$$

where $X_A \xrightarrow{M^*(\epsilon)} X_\sigma$. ◀

Proof of Corollary 10. $\tau = \Theta(|\epsilon| + n_E|G(\epsilon)| + n_E|G(1-\epsilon)|) \leq \Theta(\epsilon^{\min\{\lambda,1\}} n_E)$ implies that there exists a positive constant C such that when τ is sufficiently small, $\frac{1}{\epsilon} \leq C(\frac{n_E}{\tau})^{\frac{1}{\min\{1,\lambda\}}}$. Theorem 9 implies that we have a $poly(\frac{1}{\tau}, n_B, n_E)$ complexity algorithm to have $\pm\tau$ approximation. ◀

6 Discussion

Our FPTAS result depends on the assumption that Alice's signal space is constant size. In fact, the complexity of our algorithm depends exponentially on the size of Alice's signal space. Note that before our results even the case where Alice's signal is binary was an open question. In practice, to apply our results one might reduce the signal space size by merging similar signals to hopefully obtain a good approximation.

A potential future direction is proving a hardness result for optimal information revelation problem in the ABA case when Alice's signal is arbitrarily large, or finding a better dependence on the size of Alice's signal space.

Acknowledgements. We thank Biaoshuai Tao for helpful suggestions.

References

1 Yossi Azar, Amir Ban, and Yishay Mansour. When should an expert make a prediction? In *Proceedings of the 2016 ACM Conference on Economics and Computation*, pages 125–142. ACM, 2016.

2 Yakov Babichenko and Siddharth Barman. Computational aspects of private bayesian persuasion. *arXiv preprint arXiv:1603.01444*, 2016.

3 Yiling Chen, Stanko Dimitrov, Rahul Sami, Daniel M Reeves, David M Pennock, Robin D Hanson, Lance Fortnow, and Rica Gonen. Gaming prediction markets: Equilibrium strategies with a market maker. *Algorithmica*, 58(4):930–969, 2010.

4 Yiling Chen, Daniel M Reeves, David M Pennock, Robin D Hanson, Lance Fortnow, and Rica Gonen. Bluffing and strategic reticence in prediction markets. In *International Workshop on Web and Internet Economics*, pages 70–81. Springer, 2007.

5 Yiling Chen and Bo Waggoner. Informational substitutes. In *Foundations of Computer Science (FOCS), 2016 IEEE 57th Annual Symposium on*, pages 239–247. IEEE, 2016.

6 Thomas M. Cover and Joy A. Thomas. *Elements of information theory (2. ed.)*. Wiley, 2006.

7 Shaddin Dughmi and Haifeng Xu. Algorithmic bayesian persuasion. *arXiv preprint arXiv:1503.05988*, 2015.

8 Tilmann Gneiting and Adrian E Raftery. Strictly proper scoring rules, prediction, and estimation. *Journal of the American Statistical Association*, 102(477):359–378, 2007.

9 Robin Hanson. Combinatorial information market design. *Information Systems Frontiers*, 5(1):107–119, 2003.

10 Robin Hanson. Logarithmic markets coring rules for modular combinatorial information aggregation. *The Journal of Prediction Markets*, 1(1):3–15, 2012.

11 Emir Kamenica and Matthew Gentzkow. Bayesian persuasion. *The American Economic Review*, 101(6):2590–2615, 2011.

12 David G Luenberger. *Introduction to linear and nonlinear programming*, volume 28. Addison-Wesley Reading, MA, 1973.

13 Robert L Winkler. Scoring rules and the evaluation of probability assessors. *Journal of the American Statistical Association*, 64(327):1073–1078, 1969.

A Proofs of Lemma 14, 15, 19

Proof of Lemma 14. We start to prove (i) and (ii).

Proof of (i)

Recall that if the distribution of X is represented by a row vector $\mathbf{p}$, then the distribution over $M(X)$ is $\mathbf{p}M$ and $\Pr[M(x) = j] = \mathbf{p} \cdot M_{\cdot,j}$.

Thus $\Pr[X_\sigma = j] = \Pr[M(X_A) = j] = \Pr[\mathbf{X_A}] \cdot M_{\cdot,j}$ is a linear function of the vector $M_{\cdot,j}$.

Proof of (ii)

For $\psi_j(M)$, since

$$\begin{aligned}\Pr[X_B = x_B, X_E = x_E | X_\sigma = j] &= \frac{\Pr[X_B = x_B, X_E = x_E, X_\sigma = j]}{\Pr[X_\sigma = j]} \\ &= \frac{\Pr[X_B = x_B, X_E = x_E, \mathbf{X_A}] \cdot M_{\cdot,j}}{\Pr[\mathbf{X_A}] \cdot M_{\cdot,j}} \\ &= \frac{\Pr[X_B = x_B, X_E = x_E, \mathbf{X_A}] \cdot \frac{M_{\cdot,j}}{S(M_{\cdot,j})}}{\Pr[\mathbf{X_A}] \cdot \frac{M_{\cdot,j}}{S(M_{\cdot,j})}}\end{aligned}$$

only depends on $\frac{M_{\cdot,j}}{S(M_{\cdot,j})}$. Since $\psi_j(M)$ only depends on the joint distribution over (X_B, X_E) conditioning on $X_\sigma = j$, thus, $\psi_j(M)$ only depends on $\frac{M_{\cdot,j}}{S(M_{\cdot,j})}$ as well. ◀

Proof of Lemma 15. It is sufficient to show that if $M' = M$ except $M'_{ij} = M_{ij} + \epsilon$, $M'_{ik} = M_{ik} - \epsilon$,

$$|\mu_B^*(M') - \mu_B^*(M)| \leq \Theta(n_E|G(\epsilon) + n_E|G(1-\epsilon)| + \epsilon).$$

Note that when $G(x)$ is convex, for sufficiently small constant ϵ, $n_E|G(\epsilon)| + n_E|G(1-\epsilon)| + \epsilon$ is an increasing function of ϵ. Therefore, if $M' = M$ except $M'_{ij} = M_{ij} + \epsilon'$, $M'_{ik} = M_{ik} - \epsilon'$ where $\epsilon' \leq \epsilon$, we still have

$$|\mu_B^*(M') - \mu_B^*(M)| \leq \Theta(n_E|G(\epsilon) + n_E|G(1-\epsilon)| + \epsilon).$$

The proof of Lemma 14 shows that $\mu_B^*(M)$ can be decomposed as m parts and the only part that relates to $M_{\cdot,j}$ is

$$\Pr[M(X_A) = j]\mathbb{E}_{X_B|M(X_A)=j}[H_G(\Pr[\mathbf{X_E}|M(X_A) = j, X_B]) - H_G(\Pr[\mathbf{X_E}|M(X_A) = j])]$$

We define $\gamma(M_{\cdot,j}) := \Pr[M(X_A) = j]H_G(\Pr[\mathbf{X_E}|M(X_A) = j])$ and would like to the below claim.

▶ **Claim 20.** *When $M'_{\cdot,j} = M_{\cdot,j}$ except $M'_{ij} = M_{ij} + \epsilon$,*

$$|\gamma(M_{\cdot,j}) - \gamma(M'_{\cdot,j})| \leq \Theta(n_E|G(\epsilon)| + n_E|G(1-\epsilon)| + \epsilon).$$

The above claim is valid for every possible joint distribution over X_E, X_A. The set of all possible joint distributions over X_E, X_A equals the set of all possible joint distributions over X_E, X_A conditioning X_B for any X_B. Therefore, the above claim also implies that part $\Pr[M(X_A) = j]\mathbb{E}_{X_B|M(X_A)=j}H_G(\Pr[\mathbf{X_E}|M(X_A) = j, X_B])$ also only fluctuates at most $\Theta(n_E|G(\epsilon)| + n_E|G(1-\epsilon)| + \epsilon)$ when we perturb M_{ij} at most ϵ. We have similar analysis for $M_{\cdot,k}$, therefore, Lemma 15 follows.

It remains to show the claim.

Proof of Claim 20. Without loss of generality we assume $\epsilon \geq 0$, otherwise we can exchange $M_{\cdot,j}$ and $M'_{\cdot,j}$. In the proof, we will fix ϵ as a small constant and figure out the worst case of the joint distribution over X_A, X_E and original scheme M such that $|\gamma(M_{\cdot,j}) - \gamma(M'_{\cdot,j})|$ is maximized.

Recall that $\gamma(M_{\cdot,j}) = \Pr[M(X_A) = j]H_G(\Pr[\mathbf{X_E}|M(X_A) = j])$.

Part 1

We will first figure out the explicit relationship between ϵ, $\gamma(M'_{\cdot,j})$ and $\gamma(M_{\cdot,j})$.

$$\Pr[M(X_A)=j]=\sum_{x_A}\Pr[X_A=x_A]M_{x_A,j}=\Pr[X_A=i]M_{ij}+K(-i)$$

Thus, $\Pr[M'(X_A)=j]=\Pr[M(X_A)=j]+\epsilon\Pr[X_A=i]$.

$$\begin{aligned}\mathbf{q}:=\Pr[\mathbf{X_E}|M(X_A)=j]=&\frac{\Pr[\mathbf{X_E},M(X_A)=j]}{\Pr[M(X_A)=j]}\\ =&\frac{\sum_{x_A}\Pr[X_A=x_A]M_{x_A,j}\Pr[\mathbf{X_E}|X_A=x_A]}{\Pr[M(X_A)=j]}\\ =&\frac{\Pr[X_A=i]M_{ij}\Pr[\mathbf{X_E}|X_A=i]+K(-i)\mathbf{q}_{-i}}{\Pr[M(X_A)=j]}\\ &(\mathbf{q}_{-i}\text{ is a distribution over }X_E\text{ that is independent with }M_{ij})\\ =&\frac{\Pr[X_A=i]M_{ij}\mathbf{q}_i+K(-i)\mathbf{q}_{-i}}{\Pr[M(X_A)=j]}\qquad(\mathbf{q}_i:=\Pr[\mathbf{X_E}|X_A=i])\end{aligned}$$

Thus, $\Pr[\mathbf{X_E}|M'(X_A)=j]=\mathbf{q}_i\frac{\epsilon\Pr[X_A=i]}{\Pr[M(X_A)=j]+\epsilon\Pr[X_A=i]}+\mathbf{q}\frac{\Pr[M(X_A)=j]}{\Pr[M(X_A)=j]+\epsilon\Pr[X_A=i]}$.

$\Pr[\mathbf{X_E}|M'(X_A)=j]$ is a convex combination of the original forecast for X_E – $\mathbf{q}=\Pr[\mathbf{X_E}|M(X_A)=j]$ and the posterior forecast for X_E conditioning on $X_A=i$.

We define $\tau(\epsilon):=\frac{\epsilon\Pr[X_A=i]}{\Pr[M(X_A)=j]+\epsilon\Pr[X_A=i]}$. Note that

$$\tau^*:=\frac{\epsilon\Pr[X_A=i]}{1+\epsilon\Pr[X_A=i]}\le\tau(\epsilon)\le 1 \tag{4}$$

$\tau^*\le\frac{\epsilon}{1+\epsilon}$.

Then we have $\Pr[\mathbf{X_E}|M'(X_A)=j]=\tau(\epsilon)\mathbf{q}_i+(1-\tau(\epsilon))\mathbf{q}$. By replacing $\Pr[M(X_A)=j]$ and $\Pr[M'(X_A)=j]$ by $\tau(\epsilon)$ and $\Pr[X_A=i]$, we obtain

$$\gamma(M'_{\cdot,j})=\epsilon\Pr[X_A=i]\frac{1}{\tau(\epsilon)}H_G(\tau(\epsilon)\mathbf{q}_i+(1-\tau(\epsilon))\mathbf{q})$$

$$\gamma(M_{\cdot,j})=\epsilon\Pr[X_A=i]\frac{1-\tau(\epsilon)}{\tau(\epsilon)}H_G(\mathbf{q})$$

Part 2

We start to calculate the worst $\tau(\epsilon)$, $\mathbf{q}$ and $\mathbf{q}_i$ that maximizes the gap between $\gamma(M_{\cdot,j})$ and $\gamma(M'_{\cdot,j})$. We first tune $\tau(\epsilon)$ and then tune $\mathbf{q}$ and $\mathbf{q}_i$.

Part 2: Tuning $\tau(\epsilon)$

In this part, we focus on $\tau(\epsilon)$ and omit other variables. We define $g(\tau):=H_G(\tau\mathbf{q}_i+(1-\tau)\mathbf{q})$. Note that $g(\tau)$ is a convex function since H_G is a convex function and $|g(\tau)|$ is also bounded by 1 (Remark 2.1).

Then we have

$$\gamma(M'_{\cdot,j})=\epsilon\Pr[X_A=i]\frac{1}{\tau(\epsilon)}g(\tau(\epsilon))\qquad\gamma(M_{\cdot,j})=\epsilon\Pr[X_A=i]\frac{1-\tau(\epsilon)}{\tau(\epsilon)}g(0)$$

$$\begin{aligned}|\gamma(M'_{\cdot,j}) - \gamma(M_{\cdot,j})| =& |\epsilon \Pr[X_A = i](1-\tau(\epsilon))\frac{1}{\tau(\epsilon)}[g(\tau(\epsilon)) - g(0)] + \epsilon \Pr[X_A = i] g(\tau(\epsilon))| \\ \leq& |\epsilon \Pr[X_A = i]\frac{g(\tau(\epsilon)) - g(0)}{\tau(\epsilon) - 0} + |\epsilon \Pr[X_A = i]|\end{aligned}$$

$(1 - \tau(\epsilon) \leq 1$, $g(x)$ is bounded by 1 since $|H_G(\mathbf{p})|$ is bounded by 1 (Remark 2.1).)

Recall that

$$\tau^* := \frac{\epsilon \Pr[X_A = i]}{1 + \epsilon \Pr[X_A = i]} \leq \tau(\epsilon) \leq 1 \tag{5}$$

Note that $\frac{g(\tau)-g(0)}{\tau-0}$ is an increasing function of $0 \leq \tau \leq 1$ when g is a convex function, thus, $|\frac{g(\tau)-g(0)}{\tau-0}|$ is maximized in the endpoints,

$$\begin{aligned}|\gamma(M'_{\cdot,j}) - \gamma(M_{\cdot,j})| \leq& \epsilon \Pr[X_A = i]|\frac{g(\tau) - g(0)}{\tau - 0}| + \epsilon \\ \leq& \epsilon \Pr[X_A = i] \max\{|\frac{g(\tau^*) - g(0)}{\tau^* - 0}|, |\frac{g(1) - g(0)}{1 - 0}|\} + \epsilon \\ \leq& 2|[g(\tau^*) - g(0)]| + 2\epsilon \\ & (g(x) \text{ is bounded by 1 since } |H_G(\mathbf{p})| \text{ is bounded by 1 (Remark 2.1).})\end{aligned}$$

Part 2: Tuning $\mathbf{q}$ and $\mathbf{q}_i$.

It remains to compute the upper-bound of $|g(\tau^*) - g(0)|$.

$$\begin{aligned}& \max |g(\tau^*) - g(0)| \\ =& \max_{\mathbf{q}_i, \mathbf{q}} |H_G(\tau^* \mathbf{q}_i + (1-\tau^*)\mathbf{q}) - H_G(\mathbf{q})| \\ \leq& \max_{\mathbf{q}_i, \mathbf{q}} \sum_{\sigma \in \Sigma_E} |G(\tau^* \mathbf{q}_i(\sigma) + (1-\tau^*)\mathbf{q}(\sigma)) - G(\mathbf{q}(\sigma))|\end{aligned}$$

Consider $h(x, y) := |G(\tau^* x + (1-\tau^*)y) - G(y)|$.

$$|G(\tau^* x + (1-\tau^*)y) - G(y)| = |G(\tau^*(x-y) + y) - G(y)|$$

Fix $\tau^*(x-y)$. Note that $\tau^*(x-y)$ can be less than the any small constant by picking sufficiently small constant ϵ. Since $G(x)$ (Figure 1) is a convex function, $G'(x)$ is a monotone function. Thus, $|G'(x)|$ is maximized in end points. $h(x, y) = |G(\tau^*(x-y) + y) - G(y)|$ is maximized if y is one of the end points, that is, $y = 0, 1$.

When $y = 0, 1$, $h(x, y) = |G(\tau^*(x-y)+y) - G(y)| = |G(\tau^*(x-y)+y)|$. Note that $|G(x)|$ and $|G(1-x)|$ are increasing functions when x is sufficiently small when $G(x)$ is convex. Thus, to maximize $h(x, y)$, we should pick $|x - y| = 1$ to maximize $|\tau^*(x-y)|$. Therefore, $|G(\tau^*(x-y) + y)| \leq \max\{|G(\tau^*)|, |G(1-\tau^*)|\} \leq |G(\tau^*)| + |G(1-\tau^*)|$.

$$\begin{aligned}& \max |g(\tau^*) - g(0)| \\ \leq& \max_{\mathbf{q}_i, \mathbf{q}} \sum_{\sigma \in \Sigma_E} |G(\tau^* \mathbf{q}_i(\sigma) + (1-\tau^*)\mathbf{q}(\sigma)) - G(\mathbf{q}(\sigma))| \\ \leq& n_E \max_{x,y} h(x, y) \\ \leq& n_E(|G(\tau^*)| + |G(1-\tau^*)|)\end{aligned}$$

Recall that $\tau^* := \frac{\epsilon \Pr[X_A=i]}{1+\epsilon \Pr[X_A=i]} \le \epsilon$,

$$\begin{aligned}|\gamma(M'_{\cdot,j}) - \gamma(M_{\cdot,j})| \le& 2|g(\tau^*) - g(0)| + 2\epsilon \\ \le& 2n_E(|G(\tau^*)| + |G(1-\tau^*)|) + 2\epsilon \\ \le& 2n_E(|G(\epsilon)| + |G(1-\epsilon)|) + 2\epsilon \\ &\text{(when } x \text{ is small, both } |G(x)| \text{ and } |G(1-x)| \text{ are increasing functions.)}\end{aligned}$$

◀

◀

Proof of Lemma 19.

▶ **Definition 21** (Bad pair)**.** We say a pair $(x_B, x'_B), x_B, x'_B \in \Sigma_B, x_B \neq x'_B$ is bad for a event E if

$$\Pr[X_E = \cdot | E, X_B = x_B] = \Pr[X_E = \cdot | E, X_B = x'_B].$$

Intuitively, a bad pair cannot be distinguished conditioning on event E.

According to Claim 12, to guarantee the total expected payment being maximal, we should guarantee the information is fully aggregated, that is, Alice should identify all "meaningful" information from Bob. If there exists a pair $(x_B, x'_B), x_B, x'_B \in \Sigma_B, x_B \neq x'_B$ such that

$$\Pr[X_E = \cdot | X_A = x_A, X_B = x_B] = \Pr[X_E = \cdot | X_A = x_A, X_B = x'_B],$$

then distinguishing the event $X_B = x_B$ and the event $X_B = x'_B$ is not meaningful for Alice since she can think of x_B, x'_B as one signal without losing any information. Therefore, without loss of generality, *we assume every pair* (x_B, x'_B) *is good for at least one* $X_A = x_A$.

We start our construction for $M(\epsilon)$. Adopting $M(\epsilon)$ helps Alice identify all meaningful information of Bob means for every announced signal σ, there does not exist any *bad* pair $(X_B = x_B, X_B = x'_B)$ conditioning on Alice announced $X^\epsilon_\sigma = \sigma$ where $X^\epsilon_\sigma := M(\epsilon)(X_A)$.

We define $X_\sigma := M(X_A)$. We will show by changing each entry of M at most ϵ at most, we can fix a bad pair and will not produce more bad pairs. In the end, we fix all bad pairs and construct $M(\epsilon)$ that satisfies (i) and (ii).

For a bad pair (x_B, x'_B) conditioning on that event that Alice announces $X_\sigma = \sigma$ in stage 1. We know

$$\begin{aligned}1 =& \frac{\Pr[X_E = x_E | X_B = x_B, X_\sigma = \sigma]}{\Pr[X_E = x_E | X_B = x'_B, X_\sigma = \sigma]} \\ =& \frac{\Pr[X_E = x_E, X_B = x_B, X_\sigma = \sigma] \Pr[X_B = x'_B, X_\sigma = \sigma]}{\Pr[X_E = x_E, X_B = x'_B, X_\sigma = \sigma] \Pr[X_B = x_B, X_\sigma = \sigma]} \qquad (6) \\ =& \frac{\sum_{X_A = x_A} \Pr[X_E = x_E, X_B = x_B, X_A = x_A] M(x_A, \sigma)}{\sum_{X_A = x_A} \Pr[X_E = x_E, X_B = x'_B, X_A = x_A] M(x_A, \sigma)} \\ &\times \frac{\sum_{X_A = x_A} \Pr[X_B = x'_B, X_A = x_A] M(x_A, \sigma)}{\sum_{X_A = x_A} \Pr[X_B = x_B, X_A = x_A] M(x_A, \sigma)} \qquad (7)\end{aligned}$$

for all x_E.

Note that we have assumed that for every pair is good for at least one $X_A = x_A$. Suppose (x_B, x'_B) is good for $X_A = x_A$. If we add an ϵ in entry $M(x_A, \sigma)$ (if $M(x_A, \sigma) > 1 - \epsilon$ we can subtract an ϵ and the analysis is similar. We can pick sufficiently small ϵ to guarantee either $0 \le M(x_A, \sigma) + \epsilon \le 1$ or $0 \le M(x_A, \sigma) - \epsilon \le 1$) and tune other $M(x_A, \sigma'), \sigma' \neq \sigma$ arbitrarily

Table 1 Experiment inputs

	Left input		Middle input		Right input	
$\Pr[X_A = 1, \mathbf{X_B}, \mathbf{X_E}]$	0.0900	0.0900	0.2209	0.0947	0.0100	0.0100
	0.0900	0.1800	0.0947	0.0199	0.0100	0.0200
$\Pr[X_A = 2, \mathbf{X_B}, \mathbf{X_E}]$	0.0900	0.0900	0.0947	0.0406	0.0100	0.0100
	0.0900	0.2800	0.0406	0.3942	0.0100	0.9200

such that $M(x_A, \cdot)$ remains to be a valid distribution over Σ, and denote the new signaling scheme as M' and define $X'_\sigma := M'(X_A)$, we will know

$$\begin{aligned}
&\frac{\Pr[X_E = x_E | X_B = x_B, X'_\sigma = \sigma]}{\Pr[X_E = x_E | X_B = x'_B, X'_\sigma = \sigma]} \\
=&\frac{(\Pr[X_E = x_E, X_B = x_B, X_\sigma = \sigma] + \epsilon \Pr[X_E = x_E, X_B = x_B, X_A = x_A])}{(\Pr[X_E = x_E, X_B = x'_B, X_\sigma = \sigma] + \epsilon \Pr[X_E = x_E, X_B = x'_B, X_A = x_A])} \qquad (8) \\
&\cdot \frac{(\Pr[X_B = x'_B, X_\sigma = \sigma] + \epsilon \Pr[X_B = x'_B, X_A = x_A])}{(\Pr[X_B = x_B, X_\sigma = \sigma] + \epsilon \Pr[X_B = x_B, X_A = x_A])} \qquad \text{(according to formula (6))}
\end{aligned}$$

Since (x_B, x'_B) is bad for $X_\sigma = \sigma$ but good for $X_A = x_A$, there exists x_E such that

$$\frac{\Pr[X_E = x_E, X_B = x_B, X_\sigma = \sigma] \Pr[X_B = x'_B, X_\sigma = \sigma]}{\Pr[X_E = x_E, X_B = x'_B, X_\sigma = \sigma] \Pr[X_B = x_B, X_\sigma = \sigma]} = 1$$

$$\frac{\Pr[X_E = x_E, X_B = x_B, X_A = x_A] \Pr[X_B = x'_B, X_A = x_A]}{\Pr[X_E = x_E, X_B = x'_B, X_A = x_A] \Pr[X_B = x_B, X_A = x_A]} \neq 1$$

Thus there exists constant $\lambda \neq 0$ and μ such that the difference between the denominator and numerator of formula (8) can be represented as

$$\lambda\epsilon^2 + \mu\epsilon$$

Since $\lambda \neq 0$, we can always pick a sufficiently small $\epsilon_0 > 0$ such that $\lambda\epsilon^2 + \mu\epsilon \neq 0$ for all $0 < \epsilon < \epsilon_0$. For good pair (y_B, y'_B) for signal σ',

$$\frac{\Pr[X_E = x_E, X_B = y_B, X_\sigma = \sigma'] \Pr[X_B = y'_B, X_\sigma = \sigma']}{\Pr[X_E = x_E, X_B = y'_B, X_\sigma = \sigma'] \Pr[X_B = y_B, X_\sigma = \sigma']} \neq 1$$

the difference between the denominator and numerator of good pair's formula will change from $c \neq 0$ to $c + \Theta(\epsilon)$ which can be non-zero as well when ϵ is sufficiently small. Thus, we won't produce more bad pairs.

The time depends on the number of bad pairs we need to fix since we fix them one by one. The number of bad pairs is at most $O(mn_B^2)$. Thus we need $O(mn_An_B^2n_E)$ time to finish the construction and $\max_{x_A,\sigma} |M(x_A, \sigma) - M(\epsilon)(x_A, \sigma)| \leq \Theta(mn_B^2\epsilon)$. ◀

B Experiment Inputs

The proper scoring rule we use in the experiment is the logarithmic scoring rule. Now we give the input joint distributions over X_A, X_B, X_E for the three examples. All of X_A, X_B, X_E are binary random variables in this case. Therefore, we can show the input joint distribution via two 2×2 matrices. $U_1 = \Pr[X_A = 1, \mathbf{X_B}, \mathbf{X_E}]$ is a matrix where $U_1(i, j) = \Pr[X_A = 1, X_B = i, X_E = j]$ and $U_2 = \Pr[X_A = 2, \mathbf{X_B}, \mathbf{X_E}]$ is a matrix where $U_2(i, j) = \Pr[X_A = 2, X_B = i, X_E = j]$.

An Axiomatic Study of Scoring Rule Markets*

Rafael Frongillo[1] and Bo Waggoner[2]

1 **Colorado University, Boulder CO, USA**
raf@colorado.edu
2 **University of Pennsylvania, Philadelphia PA, USA**
bwag@seas.upenn.edu

Abstract

Prediction markets are well-studied in the case where predictions are probabilities or expectations of future random variables. In 2008, Lambert, et al. proposed a generalization, which we call "scoring rule markets" (SRMs), in which traders predict the value of arbitrary statistics of the random variables, provided these statistics can be elicited by a scoring rule. Surprisingly, despite active recent work on prediction markets, there has not yet been any investigation into more general SRMs. To initiate such a study, we ask the following question: in what sense are SRMs "markets"? We classify SRMs according to several axioms that capture potentially desirable qualities of a market, such as the ability to freely exchange goods (contracts) for money. Not all SRMs satisfy our axioms: once a contract is purchased in any market for prediction the median of some variable, there will not necessarily be any way to sell that contract back, even in a very weak sense. Our main result is a characterization showing that slight generalizations of cost-function-based markets are the only markets to satisfy all of our axioms for finite-outcome random variables. Nonetheless, we find that several SRMs satisfy weaker versions of our axioms, including a novel share-based market mechanism for ratios of expected values.

1998 ACM Subject Classification F.2 Algorithmic Game Theory

Keywords and phrases prediction markets, information elicitation, scoring rules

Digital Object Identifier 10.4230/LIPIcs.ITCS.2018.15

1 Introduction

The goal of a prediction market is to collect and aggregate predictions about some future outcome Y taking values in $\mathcal{Y}$; common examples arise from sporting, political, meteorological, or financial events. Prediction markets work by offering financial contracts whose payoffs are contingent on the eventually-observed value of Y. The agent's choices are interpreted, by revealed preference, as predictions about Y, and thus the final state of the market is interpreted as an aggregation of agent beliefs.

Hanson [13] observed that one can design a prediction market using a *proper scoring rule*, which is a contract $S(p', y)$ that scores the accuracy of prediction p' (a probability distribution over outcomes) upon outcome $Y = y$. The relevant guarantee is that the expected score $\mathbb{E}_{Y\sim p}S(p', Y)$ is maximized when the agent reports their true belief $p' = p$. In Hanson's *scoring rule market (SRM)*,[1] traders arrive sequentially and report their belief p_t, and are eventually paid $S(p_t, y) - S(p_{t-1}, y)$ when $Y = y$ is revealed.

* A full version of the paper is available at https://arxiv.org/abs/1709.10065
[1] We will use this term in lieu of the standard *market scoring rule (MSR)*, as the latter could refer to either the scoring rule or the market mechanism.

In many cases, a market designer may only be interested in specific statistics, called *properties*, of trader beliefs, rather than the entire distribution over the outcome Y. Lambert et al. [14] observed that SRMs can be generalized to arbitrary scoring rules $S(r, y)$, wherein traders are paid $S(r_t, y) - S(r_{t-1}, y)$; if the expected value of S is maximized by reporting the value of a particular property, in which case we say S *elicits* the property, then the market should intuitively aggregate trader beliefs about the property in question. For example, one could design a "median market" by leveraging the fact that $S(r, y) = -|r - y|$ elicits the median of the distribution of Y, and we might expect the corresponding SRM to aggregate predictions about the median of Y.

Surprisingly, apart from exploring which properties are elicitable, practically nothing is known about these general SRMs beyond those for expected values [13, 8, 2]. Here SRMs can be rephrased into a dual "cost-function-based" formulation where traders buy and sell *shares* in some securities $\phi_1, \ldots, \phi_k \in \mathbb{R}^{\mathcal{Y}}$, and the market prices reveal the trader's belief about the expected value of $\phi : \mathcal{Y} \to \mathbb{R}^k$ (§ 2.2, 5). The literature on prediction markets focuses on this setting, in which one can more easily study traditional market quantities like liquidity and depth.

In this paper, we step beyond expected values and study SRMs as a whole. Our first contribution is identifying which questions to ask of general SRMs. In particular, we ask the following.

(1) In what sense can we think of SRMs as "markets" in the traditional sense?
(2) Specifically, which SRMs behave like share-based markets, in the sense that traders can always "sell" their shares for some nontrivial price?
(3) Which SRMs if any maintain the known desirable characteristics of the cost-function-based framework, such as bounded worst-case loss and no arbitrage?

To answer these questions, we introduce axioms, and then study SRMs based on the property they elicit—as we will see, for example, markets eliciting modes will satisfy different axioms than those for medians, regardless of the scoring rules chosen. Our primary focus is the *trade neutralization (TN)* axiom, which captures question (2) above: traders holding a contract from the market maker can "sell" it for a nontrivial price, receiving more from transaction than the worst-case payoff of the contract. (Otherwise, traders would just keep their holdings regardless of their belief.)

Summary and results. After reviewing prediction markets and scoring rules (§ 2), we introduce our axioms (§ 3). We then classify modes, finite properties, medians, quantiles, and expectations, according to what axioms their markets can satisfy (§ 4).

In § 5, we give our main result: *any SRM satisfying trade neutralization must be a certain form of generalized cost-function-based market* (Theorem 16). We note that unlike in prior work on cost functions [2], this result applies to any SRM for any property; so in a sense, it says that only close relatives of expectations can be elicited by any market satisfying trade neutralization.

Our main result raises the question of whether a meaningful relaxation of trade neutralization is satisfied by a broader class of markets. We show that this is indeed the case: markets for ratios of expectations (§ 6) and for expectiles (full version of the paper) both satisfy "weak neutralization" (Axiom 5) but not trade neutralization. While these results suggest that a characterization of weak neutralization is an exciting future direction, they also yield new prediction market mechanisms worthy of attention on their own. For example, as we describe in § 6, our new market framework for ratios of expectations exchanges a security (random variable) not for cash but for units of another security, thereby revealing

trader beliefs about the *ratio* of their expectations.[2] Finally, we conclude in § 7 with open questions, and a discussion of other properties, elicitation complexity, and alternative market formulations. The Appendix contains all omitted proofs.

Relation to prior work. While we discuss related work in § 2, it is important to distinguish our main result from prior characterizations of cost-function-based markets in the literature. In particular, as mentioned above, cost-function-based markets are known to be equivalent to scoring rules that elicit expected values [13, 8, 2]. In contrast, we ask which among a very broad family of mechanisms satisfy certain basic "market" axioms, and find that such mechanisms must fall into (a minor generalization of) the cost-function-based framework. Indeed, based on the above known equivalence, we further conclude that our axioms imply the elicitation of expected values, though weakening them slightly allows us to elicit ratios of expectations (§ 6) and expectiles (full version) while still retaining some sense of a "market".

2 Aggregating Information with Prediction Markets

The goal of a prediction market is to crowdsource and aggregate beliefs of participants about some future event. It does so by allowing participants to buy and sell contracts which have different payoffs depending on the outcome of the event, and inferring a prediction from the participants' choices. In this section, we formally define the class of such markets that we study, *scoring rule markets (SRMs)*, with references to previous work. For other related work, see above and § 7.

2.1 Outcomes and contracts

Let $\mathcal{Y}$ denote the *outcome space* of interest to the market designer and Y a future event or random variable taking values in $\mathcal{Y}$. The designer will in general allow traders to select *contracts* $d \in \mathbb{R}^{\mathcal{Y}}$ from a list offered by the market, which can be thought of as vectors when $\mathcal{Y}$ is a finite set. The interpretation of a contract d is that, when Y is eventually revealed, the market maker will pay the agent a net payoff $d(Y)$, which may be negative. Its expected payoff under distribution p on Y is written $\mathbb{E}_p\, d(Y)$, which could be thought of as an inner product between p and d.

Note that under this formulation, any initial payment the agent might make is folded in to d. Specifically, letting $\mathbb{1} \in \mathbb{R}^{\mathcal{Y}}$ denote the "all-ones" contract $\mathbb{1}(y) = 1\, \forall y \in \mathcal{Y}$, then a contract d could be written $d = d' - \alpha\mathbb{1}$, interpreted as paying a price of α now for the contract d', whose payoff will be revealed when Y is observed. We assume agents are indifferent to timing of payments and just wish to maximize total expected payoff. We may occasionally abuse terminology by referring both to contracts d which "fold in" initial payments and to contracts d' having an additional "price" α.

2.2 Automated market makers

When designing prediction markets, rather than a typical continuous double auction ("stock exchange") mechanism, it is common to employ a centralized *automated market maker*, which

[2] To see this, note that a trader willing to "buy" one unit of $d \in \mathbb{R}^{\mathcal{Y}}$ in exchange for c units of $b \in \mathbb{R}^{\mathcal{Y}}$ is effectively expressing the belief $\mathbb{E}\, d > c\, \mathbb{E}\, b$, i.e., $\mathbb{E}\, d / \mathbb{E}\, b > c$. Similarly, selling at this "price" reveals the belief $\mathbb{E}\, d / \mathbb{E}\, b < c$.

offers to buy or sell any available contracts, and through which all trades are executed. See [2] for practical reasons behind this design choice.

Formally, a sequence of participants (traders) arrive at times $t = 1, \dots, T$ and each selects a contract from a list offered by the market at that time. It will be convenient to let the set of contracts available be indexed at each time by some *report space* $\mathcal{R} \subseteq \mathbb{R}^k$. Following Abernethy et al. [5], we consider a generic market making algorithm, termed a *mechanism*, that specifies the set of contracts available at each time. In general, this may depend on the entire past history of the market, and is represented as a mapping F that, given a report $r \in \mathcal{R}$ of the participant, returns the corresponding contract. More formally, the contract given to a participant who chooses report r_t given the past history of reports $r_1, \dots, r_{t-1}$ is denoted $\vec{F}(r_t | r_1, \dots, r_{t-1}) \in \mathbb{R}^{\mathcal{Y}}$. The net payoff to the trader upon outcome $y \in \mathcal{Y}$ will be denoted $F(r_t, y | r_1, \dots, r_{t-1})$.

A popular instantiation of such a mechanism is the *cost-function-based market maker*, in which $\mathcal{R} = \mathbb{R}^k$ and $F(r_t, y | r_1, \dots, r_{t-1}) = (r_t - r_{t-1}) \cdot \phi(y) - (C(r_t) - C(r_{t-1}))$, where $C : \mathbb{R}^k \to \mathbb{R}$ is convex and $\phi : \mathcal{Y} \to \mathbb{R}^k$ [2]. This payoff function corresponds to a trader making a fixed payment $C(r_t) - C(r_{t-1})$ in return for a bundle $r = r_t - r_{t-1} \in \mathbb{R}^k$ of *shares* in the securities $\phi_1, \dots, \phi_k \in \mathbb{R}^{\mathcal{Y}}$, i.e. r_i units of security ϕ_i, for an up-front cost paid to the market maker in terms of C. Among the several nice properties of this market maker, one can see that a trader who believes $\mathbb{E}\,\phi = x$ (a vector in $\mathbb{R}^k$, the component-wise expectation) has an incentive to buy or sell securities until $\nabla C(r_t) = x$, thereby revealing their belief.

2.3 Scoring rule markets for properties

The goal of the market is to incentivize a good prediction for some property or statistic of Y, such as the median or mean. Thus, much work considers prediction markets relying on *proper scoring rules*, which are contracts designed to elicit a single agent's belief (i.e. the case $T = 1$ of a market) [7, 12]. While originally designed to elicit an entire distribution over the outcome Y, in many cases, for example when $\mathcal{Y}$ is very large or even infinite, one may be interested in obtaining only summary information about this distribution. It is therefore natural to consider scoring rules which elicit such statistics, or *properties*, of distributions.

Here and throughout the paper, $\mathcal{P}$ is a set of distributions of interest on the domain $\mathcal{Y}$, for example, the distributions with full support, with finite expectation, or so on.

▶ **Definition 1.** A *property* is a function $\Gamma : \mathcal{P} \rightrightarrows \mathcal{R}$, which associates a set of correct report values to each distribution. We require Γ to be *non-redundant*, meaning $\Gamma^{-1}(r) \not\subseteq \Gamma^{-1}(r')$ for any reports $r, r' \in \mathcal{R}$ (i.e. we cannot have $r \in \Gamma(p) \implies r' \in \Gamma(p)$ for all p). A property is *single-valued* if each p maps to exactly one report, in which case we write Γ as a function $\Gamma : \mathcal{P} \to \mathcal{R}$.

A scoring rule $S(r, y)$ simply provides a payoff based on a reported value of the property and the observed outcome y. We say the scoring rule *elicits* the property if the correct report is incentivized.

▶ **Definition 2.** A scoring rule $S : \mathcal{R} \times \mathcal{Y} \to \mathbb{R}$ *elicits* a property $\Gamma : \mathcal{P} \rightrightarrows \mathcal{R}$ if for all $p \in \mathcal{P}$, $\Gamma(p) = \operatorname{argmax}_{r \in \mathcal{R}} \mathbb{E}_p S(r, Y)$ where $Y \sim p$. When Γ is single-valued, this condition becomes $\{\Gamma(p)\} = \operatorname{argmax}[\cdots]$. A property is *elicitable* if some scoring rule elicits it.

Some properties are not elicitable, meaning there is no way to score reports of their value based on an observed outcome in a manner which incentivizes truthfulness. A classic example is the variance of a distribution, which does not have convex level sets (mixtures of distributions with the same variance have higher variance in general), a necessary condition

for elicitability [14]. (For such non-elicitable statistics, one could still discuss their elicitation *complexity*, the number of reports needed to compute the desired property post-hoc; we discuss this concept in § 7.) However, several well-known statistics are elicitable properties, including expected values, medians, quantiles, expectiles, and ratios of expectations.

Combining the concept of scoring rules for properties with scoring rule markets yields the following natural prediction market mechanism for an arbitrary property $\Gamma : \mathcal{P} \rightrightarrows \mathbb{R}$ elicited by S [13, 14, 4]. Initialize the market state at some $r_0 \in \mathcal{R}$. When trader $t = 1, \dots, T$ arrives, they can choose to update the market state to any $r_t \in \mathcal{R}$, and once the outcome $y \in \mathcal{Y}$ is revealed, the market maker will pay the trader $S(r_t, y) - S(r_{t-1}, y)$. We can again express this mechanism using our F notation above (we will later relate F to Γ).

▶ **Definition 3.** A *scoring rule market* for scoring rule $S : \mathcal{R} \times \mathcal{Y} \to \mathbb{R}$ and initial state $r_0 \in \mathcal{R}$ is the mechanism $F(r_t, y | r_1, \dots, r_{t-1}) = S(r_t, y) - S(r_{t-1}, y)$.

For brevity, we will simply write $F(r', y|r) = S(r', y) - S(r, y)$, or using contract notation $\vec{F}(r'|r) = \vec{S}(r') - \vec{S}(r)$, where of course $\vec{S}(r)_y = S(r, y)$.

3 Axioms and Preliminaries

To motivate our choice of SRMs, we consider two preliminary axioms in the full version of the paper that turn out to characterize SRMs; here, we briefly summarize. *Incentive-compatibility (IC)* states that agents maximize expected payment by choosing contracts that reveal their true belief about $\Gamma(p)$, for the property Γ chosen by the market maker. We note that previous work [2, 5] did not explicitly consider IC, but it arose as a consequence of cost-function markets revealing expected values. In this work, however, IC is a primary axiom as we consider markets designed to elicit particular properties, but also a weak one in that essentially every scoring rule elicits some (potentially set-valued) property. *Path-independence (PI)*, an axiom appearing in prior work on prediction markets [2], ensures that a participant cannot gain more by making multiple trades in a row than by simply making the single optimal trade immediately. In the full version, we show that any mechanism which satisfies both PI and IC is an SRM for a property Γ. It is conceptually similar to results for markets eliciting the mean [2, 5], but more general as it holds for any property.

▶ **Theorem 4.** *A mechanism satisfies PI and IC for property Γ if and only if it is a scoring rule based market (SRM) with some scoring rule S that elicits Γ.*

Henceforth, unless otherwise noted we assume PI and IC, which is to say we focus on SRMs.

3.1 Arbitrage and Bounded Loss

We now express two time-honored axioms in our notation: no arbitrage, and bounded worst-case loss. Recall that we write $\mathbb{1} \in \mathbb{R}^{\mathcal{Y}}$ to mean the contract paying out $\mathbb{1}(y) = 1$ for each $y \in \mathcal{Y}$. We also will write $\inf d = \inf_{y \in \mathcal{Y}} d(y)$ and $\sup d = \sup_{y \in \mathcal{Y}} d(y)$ to denote the payout bounds of contract $d \in \mathbb{R}^{\mathcal{Y}}$; note that we use infima and suprema as $\mathcal{Y}$ may be infinite, for example when $\mathcal{Y} = \mathbb{R}$.

The following axiom has been a desiderata since the inception of prediction markets: the market maker should not risk losing an unbounded amount of money.

▶ **Axiom 1** (Worst-Case Loss (WCL))**.** An SRM F, initialized at r_0, satisfies WCL with bound $B \geq 0$ if for all $r \in \mathcal{R}$, $\sup \vec{F}(r|r_0) \leq B$.

As of course $\vec{F}(r|r_0) = \vec{S}(r) - \vec{S}(r_0)$, WCL simply says that the scoring rule S is bounded relative to the initial score $S(r_0, \cdot)$.

Another classic condition for a market mechanism is that a trader should never have an opportunity to make a guaranteed (risk-free) profit. In our contract notation, this means traders should never be able to purchase a contract d for less than its minimum payoff $\inf d$, or equivalently, the net contract provided to traders should not be unconditionally positive.

▶ **Axiom 2** (No Arbitrage (ARB))**.** SRM F satisfies ARB if $\forall\, r, r' \in \mathcal{R}$, $\inf \vec{F}(r'|r) \leq 0$.

Equivalently, F satisfies ARB if for every trade, there is an outcome yielding a profit at most 0. The no-arbitrage condition was crucial in deriving cost-function-based markets in terms of convex conjugate duality [2]. Surprisingly, it turns out that any SRM satisfies no-arbitrage automatically if it is incentive-compatible.

▶ **Proposition 5.** *SRM F satisfies ARB if it is IC for some property $\Gamma \rightrightarrows \mathcal{R}$.*

Proof. If $0 < \inf \vec{F}(r'|r) = \inf[\vec{S}(r') - \vec{S}(r)]$, then for all $y \in \mathcal{Y}$, $S(r', y) > S(r, y)$. Thus, for any $p \in \mathcal{P}$, $\mathbb{E}_p S(r', Y) > \mathbb{E}_p S(r, Y)$, so r cannot be an optimal report for any p, contradicting non-redundancy (as $\Gamma^{-1}(r) = \emptyset \subseteq \Gamma^{-1}(r')$). ◀

3.2 New Axioms

We now identify several new axioms that capture desirable characteristics of prediction markets. The first is motivated by traditional markets, wherein a trader can always "neutralize" or "liquidate" their holdings. For example, if a trader buys a bar of gold, apart from apocalyptic scenarios, she can always sell it at any time for a strictly positive price. As another example, a trader holding an Arrow-Debreu contract, paying \$1 upon event E and \$0 otherwise, should be able to sell it for some nonzero (perhaps very low) price. More generally, if the contract purchased is $d \in \mathbb{R}^{\mathcal{Y}}$ and is non-constant, the trader should receive strictly more than $\inf d$ in cash (in both examples above, $\inf d = 0$). This condition is the *trade neutralization (TN)* axiom, which we now give along with two variants, one stronger and one weaker.

▶ **Axiom 3.** An SRM satisfies *Trade Neutralization (TN)* if for all trades $r_1 \to r_1'$, and all reports r_2, there is a trade $r_2 \to r_2'$ such that $\vec{F}(r_1'|r_1) + \vec{F}(r_2'|r_2) = c\mathbb{1}$ for some scalar $c > \inf \vec{F}(r_1'|r_1)$.

▶ **Axiom 4.** An SRM satisfies *Portfolio Neutralization (PN)* if for all sets of trades $r_i \to r_i'$, $1 \leq i \leq m$, and all reports r, there is a trade $r \to r'$ such that $\vec{F}(r'|r) + \sum_i \vec{F}(r_i'|r_i) = c\mathbb{1}$ for some scalar $c > \inf \left[\sum_{i=1}^m \vec{F}(r_i'|r_i)\right]$.

▶ **Axiom 5.** An SRM satisfies *Weak Neutralization (WN)* if for all trades $r_1 \to r_1'$, and all reports r_2, there is a trade $r_2 \to r_2'$ such that $\inf\left[F(r_1'|r_1) + F(r_2'|r_2)\right] > \inf F(r_1'|r_1)$.

Portfolio neutralization (PN) is stronger than TN: for any *set* of purchased contracts, a trader should be able to sell the entire portfolio for more than its minimum payoff, all in one go, at any time. Surprisingly, we will see that in fact TN is equivalent to PN (Theorem 16).[3] Similarly, weak neutralization (WN) is weaker than TN, as it only requires that a trader

[3] Note that this equivalence does not follow from path independence (PI), which only refers to *consecutive* sequences of trades. PN is most interesting for nonconsecutive trades, e.g. the same contract $r \to r'$ purchased multiple times.

be able to improve her worst-case payoff, but not necessarily render her holdings outcome-independent. More precisely, WN states that traders holding a non-constant contract d should be able to purchase a contract d' such that $\inf[d+d'] > \inf d$. Note that WN is not far off from TN in the sense that a WN market maker allowing traders to exchange any contract for the cash value of its worst-case payout, a favorable option for the market maker, would satisfy TN: a trader can purchase d' as above, and then exchange $d+d'$ for $(\inf(d+d'))\mathbb{1}$, thus neutralizing while improving the worst-case payout. In fact, path independence (PI) ensures that the trader can do both in a single trade.

Note that the inequalities above are all strict. If we replaced them by weak inequalities, then WN would be trivially satisfied by essentially every conceivable mechanism: by just keeping your current contract d, you are guaranteed a payoff of at least $\inf d$, by definition. This makes intuitive sense, as unless the contract d is constant, it strictly dominates $(\inf d)\mathbb{1}$, so no rational trader would ever consent to trading d for $(\inf d)\mathbb{1}$. The strict inequality thus captures a reasonable possibility for traders to "sell back" their previously-purchased contracts, in the sense that there is some belief a trader could hold where such a trade would be beneficial. More importantly, the strict inequalities ensure that the market maker does indeed "make a market": the defining quality of a market maker is a willingness to buy or sell securities at some price, but prices equal to the best- or worst-case payoff of the securities are equivalent to refusing to buy or sell, as no rational trader regardless of belief would accept such a trade.

Our final axiom says that traders with arbitrarily small budgets who disagree with the current prediction can always make some profitable trade. In other words, as the overall market scales larger relative to the maximum allowable loss ϵ of individual traders, they still have incentives to participate (although naturally, the "size" of their updates will be small relative to the entire market). Thus, the following axiom captures scalability of a market both in terms of large numbers of individually small trader budgets and informational updates.

▶ **Axiom 6.** A market F satisfies *bounded trader budget (BTB)* if, for all market states r and beliefs p with $\Gamma(p) \neq r$, and for all $\epsilon > 0$, there exists a contract $F(r'|r)$ with $\inf F(r'|r) > -\epsilon$ and $\mathbb{E}_p F(r', Y|r) > 0$.

4 Central Examples

With our axioms in hand, we now turn to specific properties: the mode, median, quantiles, and expectations. For each, we wish to understand which axioms the corresponding SRMs can satisfy. As we will see, each of these properties has a unique signature with respect to our axioms.

4.1 Mode and Finite Property Markets

Perhaps the simplest example of any SRM is the canonical mode market for a distribution on k outcomes $\mathcal{Y} = \{1, \ldots, k\}$, with report space $\mathcal{R} = \mathcal{Y}$ and $S(r, y) = \mathbb{1}\{y = r\}$ ("\$1 iff you guess correctly"). Here we find that even the weakest version of neutralization, WN, is violated: a trade moving r from 1 to 2 yields payoff $\mathbb{1}\{y = 2\} - \mathbb{1}\{y = 1\}$, meaning this contract will lose the owner 1 if $Y = 1$. But if the current market state is, say, 3, then no report the agent can make will avoid this loss of 1 when $Y = 1$. For any report r', the final payoff is $\mathbb{1}\{y = 2\} - \mathbb{1}\{y = 1\} - \mathbb{1}\{y = 3\} + \mathbb{1}\{y = r'\}$. See Figure 1.

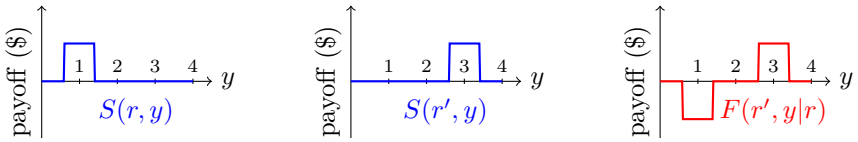


Figure 1 Visualizing trader "position" in a mode market. Each curve gives the payoff of a given scoring rule or contract as a function of the outcome y. Here $S(r, y) = 1 \iff r = y$. Left: A trader who has reported $r = 1$ stands to gain 1 if $Y = 1$ and gain 0 otherwise. Center: Similarly for $r = 3$. Right: A trader who moves the market from r to r' gets the function $F(r', \cdot|r) = S(r', \cdot) - S(r, \cdot)$. She stands to gain 1 if $Y = r'$, lose 1 if $Y = r$, and gain 0 otherwise.

Intuitively, the lack of WN has negative implications for agents, who must take on risk that cannot be mitigated later. It also violates BTB: The potential agent loss from any trade is a constant 1, so an agent with a budget smaller than 1 will not be able to participate. This causes significant problems in practice for market designers as well, because the only possible solution, scaling down the scoring rule, is also unpleasant: many agents will be able to participate without much risk, so the market prediction will flip from outcome to outcome without necessarily aggregating information.

We emphasize that these characteristics are inherent to the mode, in the sense that they hold for any other scoring rule eliciting it. In fact, the same conclusions hold for markets eliciting any *finite property* $\Gamma : \mathcal{P} \to \{1, \ldots, k\}$, i.e. a property with a finite set of possible values; see [15] for motivation and examples of such properties.

▶ **Theorem 6.** *Any market for a finite property satisfies WCL, but not TN or BTB.*

4.2 Mean and expectation markets

Consider now the mean of a random variable over the reals, $\mathcal{R} = \mathcal{Y} = \mathbb{R}$. (In this setting, we naturally take $\mathcal{P}$ to be the set of beliefs with well-defined expectations.) As an illustrative example, consider the scoring rule $S(r, y) = 2ry - r^2$, which elicits the mean. The corresponding SRM takes the form $F(r', y|r) = r^2 - (r')^2 + 2y(r' - r)$. Perhaps the first observation one makes is that, because $y \in \mathbb{R}$, any nontrivial trade leaves the trader exposed to unbounded potential loss, as well as unbounded potential gain. This implies that BTB and WCL cannot hold.

What about trade neutralization? Consider a trader holding the contract $\vec{F}(r_1', y|r_1) = \alpha_1 + \alpha_2 y$ for constants $\alpha_1 = (r_1)^2 - (r_1')^2$ and $\alpha_2 = 2(r_1' - r_1)$. She would like to neutralize this position, so she must purchase some other contract of the form $\alpha_3 - \alpha_2 y$, so that her net position will be the constant $\alpha_1 + \alpha_2$. If the current market state is r_2, our hero can neutralize her previous trade by choosing $r_2' = (r_1 - r_1')$, so that $2y(r_2' - r_2) = 2y(r_1' - r_1) = 0$. Her worst-case liability decreases from $-\infty$ to a constant, showing that both WN and TN are satisfied. And in fact, even if she holds a set of contracts, her net position is simply the sum and can still be written in the form $\alpha_1 + \alpha_2 y$ and the same argument goes through; this shows that PN is also satisfied.

This example raises the questions of whether other markets eliciting the mean (if any) would have similar properties, and how this might depend on the random variable in question.

Motivated by this example, we now consider a much more general setting. Let $\phi : \mathcal{Y} \to \mathbb{R}^k$ be a given "random variable" and let $\Gamma(p) = \mathbb{E}_p[\phi(Y)]$ be its expected value. Such a Γ is called a *linear property*. We call SRM S an *expectation market* if S is IC for such a Γ. For

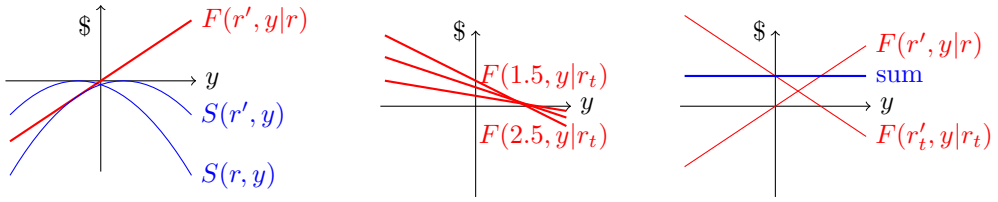


Figure 2 Trader position in a mean market. $S(r, y) = -(r - y)^2$. Left: Moving the market from $r = -1$ to $r' = 1$ gives a position $F(r', \cdot|r)$ that pays off linearly in y. Center: At r_t, the trader chooses from a set of possible contracts, depending on if she reports 1.5, 2.5, etc. Right: choosing the contract r'_t that neutralizes the previous trade $r \to r'$.

ease of exposition (i.e. to avoid relative interiors) we will assume that the range $\mathcal{R} = \Gamma(\mathcal{P})$ is full-dimensional in $\mathbb{R}^k$.

Capitalizing on the known characterization of scoring rules for expectations [1, 10] (Theorem 18), we know that any scoring rule S eliciting Γ takes the form,

$$S(r, y) = G(r) + dG_r \cdot (\phi(y) - r) + f(y) \ , \tag{1}$$

for G strictly convex with subgradients $\{dG_r\}_{r \in \mathcal{R}}$ and f an arbitrary $\mathcal{P}$-integrable function. Note however that the f term will vanish in the definition of $F(r', y|r)$, so without loss of generality we may take $f(y) = 0$ for all y. We thus refer to expectation markets as being *defined by* G if they satisfy Equation 1 for that G (for any f). Letting $\text{conv}(\phi(\mathcal{Y}))$ be the convex hull of the set of outcomes of the random variable ϕ, we have the following characterization of worst-case loss for expectation markets.

▶ **Proposition 7.** *The linear property $\Gamma(p) = \mathbb{E}_p \phi(Y)$ has an expectation market satisfying WCL if and only if its domain $\text{conv}(\phi(\mathcal{Y}))$ is bounded, in which case the market defined by any bounded G satisfies WCL.*

▶ **Proposition 8.** *The linear property $\Gamma(p) = \mathbb{E}_p \phi(Y)$ has an expectation market satisfying BTB if its domain $\text{conv}(\phi(\mathcal{Y}))$ is bounded, in which case the market defined by any differentiable G satisfies BTB.*

To illustrate, let us revisit the example at the beginning of this subsection, i.e. $\phi(y) = y$ and $S(r, x) = 2ry - r^2$. (This corresponds to the convex function $G(r) = r^2$, up to a shift.) We saw that this market satisfies neither BTB nor WCL for $\mathcal{Y} = \mathbb{R}$, the entire real line. However, for $\mathcal{Y} = (0, 1)$, it satisfies both BTB and WCL, as $\text{conv}(\phi(\mathcal{Y})) = (0, 1)$ and G is bounded and differentiable on this domain. Finally, recall that it also (intuitively) satisfied TN, and furthermore, PN. These qualities can be shown to generalize; in particular, prior work has shown that linear properties have nice *cost function* based markets. Such markets take the form $F(r', y \mid r) = C(r) - C(r') + (r - r') \cdot \phi(y)$, for some convex function C and $\mathcal{R} = \mathbb{R}^k$. It then follows relatively directly that such markets satisfy PN, as any position (set of contracts held by a trader) can be written in the form $d(y) = \alpha_1 + \alpha_2 \cdot \phi(y)$. This can be interpreted as a fixed payment of α and $\alpha_{2,i}$ "shares" in a "security" $\phi(Y)_i$. To neutralize, when the current market state is r', it turns out to suffice to select the r satisfying $r - r' = -\alpha_2$.

▶ **Theorem 9.** *On a finite outcome space, i.e. $|\mathcal{Y}| < \infty$, for any linear property there exists an expectation market satisfying PN, WCL, and BTB.*

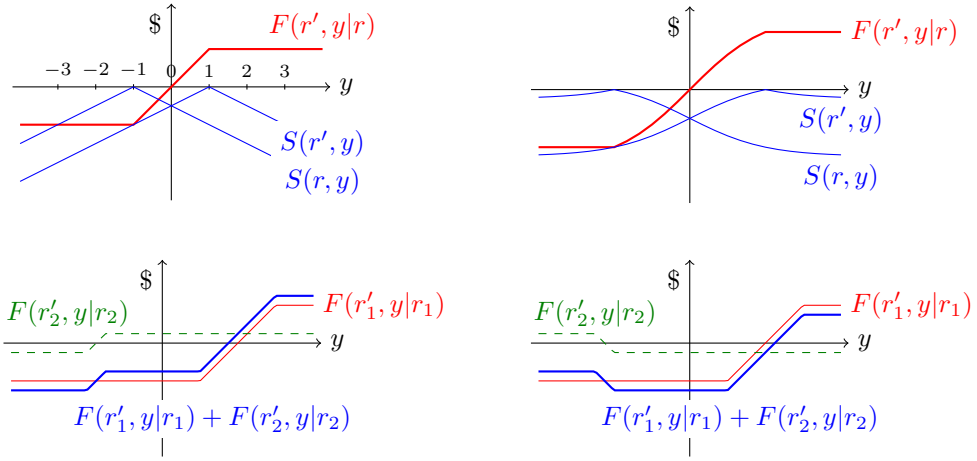


Figure 3 Trader position in a median market. Here $S(r, y) = -|r - y|$. Top left: The payoffs for $S(r, \cdot)$ and $S(r', \cdot)$, and the contract that moves the market $r \to r'$, namely $F(r', \cdot|r) = S(r', \cdot) - S(r, \cdot)$. Top right: The same example but with $S(r, y) = -|g(r) - g(y)|$ with g the sigmoid function. Bottom: Two examples where a trader with position "red" $F(r'_1, y|r_1)$ considers a potential contract "green" $F(r'_2, y|r_2)$. If purchased, the net position will be the blue curve. It sometimes falls below the original position, meaning the trader's worst case has gotten worse.

4.3 Median and Quantile Markets

We previously gave an example of a "mean market" given by $S(r, y) = 2ry - r^2$, the scoring rule analog of squared loss $L(r, y) = (r-y)^2$. Perhaps the most natural statistic to investigate next is the median, elicited by the analog of "absolute loss", $S(r, y) = -|r - y|$. What we find is surprising: unlike squared loss, the absolute loss market does not satisfy PN, and in fact does not even satisfy WN. We show a general version of this result: no market for the median, or indeed any quantile, can satisfy WN. On the other hand, while the mean market could not satisfy WCL except on bounded domains, there are median and quantile markets that can.

Our setting in this subsection is as follows. Letting $\mathcal{R} = \mathcal{Y} = \mathbb{R}$ and $\alpha \in (0, 1)$, the *α-quantile* of probability distribution p with continuous CDF is the q_α satisfying $\Pr_p[Y \leq q_\alpha(p)] = \alpha$. (The continuity assumption is dropped in the full version of the paper.) Of course, the median is simply q_α for $\alpha = 1/2$.

We first show that quantile markets do not satisfy WN: there may not be trades which improve the trader liability at all. Figure 3 gives an example with absolute loss.

▶ **Theorem 10.** *No SRM for any α-quantile satisfies WN.*

Despite this negative result, quantile markets satisfy a surprisingly strong positive property. Recall that the squared loss market with $S(r, y) = 2ry - r^2$, which elicits the mean of Y, could not hope to satisfy bounded worst-case loss if $\mathcal{Y} = \mathbb{R}$. And indeed, the absolute loss market $S(r, y) = -|r - y|$ shares this issue. There is an elegant work-around, however: one can use the sigmoid function $g(r) = e^r/(1 + e^r)$, or another strictly monotone transformation, to map reports continuously into the interval $(0, 1)$. Then $S(r, y) = -|g(r) - g(y)|$ is clearly bounded, but still proper, as strictly monotone functions commute with the median: for any y, all $y' \leq y$ are mapped below y and all $y' \geq y$ are mapped above y, so the quantiles are simply mapped as well.

▶ **Theorem 11.** *For all $\alpha \in (0, 1)$, there is an SRM for the α-quantile satisfying WCL.*

Finally, quantiles also behave nicely with respect to bounded-budget traders. To see this for $S(r, y) = -|r - y|$, notice that a small trade $r \to r + \epsilon$ carries a liability of only ϵ, in the case $Y < r$. (See Figure 3.) But any trade smaller than ϵ can carry positive expected payoff for a trader if it moves the market closer to her belief.

▶ **Theorem 12.** *If distributions in $\mathcal{P}$ do not contain point masses, then any α-quantile market satisfies BTB.*

5 Characterizing Trade Neutralization

In this section, we will characterize mechanisms that satisfy trade neutralization (TN) when the set of outcomes is finite, $|\mathcal{Y}| = n < \infty$. Recall that TN captures the quality of traditional markets, particularly in the sense of a "market maker": the mechanism is always willing to buy back a contract it has previously sold, for a reasonable price.

TN may seem intuitively easy to satisfy, or at least might not appear to impose much structure on the market maker. In fact, we will see that the opposite is the case. To gain intuition for why TN might impose structure, recall that trades made by a participant must be interpretable as beliefs about the property the market is predicting. This applies to both the purchase of a contract and its sale back to the mechanism. Furthermore, this "canceling" trade must be available regardless of the current state of the market. We will leverage this intuition to show that the mechanisms satisfying TN are quite special and closely related to expectation markets.

In our analysis, it will prove useful to separate out "contracts" $d \in \mathbb{R}^{\mathcal{Y}}$ from "cash" $\mathbb{1} \in \mathbb{R}^{\mathcal{Y}}$, even though technically both reside in the same space. In particular, we would like a canonical way to take a contract d and separate out its "cash" component as $d = d_0 + c\mathbb{1}$, where intuitively \$$c$ is always paid regardless of the outcome, but d_0 depends on outcome. To do this, we simply define d_0 as the projection of d onto the hyperplane normal to $\mathbb{1}$.

We define the range of a scoring rule S by $\mathcal{S} = \{\vec{S}(r) : r \in \mathcal{R}\} \subseteq \mathbb{R}^{\mathcal{Y}}$, and define the corresponding "cashless" contract space by $\mathcal{H} \doteq \mathcal{S}/\mathbb{1} = \{\vec{S}(r) - (\vec{S}(r) \cdot \mathbb{1}/|\mathcal{Y}|)\mathbb{1} : r \in \mathcal{R}\}$. (Recall that one can define $\vec{S}(r) = \vec{F}(r|\emptyset)$. We indicate throughout where we assume IC.) Again, $\mathcal{H}$ is simply the projection of the range of $\vec{S}$ onto the hyperplane normal to $\mathbb{1}$. Similarly, we define $\mathcal{D} \doteq \mathcal{H} - \mathcal{H} = \{h_2 - h_1 : h_1, h_2 \in \mathcal{H}\}$ to be the set of possible score differences modulo $\mathbb{1}$. Equivalently, $\mathcal{D}$ is the projection of the range of $\vec{F}$ onto the same hyperplane.

We now show that if an SRM satisfies TN, its corresponding set $\mathcal{D}$ must have considerable structure: it must form an additive subgroup of $\mathbb{R}^{\mathcal{Y}}$.

▶ **Lemma 13.** *If SRM F satisfies TN, then $\mathcal{D}$ is an additive subgroup of $\mathbb{R}^{\mathcal{Y}}$ in the sense that $d, d' \in \mathcal{D}$ implies $-d, d + d' \in \mathcal{D}$. Moreover, the entire set $\mathcal{D}$ of contracts is available at all times: for any $h \in \mathcal{H}$, we have $\mathcal{D} = \mathcal{H} - \{h\} = \{h' - h \,|\, h \in \mathcal{H}\}$.*

Proof. From the definition of TN, we must have the following: $\forall r_1, r_2, r_3 \exists r_4$ with $F(r_2|r_1) + F(r_4|r_3) = c\mathbb{1}$ for some $c \in \mathbb{R}$. In terms of $\mathcal{H}$, as we have taken everything modulo $\mathbb{1}$, this implies $\forall h_1, h_2, h_3 \in \mathcal{H}\ \exists h_4 \in \mathcal{H}$ such that $h_2 - h_1 + h_4 - h_3 = 0$. In other words, $h_1, h_2, h_3 \in \mathcal{H} \implies h_3 + h_1 - h_2 \in \mathcal{H}$.

To show that $\mathcal{D}$ is a group under vector addition, we must show closure under addition and the existence of additive inverses. The latter is simple, as given any $d \in \mathcal{D}$ we can write $d = h_2 - h_1$, so of course $-d = h_1 - h_2 \in \mathcal{D}$. We also have $0 = h_1 - h_1 \in \mathcal{D}$. For closure, consider $d, d' \in \mathcal{D}$, and write $d = h_2 - h_1$ and $d' = h_2' - h_1'$. By the above, $(h_2 + h_2' - h_1') \in \mathcal{H}$, so $h_2 - (h_1 + h_1' - h_2') = d + d' \in \mathcal{D}$.

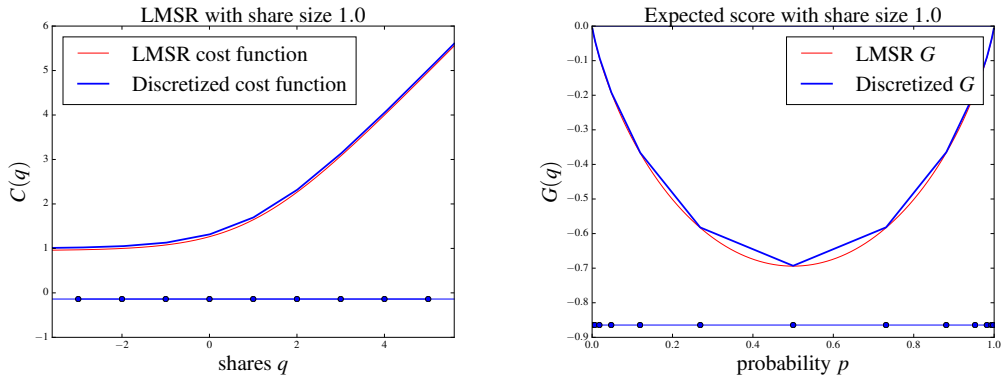


Figure 4 The "discretized" LMSR. This market for $\Pr[Y = 1]$ allows only integer trades of a single security, using the LMSR cost function $C(q) = \log(1 + e^q)$ and its conjugate $G(p) = p\log(p) + (1-p)\log(1-p)$. Traders may move the market from any "dot" to any other. The possible market states are equally spaced in share space (horizontal axis, left figure), translating to *non*-equally spaced *prices* (horizontal axis, right figure). Theorem 16 implies that all mechanisms satisfying PI, IC, and TN have this general format, being discretizations of cost-function markets.

Finally, we must show that for any $h \in h$, we have $\mathcal{D} = \mathcal{H} - \{h\}$. Clearly $\mathcal{H} - \{h\} \subseteq \mathcal{D}$; for the converse, consider $d = h_1 - h_2$. By the above, $h + h_1 - h_2 \in \mathcal{H}$, and thus $d = (h + h_1 - h_2) - h \in \mathcal{H} - \{h\}$. ◀

The implication of Lemma 13 is that under TN, we must have a highly structured set $\mathcal{D}$ of possible contracts (modulo cash) available at all times with only the cash price for each contract changing depending on history. For example, a construction allowed by Lemma 13 is $\mathcal{D} = \mathbb{Z}^k$ (the integer lattice of dimension k), and this is indeed possible. Looking ahead, we will see that this would be interpreted as offering to buy or sell an integer number of "shares" in each of k different "securities" at any time. We give a $k = 1$ example in Figure 4.

We will need the following generalization of standard cost-function based markets.

▶ **Definition 14.** The *generalized cost-function based market* parameterized by a convex "cost function" $C : \mathbb{R}^k \to \mathbb{R}$, "securities" $\phi : \mathcal{Y} \to \mathbb{R}^k$, and "share space" $\mathcal{Q} \subseteq \mathbb{R}^k$ is the SRM of the form $F(r', y \mid r) = C(r) - C(r') + (r' - r) \cdot \phi(y)$ with report space at market state r given by $\mathcal{R} = \{r + q : q \in \mathcal{Q}\}$.

Recall that in standard cost-function markets, $\mathcal{Q} = \mathbb{R}^k$, so that any trade $d = r - r' \in \mathbb{R}^k$ is allowable. Furthermore, here d_i is interpreted as the number of shares purchased of security ϕ_i. (For example, in the $\mathcal{D} = \mathbb{Z}^k$ construction mentioned above, we also have $\mathcal{Q} = \mathbb{Z}^k$, so traders may only purchase integer numbers of shares in this example.)

▶ **Definition 15.** A cost-function based market defined by C and ϕ is *open* if C is differentiable and[4] $\{\nabla C(q) : q \in \mathbb{R}^k\} = \text{int}(\text{conv}(\phi(\mathcal{Y})))$. It is *quasi-open* if for every pair of "share vectors" $q, v \in \mathcal{Q}$ and subgradient $x \in \partial C(q)$, we have $x \cdot v < \max_{y \in \mathcal{Y}} v \cdot \phi(y)$.

To see that quasi-openness is a relaxation of openness, we comment that if we required the condition to hold for all $v \in \mathbb{R}^k$, then it would be equivalent to $\cup_{q \in \mathcal{Q}} \partial C(q) \subseteq \text{int}(\text{conv}(\phi(\mathcal{Y})))$, which is a clear relaxation of openness; and quasi-openness requires somewhat less, as it only must hold for all $v \in \mathcal{Q}$.

[4] We write $\text{int}(A)$ for the interior of the set A and $\text{conv}(A)$ for its convex hull.

▶ **Theorem 16.** *Let SRM F which is IC for Γ be given. Then F satisfies TN if and only if it is a generalized cost-function-based market for securities $\phi : \mathcal{Y} \to \mathbb{R}^k$, where:*
(i) *The share space $\mathcal{Q}$ of possible purchases is an additive subgroup of $\mathbb{R}^k$, and*
(ii) *The market is quasi-open.*
Moreover, TN implies PN.

Proof. For the forward direction, that TN implies cost-function-based and PN, we proceed in five parts: (1) construct a basis for $\mathcal{D}$, which will be the securities ϕ; (2) rewrite $\vec{F}$ in terms of that basis; (3) show that the map from reports to "shares" is bijective; (4) extract the cost function C from the share representation of F; (5) show that C satisfies our relaxed version of openness. For the converse, we will appeal to Theorem 22.

1. Basis of $\mathcal{D}$. By Lemma 13, $\mathcal{D}$ is an additive subgroup of $\mathbb{R}^{\mathcal{Y}}$. Let $d^1, \dots, d^k \in \mathcal{D}$ be a basis for the linear span of $\mathcal{D}$. We can now write our securities in terms of this basis: define $\phi : \mathcal{Y} \to \mathbb{R}^k$ by $\phi(y)_i = d^i_y$. It will be convenient to work with ϕ in matrix form $\Phi \in \mathbb{R}^{n \times k}$, which naturally we define as $\Phi_{y,i} = \phi(y)_i$. Thus, we now have $\mathcal{D} = \{\Phi \cdot q : q \in \mathcal{Q}\}$ where $\mathcal{Q}$ is an additive subgroup of $\mathbb{R}^k$.

2. Rewrite $\vec{F}$. By the definition of $\mathcal{D}$ as the range of F modulo $\mathbb{1}$, and the decomposition above, we know that for every $r, r' \in \mathcal{R}$ we can write $\vec{F}(r'|r) = \Phi \cdot v + c\mathbb{1}$ for some $v \in \mathbb{R}^k$ and $c \in \mathbb{R}$. Thus, letting r_0 be the initial state of the given SRM, we have functions $g : \mathcal{R} \to \mathbb{R}$ and $v : \mathcal{R} \to \mathbb{R}^k$ such that for all $r \in \mathcal{R}$, $\vec{F}(r|r_0) = \Phi \cdot v(r) + g(r)\mathbb{1}$. By definition $\Phi \cdot v(r) \in \mathcal{D}$ and thus we must have $v(r) \in \mathcal{Q}$, meaning we can write $v : \mathcal{R} \to \mathcal{Q}$. Note that the expected payoff when $Y \sim p$ can now be written $\mathbb{E}_p F(r, Y|r_0) = p^\top \vec{F}(r|r_0) = p^\top \Phi\, v(r) + g(r) p^\top \mathbb{1} = \mathbb{E}_p \phi \cdot v(r) + g(r)$. In particular, Γ can only depend on p through $\mathbb{E}_p \phi$, so letting $\mathcal{X} = \mathrm{conv}(\phi(\mathcal{Y}))$ as before, we have some $\psi : \mathcal{X} \rightrightarrows \mathcal{R}$ such that $\Gamma(p) = \psi(\mathbb{E}_p \phi)$.

3. The map $v : \mathcal{R} \to \mathcal{Q}$ is a bijection. Surjectivity of v follows from the fact that transformation by the change of basis matrix Φ is a bijection, as by definition we are simply rewriting elements of $\mathcal{D}$ via elements of $\mathcal{Q}$. For injectivity, we will use IC. Suppose $v(r) = v(r')$. If $g(r) = g(r')$, Γ is not non-redundant as $\vec{F}(r|r_0) = \vec{F}(r'|r_0)$, so clearly $\Gamma(p) = r \iff \Gamma(p) = r'$. Thus without loss of generality $g(r) > g(r')$, which implies $r' \notin \operatorname{argmax}_{x \in \mathcal{R}} v(x) \cdot \mathbb{E}_{p'} \phi(Y) + g(x) = \operatorname{argmax}_{x \in \mathcal{R}} \mathbb{E}_p F(x, Y|r_0)$ for any $p' \in \mathcal{P}$, so r' cannot be in the range of Γ at all (i.e. the report r dominates r'), a contradiction. Thus, v is injective and hence a bijection.

4. Extracting the cost function. Now that we have a bijection from reports to "shares", intuitively, we should just be able to take $C(q) = -g(v^{-1}(q))$. More care is needed, however, as C must be convex with the correct subgradients. Let us define a convex function $G : \mathcal{X} \to \mathbb{R}$ by $G(x) = \sup_r v(r) \cdot x + g(r)$, which is the optimal expected score when $\mathbb{E}_p \phi = x$. By IC and the definition of ψ above, we have $G(x) = v(r) \cdot x + g(r)$ if and only if $r \in \psi(x)$. Now we can define $C : \mathbb{R}^k \to \mathbb{R}$ as the conjugate of G, namely $C(q) = \sup_{x \in \mathcal{X}} q \cdot x - G(x)$. Thus, we need only show that $C(v(r)) = -g(r)$, as alluded to above, as this would imply $F(r, y|r_0) = v(r) \cdot \phi(y) - C(v(r))$ as desired.

To make headway, we will appeal to convex analysis. First, we will establish that $G(x) = v(r) \cdot x + g(r) \iff v(r) \in \partial G(x)$, meaning if r is the optimal report for expected value x, then $v(r)$ is a subgradient of G. This follows directly from the subgradient inequality; if r is optimal for x, then for all x', $G(x') = sup_{r'} v(r') \cdot x + g(r') \geq v(r) \cdot x + g(r) = G(x) + v(r) \cdot (x' - x)$, so $v(r) \in \partial G(x)$. Conversely, suppose $v(r) \in \partial G(x)$ but r is optimal for some x', so $G(x') = v(r) \cdot x' + g(r)$. The subgradient inequality gives $v(r) \cdot x' + g(r) = G(x') \geq G(x) + v(r) \cdot (x' - x)$, which implies $v(r) \cdot x + g(r) \geq G(x)$, so r must be optimal for x as well: $G(x) = v(r) \cdot x + g(r)$.

In summary, we have $r \in \psi(x) \iff G(x) = v(r) \cdot x + g(r) \iff v(r) \in \partial G(x)$. By [18, Thm E.1.4.1], we immediately have $C(q) = q \cdot x - G(x) \iff q \in \partial G(x)$. Putting these

together, we have $r \in \psi(x) \iff v(r) \in \partial G(x) \iff C(v(r)) = v(r) \cdot x - G(x) = -g(r)$. Thus, as every $r \in \mathcal{R}$ is in the image of ψ, we must have $C(v(r)) = -g(r)$ for all r.

5. C is (almost) open. We have established that F can be written as a cost-function-based market with securities ϕ and a potentially nondifferentiable C. To conclude this direction of the proof, suppose that for some $x \in \partial C(q)$ we have $x \cdot v \geq \max_{y \in \mathcal{Y}} v \cdot \phi(y)$. Following the proof of Theorem 22, we start the trader at q selling v (as $(-v) \in \mathcal{Q}$) and place the state back at q. To neutralize, the trader must now purchase $v \in \mathcal{Q}$, but by the same argument as before (using the fact that $\partial C(\mathbb{R}^k) \subseteq \mathcal{X}$ by construction of C), $C(q + v) - C(q) = x \cdot v \geq \max_{y \in \mathcal{Y}} v \cdot \phi(y)$, which contradicts TN.

Converse. Finally, for the reverse direction, we can simply repeat the proof of Theorem 22: if the trades v_i are elements of the group $\mathcal{Q}$ for all v_i, then so is $v = \sum_{i=1}^{w} v_i$. It only remains to be shown that the price is less than the worst-case payoff of v, which follows from integrating the guarantee in (ii) as in Lemma 21. ◀

Theorem 16 shows that, despite the apparent freedom allowed by the TN axiom, the only SRMs satisfying TN are cost-function-based, or variations thereof (see Figure 4). By way of contrast, note that apart from other axiomatic characterizations with axioms such as "information incorporation", previous work has only established that cost-function-based market makers elicit expectations and satisfy TN [2], whereas here we show that in fact they are the *only* markets satisfying TN.

Given that a market satisfying TN must elicit expectations (or "discretized" expectations), the following natural question arises: does WN allow for a broader family of markets, for properties beyond expectations? If false, this would mean that TN and WN were equivalent in a formal sense. We now give an example which answers this question in the affirmative: a market for the ratio of expectations which (by virtue of not eliciting a discretized expected value) does not satisfy TN, but does satisfy WN. In the full version we give another such example: expectiles. Together, these examples show that WN is a potentially useful condition for the design of non-expectation prediction markets, and warrants further exploration.

6 Share-Like Market for Ratios of Expectations

From Theorem 16, we know that any SRM satisfying the TN axiom will effectively be a cost-function-based market dealing in shares of some set of securities, perhaps restricting trades to some discretization of the share space. As a corollary, all such TN markets must elicit an expected value either directly or indirectly, or perhaps a discretization thereof. Theorem 16 leaves open the possibility, however, of a "share-like" market for a property other than an expected value which satisfies WN but not TN. We now give one such example: a ratio of expectations.

Before diving into the formalism, let us expand on the intuition given in § 1. In a cost-function-based market, traders purchase bundles $r \in \mathbb{R}^k$ of securities $\phi : \mathcal{Y} \to \mathbb{R}^k$, in exchange for paying some up-front cost $C(r_t + r) - C(r_t)$ to the market maker in the form of cash. In other words, traders are exchanging the bundle r of securities ϕ for an amount $C(r_t + r) - C(r_t)$ of the security $\mathbb{1}$. The ratio of expectations market can be thought of similarly, but now traders exchange the bundle r of securities ϕ for an amount $C(r_t+r)-C(r_t)$ of some new security b, which by convention will be nonnegative (and bounded away from 0 for infinite $\mathcal{Y}$). In other words, the new market is exactly the same as the old, but the cost function is in units of the security b rather than cash ($\mathbb{1}$). Immediately we can glean that, just as in a cost-function-based market traders with belief p had an incentive to trade until $\nabla C(r_t) = \mathbb{E}_p \phi$, i.e. the change in cost is equal to the expected security payoff, now traders

will trade until $\nabla C(r_t)\mathbb{E}_p b = \mathbb{E}_p \phi$. Thus, the prices of the new market will be the market's consensus about the ratio of expectations of ϕ and b, $\nabla C(r_t) = \mathbb{E}_p\phi / \mathbb{E}_p b$.

What axioms does this new market satisfy? It would seem that its share-like nature could somehow circumvent Theorem 16 and satisfy TN, but of course this is not the case. As we will show, however, the ratio of expectations market does satisfy WN. First, we must formally define our new market.

Let $\mathcal{Y}$ be a finite set. For $\phi : \mathcal{Y} \to \mathbb{R}^k$ and $b : \mathcal{Y} \to \mathbb{R}$ with $\inf b > 0$, we define the ratio of expectations $\Gamma(p) = \mathbb{E}_p\phi / \mathbb{E}_p b$. As usual we assume that ϕ is affinely independent, so $\{\mathbb{E}_p\phi : p \in \mathcal{P}\}$ is full-dimensional, and that $\inf b > 0$, which since $\mathcal{Y}$ is finite is equivalent to the payoffs $b(y)$ being positive for all $y \in \mathcal{Y}$. A characterization of scoring rules eliciting Γ was shown by Frongillo and Kash [10], which extended the real-valued case given by Gneiting [11]: a scoring rule S elicits Γ if and only if,

$$S(r, y) = b(y)G(r) + dG_r \cdot (\phi(y) - rb(y)) + f(y) \ , \tag{2}$$

where as in Theorem 18, $G : \mathcal{R} \to \mathbb{R}$ is strictly convex with selection of subgradients dG, and f is arbitrary (and, as usual, irrelevant for SRMs).

Our main result for this section is that essentially all "open" markets for Γ satisfy WN. The proof first applies convex conjugate duality to arrive at the cost-function-like market described above, and then applies facts about the usual cost-function-based framework (e.g. Lemma 21) to show that the prices for trading in a bundle must be strictly less than its worst-case payoff.

▶ **Theorem 17.** *A ratio-of-expectations market for differentiable G and $\nabla G(\mathcal{R}) = \mathbb{R}^k$ satisfies WN.*

Proof. We construct a "cost function" $C(q) = \sup_{r \in \mathcal{R}} r \cdot q - G(r)$ as the convex conjugate of G. The corresponding scoring rule is $S_C(q, y) = \phi(y) \cdot q - C(q)b(y)$. By assumption, $\nabla G : \mathcal{R} \to \mathbb{R}^k$ is a bijection, and thus markets F_G with G and F_C with C are equivalent. It therefore suffices to show that F_C satisfies WN.

Consider any trade $q_1 \to q_1'$ and let $v = q_1' - q_1$. For any q_2, we will show that the trade $q_2' = q_2 - v$ satisfies WN. The difference in net payoff between the first and second trade is $(C(q_2) - C(q_2 - v))b(y) - v \cdot \phi(y)$, so it suffices to show that this is always positive.

By the proof of Lemma 21, we then have for any $q, w \in \mathbb{R}^k$

$$C(q + w) - C(q) < \sup_{x \in \mathcal{R}} x \cdot w = \max_{y \in \mathcal{Y}} (\phi(y)/b(y)) \cdot w \ . \tag{3}$$

Taking $q = q_2$ and $w = -v$, we have $C(q_2) - C(q_2 - v) < \phi(y) \cdot (-v)/b(y)$ for all y, which is equivalent to $(C(q_2 - v) - C(q_2))b(y) > \phi(y) \cdot v$ for all y, thus establishing WN. ◀

By establishing WN for SRMs eliciting a ratio of expectations, we are essentially saying that these markets are "reasonable" and could plausibly aggregate trader beliefs effectively. It would be interesting to see how such markets perform in practice.

7 Discussion and Directions

We have presented market axioms for scoring rule markets (SRMs) and studied a handful of well-known statistics/properties to see which of these axioms their corresponding markets can satisfy. Interestingly, we have seen wide variation among the satisfied axioms, most dramatic of which are the neutralization axioms: TN is satisfied essentially exclusively by cost-function-based markets, whereas median/quantile and mode markets do not even satisfy

its weakest version WN. This raised the question of whether any market satisfied WN but not TN, i.e., whether any non-cost-function-based markets satisfyied WN. We saw that indeed markets for ratios of expectations are one such example, and in the full version we show that markets for expectiles are another. As discussed in § 3, WN is nearly equivalent to TN, and thus these markets could be expected to perform similarly to cost-function-based markets.

We conclude with future directions and discussion.

Open questions. Our work leaves several questions open. There are of course many other properties to explore and categorize with respect to our axioms, and perhaps the most compelling question along these lines would be a characterization of SRMs satisfying WN, or of properties elicited by WN SRMs. We would also like to show that the share space $\mathcal{Q}$ in Theorem 16 is in fact isomorphic to to $\mathbb{R}^{k_1} \oplus \mathbb{Z}^{k_2}$, meaning that shares could be purchased with arbitrary precision in some directions but in units of some smallest purchase in others; this isomorphism would also mean that IC, TN, and BTB would imply cost-function-based markets without qualification. Finally, a construction of pseudo-barriers for arbitrary convex sets would allow Proposition 8 to be an "if and only if", and would be interesting in their own right.

Other market forms. [19] study prediction markets from a more empirical perspective, and suggest a number of possible market formulations. One which is not covered here is to offer contracts d^r such that $\mathbb{E}_p d^r = 0 \iff \Gamma(p) = r$. (This $V(r, y) = d^r(y)$ is called an *identification function*.) They illustrate this idea with a market eliciting the median of Y, where d^r pays \$1 if $Y > r$ and $-$\$1 otherwise. While the properties of such a contract space would be interesting to study in a continuous double-auction, one may ask how to translate it to an automated market maker. Here the market maker would maintain a centralized median r_t and traders could either buy or sell d^{r_t}, moving $r_{t+1} \leftarrow r_t \pm \epsilon$. Unfortunately, for any $\epsilon > 0$, such a market has unbounded worst-case loss even on a bounded domain (traders buy and sell between $r = 0$ and $r = \epsilon$ and $y = \epsilon/2$). For infinitesimal ϵ however, the market becomes the absolute-loss SRM with $S(r, y) = -|r - y|$, because $d^r = \frac{d}{dr} S(r, y)$.

Complexity. Finally, we remark that the recent concept of *elicitation complexity* brings interesting implications for our study. Here one asks, given a property Γ of interest, how many dimensions k does one need for there to exist an elicitable $\Gamma' : \mathcal{P} \to \mathbb{R}^k$ from some "nice" class of properties, where one can compute Γ from Γ' via a link function f, i.e. $\Gamma = f \circ \Gamma'$. In our context, one may choose "nice" to mean a property having an SRM satisfying any one of the axioms we discuss. The most natural may be TN, in which case one is essentially asking Γ' to be an expected value, a case studied by [6]. It would be interesting to characterize properties having markets satisfying WN, and identify properties with low complexity with respect to these WN properties.

Acknowledgements. Thanks to Yiling Chen for helpful comments and pointers, and several referees for constructive feedback. This material is based upon work supported by the National Science Foundation under Grant No. 1657598.

References

1 J. Abernethy and R. Frongillo. A characterization of scoring rules for linear properties. In *Proceedings of the 25th Conference on Learning Theory*, pages 1–27, 2012. URL: `http://jmlr.csail.mit.edu/proceedings/papers/v23/abernethy12/abernethy12.pdf`.

2 Jacob Abernethy, Yiling Chen, and Jennifer Wortman Vaughan. Efficient market making via convex optimization, and a connection to online learning. *ACM Transactions on Economics and Computation*, 1(2):12, 2013.

3 Jacob Abernethy, Sindhu Kutty, Sébastien Lahaie, and Rahul Sami. Information aggregation in exponential family markets. In *Proceedings of the fifteenth ACM conference on Economics and computation*, pages 395–412. ACM, 2014.

4 Jacob D. Abernethy and Rafael M. Frongillo. A collaborative mechanism for crowdsourcing prediction problems. In *Advances in Neural Information Processing Systems 24*, pages 2600–2608, 2011.

5 Jacob D. Abernethy, Rafael M. Frongillo, Xiaolong Li, and Jennifer Wortman Vaughan. A general volume-parameterized market making framework. In Moshe Babaioff, Vincent Conitzer, and David Easley, editors, *ACM Conference on Economics and Computation, EC '14, Stanford , CA, USA, June 8-12, 2014*, pages 413–430. ACM, 2014. doi:10.1145/2600057.2602900.

6 Arpit Agarwal and Shivani Agarwal. On consistent surrogate risk minimization and property elicitation. In *JMLR Workshop and Conference Proceedings*, volume 40, pages 1–19, 2015. URL: http://www.jmlr.org/proceedings/papers/v40/Agarwal15.pdf.

7 G.W. Brier. Verification of forecasts expressed in terms of probability. *Monthly weather review*, 78(1):1–3, 1950.

8 Y. Chen and D.M. Pennock. A utility framework for bounded-loss market makers. In *Proceedings of the 23rd Conference on Uncertainty in Artificial Intelligence*, pages 49–56, 2007.

9 Rafael Frongillo and Ian Kash. General truthfulness characterizations via convex analysis. In *Web and Internet Economics*, pages 354–370. Springer, 2014.

10 Rafael Frongillo and Ian Kash. Vector-Valued Property Elicitation. In *Proceedings of the 28th Conference on Learning Theory*, pages 1–18, 2015.

11 T. Gneiting. Making and Evaluating Point Forecasts. *Journal of the American Statistical Association*, 106(494):746–762, 2011.

12 Tilmann Gneiting and Adrian E. Raftery. Strictly proper scoring rules, prediction, and estimation. *Journal of the American Statistical Association*, 102(477):359–378, 2007.

13 R. Hanson. Combinatorial Information Market Design. *Information Systems Frontiers*, 5(1):107–119, 2003.

14 Nicolas S. Lambert, David M. Pennock, and Yoav Shoham. Eliciting properties of probability distributions. In *Proceedings of the 9th ACM Conference on Electronic Commerce*, pages 129–138, 2008.

15 Nicolas S. Lambert and Yoav Shoham. Eliciting truthful answers to multiple-choice questions. In *Proceedings of the 10th ACM conference on Electronic commerce*, pages 109–118, 2009.

16 R.T. Rockafellar. *Convex analysis*, volume 28 of *Princeton Mathematics Series*. Princeton University Press, 1997.

17 L.J. Savage. Elicitation of personal probabilities and expectations. *Journal of the American Statistical Association*, pages 783–801, 1971.

18 Jean-Baptiste Hiriart Urruty and Claude Lemaréchal. *Fundamentals of Convex Analysis*. Springer, 2001.

19 J. Wolfers and E. Zitzewitz. Prediction Markets. *Journal of Economic Perspective*, 18(2):107–126, 2004.

A Proofs for Mean and Expectation Markets

We note the characterization of scoring rules for the expectation, which has been established with varying degrees of generality [17, 1, 10]:

▶ **Theorem 18.** *A scoring rule S elicits $\Gamma : p \mapsto \mathbb{E}_p[\phi(Y)]$ if and only if $S(r, y) = G(r) + dG_r \cdot (\phi(y) - r) + f(y)$ for some G strictly convex with subgradients $\{dG_r\}_{r\in\mathcal{R}}$, and f an arbitrary $\mathcal{P}$-measurable function.*[5]

▶ **Proposition (7).** *The linear property $\Gamma(p) = \mathbb{E}_p\phi(Y)$ has an expectation market satisfying WCL if and only if its domain $\mathrm{conv}(\phi(\mathcal{Y}))$ is bounded, in which case the market defined by any bounded G satisfies WCL.*

Proof. Lemma 19 proves that on a bounded domain, WCL is equivalent to boundedness of G. (Some bounded convex G always exists for a bounded domain.) Lemma 20 proves that WCL cannot be satisfied on an unbounded domain. ◀

▶ **Lemma 19.** *On a bounded domain $\mathrm{conv}(\phi(\mathcal{Y}))$, an expectation market defined by G satisfies WCL if and only if G is bounded.*

Proof. Let the initial market state be any r where $d_G(r)$ is finite, such as any r in the interior of the domain.[6] We show that for this fixed r, worst-case loss is bounded by a constant if and only if G is bounded.

Recall from the characterization that $S(r, y) = G(r) + d_G(r) \cdot \phi(y)$. By bounded domain, $\|\phi(y)\|$ is bounded, so there exists a constant B with $-B \le S(r, y) \le B$ for all y.[7]

The loss of the market maker when the final state is r' is $S(r', y) - S(r, y)$, and the worst-case loss is $WCL \doteq \sup_{r',y} [S(r', y) - S(r, y]$. We have $WCL \in \left(\sup_{r',y} S(r', y)\right) \pm B$, hence it is bounded if and only if the first term is. For each y, the supremum over r is achieved at $r = \phi(y)$ by properness of the scoring rule (r is the optimal report for the distribution δ_y), so the first term is $\sup_y S(\phi(y), y) = \sup_y G(\phi(y))$. So worst-case loss is bounded if and only if G is. ◀

▶ **Lemma 20.** *On an unbounded domain, no expectation market satisfies WCL, assuming the initial market prediction lies in the interior of $\mathcal{R}$.*

Proof. Let r be the market's initial starting point. We assume that r is in the interior of the convex hull of $\{\phi(y) : y \in \mathcal{Y}\}$, and in particular lies in an ϵ-ball contained in the interior. The idea is to pick a "direction" and consider a sequence of outcomes that are farther and farther away. If the final market state is some distance in that direction, then loss will be unbounded.

To formalize this, let $y_1, y_2, \dots$ be a sequence with $\|\phi(y_i) - r\|$ positive and increasing without bound. Such a sequence must exist by unboundedness of the domain. Consider the associated sequence of size-ϵ vectors $\{v_i = \epsilon\frac{\phi(y_i)-r}{\|\phi(y_i)-r\|} : i = 1, 2, \dots\}$. As a sequence in a compact set (the ϵ-sphere), it has a convergent subsequence with a limit v. This is the

[5] Typically scores are allowed to take on values in $\mathbb{R} \cup \{\infty\}$, essentially to accommodate the log scoring rule, but we will typically restrict to the relative interior of the domain anyhow, thus avoiding this issue.

[6] This technical condition only rules out boundary such as the log scoring rule with initial prediction $p = 0$.

[7] If $f(y) \neq 0$, then we can say $S(r, y) + f(y) \in [-B, B] + f(y)$, and we would see $f(y)$ cancel out later in the proof.

"direction". By monotonicity of strictly convex functions, we have $\gamma \doteq v \cdot (dG_{r+v} - dG_r) > 0$. Let $K \subseteq \mathbb{N}$ be the indices of this subsequence.

Now consider the following sequence of markets closing states and outcomes, indexed by $i \in K$. Each market i has initial state r and final state $r + v$. The outcome is $Y = y_i$. The initial state is valid because r was fixed at the beginning of the proof, and $r + v$ is a valid final closing state because it lies on the ϵ-ball around r, assumed to lie in the report space because r was in the interior. The worst-case loss is at least the following quantity, which is then rearranged:

$$\begin{aligned}
\text{Loss} &= S(r+v, \phi(y_i)) - S(r, \phi(y_i)) \\
&= G(r+v) - G(r) + dG_{r+v} \cdot (\phi(y_i) - r - v) - dG_r \cdot (\phi(y_i) - r) \\
&= [G(r+v) - G(r) - dG_r \cdot (r + v - r)] + (dG_{r+v} - dG_r) \cdot (\phi(y_i) - r - v) \\
&\geq (dG_{r+v} - dG_r) \cdot (\phi(y_i) - r - v)
\end{aligned}$$

using the definition of subgradient to conclude that the bracketed term is at least zero. Now we divide both sides by $\|\phi(y_i) - r\|$ and recall our definition of v_i:

$$\frac{\text{Loss}}{\|\phi(y_i) - r\|} \geq \epsilon \left[(dG_{r+v} - dG_r) \cdot v_i\right] - \left[(dG_{r+v} - dG_r) \cdot \frac{v}{\|\phi(y_i) - r\|}\right].$$

The first term on the right side converges to $\epsilon\gamma$ because $v_i \to v$; the second term is equal to $\gamma / \|\phi(y_i) - r\|$ which converges to zero. The ratio on the left therefore either diverges or converges to some constant larger than $\epsilon\gamma$, and in either case, worst-case loss is unbounded. ◀

▶ **Proposition** (8). *The linear property $\Gamma(p) = \mathbb{E}_p \phi(Y)$ has an expectation market satisfying BTB if its domain $\text{conv}(\phi(\mathcal{Y}))$ is bounded, in which case the market defined by any differentiable G satisfies BTB.*

Proof. Suppose G is differentiable and its domain is bounded, e.g. $\|x\| \leq B$ for all x. Let x^0 be the market state and consider belief μ. By monotonicity of strictly convex functions, any trade $x' = \alpha\mu + (1-\alpha)x^0$ has strictly positive expected score. Meanwhile, the worst-case score for any trade from x^0 to x' is

$$\begin{aligned}
&\sup_x G(x^0) - G(x') + dG_{x'} \cdot (x - x') - dG_{x^0} \cdot (x - x^0) \\
&= \sup_x G(x^0) - G(x') + x \cdot (dG_{x'} - dG_{x^0}) - x' \cdot dG_{x'} - x^0 \cdot dG_{x^0} \\
&\leq B\|dG_{x'} - dG_{x^0}\| + O(\|x^0 - x'\|)
\end{aligned}$$

where both terms can be made arbitrarily small with $\alpha \to 0$, $x' \to x^0$: G is continuous, and it is a convex differentiable function so dG is as well. ◀

▶ **Theorem** (9). *On a finite outcome space, i.e. $|\mathcal{Y}| < \infty$, for any linear property there exists an expectation market satisfying PN, WCL, and BTB.*

Proof. We utilize Theorem 22, which says that if a cost-function based market is *open* (Definition 15), then it satisfies PN.

The primary examples are *exponential-family* markets [3], where

$$C(q) = \log \sum_{y \in \mathcal{Y}} \exp(q \cdot \phi(y)),$$

the "log-partition" function. We recall the key facts behind the construction and refer the reader to [3]. One interprets $q \cdot \phi(y)$ as a "weight" on y and define the (exponential-family) distribution $p \in \text{int}(\Delta_{\mathcal{Y}})$ with $p_y = \frac{\exp(q \cdot \phi(y))}{\sum_{y \in \mathcal{Y}} \exp(q \cdot \phi(y))}$. One obtains

$$\nabla C(q) = \sum_y p_y \phi(y)$$

so the set of gradients is exactly the interior of $\text{conv}(\phi(\mathcal{Y}))$, i.e. the market is open.

By Theorem 22, this implies TN. The convex conjugate $G(\mu)$ is bounded (equaling the negative entropy of p); this implies WCL by Lemma 19. Finally, it and C are both strictly convex and differentiable. This implies BTB by Proposition 8. ◀

▶ **Lemma 21.** *Given $\phi : \mathcal{Y} \to \mathbb{R}^k$ such that $\mathcal{X} := \text{conv}(\phi(\mathcal{Y}))$ is full-dimensional in $\mathbb{R}^k$, let $C : \mathbb{R}^k \to \mathbb{R}$ be convex with subgradients $\partial C(\mathbb{R}^k) \subseteq \text{int}(\mathcal{X})$. Then For all $q, v \in \mathbb{R}^k$, $\max_{y \in \mathcal{Y}} v \cdot \phi(y) > C(q + v) - C(q)$.*

Proof. As $\partial C(q') \subseteq \text{int}(\mathcal{X})$ for all $q' \in \mathbb{R}^k$, in particular $dC(q') \cdot v < \max_{x \in \mathcal{X}} x \cdot v = \max_{y \in \mathcal{Y}} v \cdot \phi(y)$ [18, Prop A.2.4.6], where dC is a selection of subgradients of C. By [9, Thm B.4], the function $t \mapsto dC(q + tv) \cdot v$ is monotone and therefore integrable, and thus $C(q + v) - C(q) = \int_{t=0}^{1} dC(q + tv) \cdot v \, dt < \max_{y \in \mathcal{Y}} v \cdot \phi(y)$. ◀

▶ **Theorem 22.** *Cost-function-based markets satisfy TN if and only if they are open. Moreover, if they satisfy TN, the satisfy PN.*

Proof. Suppose a cost-function-based market is open. We will show that it satisfies not only TN but PN. Consider a non-empty set of bundles purchased $v_i = q_i' - q_i$ for $1 \leq i \leq m$, for a total cost of $c = \sum_i C(q_i') - C(q_i)$, and let $v = \sum_{i=1}^{m} v_i$ be their sum. Clearly, to neutralize this position from the current market state q, the trader must sell v, for a cost of $C(q - v) - C(q)$. After this trade, the trader's total contract is $c\mathbb{1} - (C(q - v) - C(q))\mathbb{1}$, so to establish PN, we need only show $c - C(q - v) + C(q) > \inf_{y \in \mathcal{Y}}[c + (-v) \cdot \phi(y)]$. Subtracting c from both sides and then negating, the rest follows from observing sup = max as $\mathcal{Y}$ is finite, and applying Lemma 21.

Now suppose a cost-function-based market satisfies TN; we will show it is open. By assumption, such a market has a differentiable C with $\text{cl}\left(\{\nabla C(q) : q \in \mathbb{R}^k\}\right) = \text{conv}(\phi(\mathcal{Y})) =: \mathcal{X}$. Suppose the market is not open; this implies that $\nabla C(q)$ lies on the boundary of $\mathcal{X}$ for some q. As $\mathcal{X}$ is a convex polytope, $\nabla C(q)$ must lie on an exposed face of $\mathcal{X}$, so let $v \in \mathbb{R}^k$ be a direction exposing that face, meaning $\max_{x \in \mathcal{X}} x \cdot v = \nabla C(q) \cdot v$ [18, Sec A.2.4].

Now suppose the trader buys the bundle $(-v)$ at state q, and the market state has returned to q, which corresponds to $q_1 = q$, $q_1' = q - v$, $q_2 = q$. To satisfy TN, the trader must now purchase v, but we must additionally have $\inf[(C(q) - C(q - v))\mathbb{1} + (C(q) - C(q + v))\mathbb{1}] > \inf[(C(q) - C(q - v))\mathbb{1} + (-v) \cdot \phi]$, which is equivalent to $C(q + v) - C(q) < \max_{y \in \mathcal{Y}} v \cdot \phi(y)$. By weak monotonicity [16, Thm 24.9], the map $t \mapsto \nabla C(q + tv) \cdot v$ is monotone increasing in t, and thus as $\nabla C(q + tv) \in \mathcal{X}$ and $\nabla C(q) \cdot v = \max_{x \in \mathcal{X}} x \cdot v$, we have $\nabla C(q + tv) \cdot v = \nabla C(q) \cdot v$ for all $t \geq 0$. But now we have $C(q + v) - C(q) = \int_{t=0}^{1} \nabla C(q + tv) \cdot v dt = \nabla C(q) \cdot v = \max_{y \in \mathcal{Y}} \phi(y) \cdot v$, so the trader's minimum payoff has not increased, violating TN. ◀

On Price versus Quality

Avrim Blum[*1] **and Yishay Mansour**[†2]

1 **Toyota Technological Institute at Chicago, USA**
avrim@ttic.edu
2 **Tel-Aviv University, and Google Research, Israel**
mansour.yishay@gmail.com

Abstract

In this work we propose a model where the value of a buyer for some product (like a slice of pizza) is a combination of their personal desire for the product (how hungry they are for pizza) and the quality of the product (how good the pizza is). Sellers in this setting have a two-dimensional optimization problem of determining both the quality level at which to make their product (how expensive ingredients to use) and the price at which to sell it. We analyze optimal seller strategies as well as analogs of Walrasian equilibria in this setting. A key question we are interested in is: to what extent will the price of a good be a reliable indicator of the good's quality?

One result we show is that indeed in this model, price will be a surprisingly robust signal for quality under optimal seller behavior. In particular, while the specific quality and price that a seller should choose will depend highly on the specific distribution of buyers, for optimal sellers, price and quality will be linearly related, independent of that distribution. We also show that for the case of multiple buyers and sellers, an analog of Walrasian equilibrium exists in this setting, and can be found via a natural tatonnement process. Finally, we analyze markets with a combination of "locals" (who know the quality of each good) and "tourists" (who do not) and analyze under what conditions the market will become a tourist trap, setting quality to zero while keeping prices high.

1998 ACM Subject Classification F.0 General

Keywords and phrases Algorithmic game theory, pricing, Cobb-Douglas valuations

Digital Object Identifier 10.4230/LIPIcs.ITCS.2018.16

1 Introduction

Buyers often use price as a simple proxy for quality: "if it's more expensive, it must be better". This can produce seemingly irrational behavior, such as purchasing a more expensive good or service (like a bottle of wine or a hotel room) even if no other information besides price is known. In fact, filters on hotel reservation web sites such as `hotels.com` even allow one to specify a minimum price in addition to a maximum price for searching hotel rooms, which seems somewhat strange under usual models for buyers. Why would anyone refuse a cheaper price? Would the web site be better-off by simply increasing the hotel rooms' prices to the minimum specified price? This can be viewed as irrational behavior by users

* AB was supported in part by NSF grants CCF-1800317, CCF-1525971 and CCF-1331175. This work was conducted in part while the author was visiting the Simons Institute for the Theory of Computing and in part while the author was at Carnegie Mellon University.
† YM was supported in part by a grant from the Israel Science Foundation (ISF), a grant from the United States-Israel Binational Science Foundation (BSF), and the Israeli Centers of Research Excellence (I-CORE) program (Center No. 4/11). This work was conducted in part while the author was visiting the Simons Institute for the Theory of Computing.

that might be targeted and explained by behavioral economics. Our goal in this work is to understand, model, and analyze the effect of this phenomenon within the assumptions of rationality.

In this work, we propose a model where buyers have private "desires" for goods (as in the usual auction setting [8]) but their true value also depends on the intrinsic quality q of the good (which might or might not be observable to them). Here, we *define* quality as the cost to the seller to produce that good or service, so a higher-quality good is by definition more expensive to make. Sellers, in our model, have the ability to choose both the price and quality level of the goods they aim to sell. We then analyze how markets of this kind behave. For instance, under what conditions will the price of a good indeed be a reliable signal of its quality, and how sensitive is this signal to the distribution of buyer valuations? What fraction of maximum social welfare can sellers extract in such a model? And do Walrasian equilibria exist and can they be produced by some analog of the usual tatonnement process? More broadly, our aim in this work is to think about (seemingly) irrational behaviors of buyers and to understand how robust various classic economics results are to these deviations from the standard setup for them.

1.1 Our Basic Model

We propose a model where the value that a buyer has for a product (like a slice of pizza) depends on both his personal desire for the product (how hungry he is for pizza) and the quality of the product (how good the pizza is). Our basic model builds on the well known Cobb-Douglas production and valuation function (see [7]). In the Cobb-Douglas function for k quantities, the value is $\prod_{i=1}^{k} x_i^{\alpha_i}$, where $\sum_{i=1}^{k} \alpha_i = 1$. For example, the output of an economy is often modeled as a Cobb-Douglas function over labor and capital, and Cobb-Douglas functions are often used to model utilities over combinations of different goods. We similarly, consider the case that the two quantities are the private valuation and the quality, and specifically focus on the case that the value of a buyer is the geometric mean of the two. Formally, each buyer i has some *intrinsic value* v_{ij} for each product j, and if product j has quality q_j then buyer i's valuation will be $\sqrt{v_{ij}q_j}$. We call this *geometric-mean valuations*, which is a Cobb-Douglas valuation with $\alpha_i = 1/2$. (We also generalize our results to buyers with *α-geometric-mean valuations*, in the spirit of the Cobb-Douglas valuation, which are of the form $v_{ij}^{\alpha} q_j^{1-\alpha}$. Note that for $\alpha = 1$ we get the "standard" case of private valuations.)

We assume that higher quality products are more expensive for the seller to produce. In fact, since it's unclear how to define "units of quality" anyway, we simply *define* quality to be the cost to the seller for making a good of that quality. That is, a slice of pizza that costs twice as much to make is twice the quality by definition.

Finally, we assume quasilinear utilities. So, the *utility* to a buyer of intrinsic value v for a good of quality q that is priced at p is:

$$\sqrt{vq} - p.$$

For example, if a buyer has intrinsic value 10 for a slice of pizza, and a pizzeria is selling slices of quality 2.5 at price 4, then the buyer's utility would be $\sqrt{25} - 4 = 1$ and the seller's profit would be $4 - 2.5 = 1.5$. The overall social welfare would be 2.5, the sum of buyer utility and seller profit.

We consider this model in two classic settings. The first is where there is a single seller and some distribution $\mathcal{F}$ over the nonnegative reals from which buyer "intrinsic values" v are drawn. Assuming that quality *is* observable by buyers, the question then is what price and quality should a seller choose in order to maximize profit, and to what extent will

price and quality be related in a clean way? The second setting is a market with multiple unit-demand buyers and multiple single-item sellers. In this case, the natural questions involve the existence of market equilibria along with natural procedures for finding them.

1.2 Rationale for Cobb-Douglas valuations

There are many ways one can combine an intrinsic value and a quality to derive a valuation, and using the Cobb-Douglas valuations is one way to do it. One benefit of using the Cobb-Douglas valuation is its long history and usefulness in Microeconomics. In this section we will motivate why such Cobb-Douglas valuations are a better fit to our proposed setting than other simple combination rules.

Taking the geometric mean of the intrinsic value v and quality q has several appealing properties. First, if you are making the good for your own consumption (buying ingredients to make pizza at home) your utility $\sqrt{vq} - q$ is maximized at $q = v/4$, giving utility equal to $v/4$. So the maximum social welfare is linearly related to v. Second, we can think of v as the maximum price a buyer would be willing to pay if a good were priced at cost (i.e., if the price is set to q then a buyer would purchase only quality $q \leq v$, since otherwise he will have a negative utility). Third, the function is monotone in both v and q.

The alternative of taking a linear combination of v and q is more problematic. If the buyer's valuation for a good of quality q were, say, $(v + q)/2$, then social welfare and seller profit would be optimized at $q = 0$; on the other hand if it were, say, $v + 2q$ then social welfare and seller profit would be optimized at $q = \infty$. Another alternative might be to combine on a logarithmic scale: setting valuation to be $\ln(vq)$. However, in this case, social welfare and seller profit would always be optimized at $q = 1$, and there would be no correlation at all between price and quality.

1.3 Our Results

We begin by considering a single seller (such as a pizzeria) who can produce one good (pizza) and wishes to determine the quality level q at which to make the good and the price p at which to sell it. The aim of the seller is to maximize profit given some distribution $\mathcal{F}$ over buyer intrinsic values v. Buyers are assumed to be able to observe quality so that the seller cannot just set $q = 0$. What we show is that in such a setting, price will be a surprisingly good signal for quality. In particular, while the optimal price p^* and quality q^* will depend heavily on the buyer distribution $\mathcal{F}$, their *ratio* is fixed: for all $\mathcal{F}$ we have $q^* = p^*/2$ where p^* and q^* are the price and quality which maximize the seller's profit.

Note that this gives a very simple guideline to sellers on how to price their products. For example, for restaurants, pricing the items on the menu is a potentially very involved task. A common rule of thumb which is widely used is to price a dish at three times the cost of its ingredients.[1] This simple rule of thumb is used across many different geographies, cuisines, and price levels. In part, our results justify the ability to give such a simple universal pricing rule.

We also consider a setting where sellers can produce goods of multiple different quality levels. We give a simple strategy for sellers and we show that it is a constant-factor approximation to optimal for revenue.

[1] See, e.g., https://www.forbes.com/sites/priceonomics/2017/04/07/how-much-do-the-ingredients-cost-in-your-favorite-foods/#5be51ac611ed.

Next we turn to the question of market clearing (Walrasian) equilibria. Consider a market with n_b unit-demand buyers and n_g single-good sellers. In the usual model (buyers with private valuations and no quality levels) there always exists a Walrasian equilibrium: an assignment of prices to items and a matching of buyers to items such that (a) every buyer is purchasing his most-desired item at these prices (it is envy-free), and (b) any unpurchased item has price 0. Moreover (c) this matching will maximize social welfare (it is a maximum-weight matching), and (d) it can be found via a natural tatonnement process.[2] We show here that something very similar exists in our model. We begin by proving an analog of (a,b,c). Namely we show that for the social-welfare maximizing assignment of qualities to items and items to buyers, there exist prices that make this assignment an equilibrium.[3] Next we prove an analog of (d), giving a natural process that achieves such an equilibrium.

Finally, we analyze markets that contain both "tourists" and "locals". In these markets, the locals are assumed to know the qualities of various goods or services, whereas the tourists are using price as a signal for quality. For a single good, we analyze how quality will relate to price under optimal seller behavior in such markets and under what conditions the market will become a "tourist trap", setting quality to zero while keeping prices high.

1.4 Related Work

There are many studies in marketing that show how consumers use price as an indicator for quality, and there are empirically-based models that relate perceived quality, price and value, see [12]. There is an ongoing discussion of how a firm can signal an unobservable quality [6]. There is also evidence that the price of products influences the consumer view of their efficacy [9]. (We note that there are also many other works related to quality and pricing from a marketing perspective.)

There are also theoretical works that model the quality-price relationship. The work of [10] studies a market where some of the consumers are price sensitive and some are quality sensitive, and the main issue is that certain consumers might buy products of high price and low quality. The work of [1] shows that in certain instances firms might use an initial high price for a new product, in order to signal to the consumers that the new product is of potentially high quality.

The Cobb-Douglas valuation function is widely studied in Microeconomics (see, [7]), and it has been also applied in various empirical studies. There are empirical studies that relate quality to utility in various domains, such as hospitals [4], mail service [11], soccer coaches [3] and more. In those empirical studies the goal is to find a (best) fit for the Cobb-Douglas parameters, and deduce whether or not the quality plays an important role in influencing the outcome.

2 When price is a signal for quality

Our first result is that in our model of geometric-mean buyers, when there is a single seller who can produce one good (like pizza) at a single quality level, under quite general conditions, price in this setting will in fact be a reliable proxy for quality. This is perhaps surprising since there is no competition.

---

[2] For the existence of a Walresian equilibrium see [5], and for the analysis of tatonnement see [2]. See also [8].

[3] We define the social welfare as the sum of the buyers' and sellers' utilities, or equivalently, the total value to buyers minus the total cost to sellers.

▶ **Theorem 1.** *For any distribution $\mathcal{F}$ over intrinsic values v, and geometric-mean buyer valuations, the seller maximizes its expected profit per potential customer by choosing price p^* and quality q^* such that $q^* = p^*/2$.*

Proof. Let $\mathcal{F}$ be the distribution over the intrinsic values v of people for the given good. Suppose that the seller chooses some quality level q and price p. Then the seller's expected profit on a potential customer is:

$$(p-q) \Pr_{v\sim\mathcal{F}}[\sqrt{vq} \geq p]$$

or equivalently,

$$(p-q) \Pr_{v\sim\mathcal{F}}\left[v \geq p^2/q\right].$$

Define $\tau = p^2/q$. The profit-maximizing threshold τ will of course depend on specifics of the distribution $\mathcal{F}$. However, notice that once τ is fixed, the probability that a random buyer will purchase the good is fixed as well. This means that for any given τ, the optimal values of p and q for the seller are just the values that maximize $p - q$ subject to $p^2/q = \tau$. We can solve for this by taking the derivative of $p - q = p - p^2/\tau$ with respect to p and setting to 0 to get $1 - 2p/\tau = 0$ or $p^* = \tau/2$. This in turn implies that $q^* = (p^*)^2/\tau = \tau/4$. This implies that for any value of τ we have $q^* = p^*/2$. ◀

The above analysis means that for any distribution $\mathcal{F}$, even though a seller who wants to maximize profit will set both price and quality depending on specifics of $\mathcal{F}$, we will always have price equal to twice the quality at the profit-maximizing solution. This means it is reasonable for buyers to use price as a signal for quality, even if they don't know anything about the market.

We can also relate the outcome in our model compared to the standard quasi-linear model (i.e., with no quality, and utility is $v - p$).

▶ **Claim 2.** *Consider a single good setting and let $z^* = \arg\max_z z \Pr[v \geq z]$ the monopolist maximum revenue price for standard quasi-linear valuations. In the geometric-mean valuation model the revenue-maximizing price will be $p^* = z^*/2$ and quality $q^* = z^*/4$. The ratio between the optimal revenue in the geometric-mean valuation and the standard quasi linear valuation is $1/4$.*

Proof. In the geometric mean model the expected profit is $(p-q)Pr[v > p^2/q]$. Since we have shown that the optimal quality q^* equals $p^*/2$, where p^* is the optimal price, the optimal price is $p^* = \arg\max_p (p/2)Pr[v > 2p]$, which is equivalent to $\arg\max_p (2p)Pr[v > 2p]$, and hence $p^* = z^*/2$, $q^* = z^*/4$. The revenue in this case is $\max_p (p/2)Pr[v > 2p] = (1/4)\max_z z\Pr[v \geq z]$, which implies that the expected revenue in the geometric-mean valuation is a $1/4$ of the revenue in the standard quasi-linear valuation. ◀

We can also consider α-geometric-mean buyers, where the value of a buyer with intrinsic value v for a good of quality q is $v^\alpha q^{1-\alpha}$, i.e., the Cobb-Douglas valuation with parameter α. In this case, we have a linear relation between price and quality where now the slope is $1 - \alpha$.

▶ **Theorem 3.** *For any distribution $\mathcal{F}$ over intrinsic values v, and buyers with α geometric mean valuations, where $\alpha \in (0, 1)$, the seller maximizes its expected profit per buyer by choosing price p^* and quality q^*, where $q^* = (1 - \alpha)p^*$.*

Proof. We can rewrite $\Pr_{v\sim\mathcal{F}}[v^\alpha q^{1-\alpha} \geq p]$ as $\Pr_{v\sim\mathcal{F}}[v \geq \tau]$ for $\tau = \frac{p^{1/\alpha}}{q^{(1-\alpha)/\alpha}}$. Fixing τ, we wish to maximize $p - q$. So, plugging in $q = \frac{p^{1/(1-\alpha)}}{\tau^{\alpha/(1-\alpha)}}$ and setting the derivative with respect to p to 0 gives

$$1 - \left(\frac{1}{1-\alpha}\right)\left(\frac{p}{\tau}\right)^{\frac{\alpha}{1-\alpha}} = 0,$$

which solves to $p^* = (1-\alpha)^{\frac{1-\alpha}{\alpha}}\tau$. This in turn implies $q^* = \frac{p^{*1/(1-\alpha)}}{\tau^{\alpha/(1-\alpha)}} = (1-\alpha)^{\frac{1}{\alpha}}\tau$, so $q^* = (1-\alpha)p^*$ as desired. ◀

Note that for $\alpha = 2/3$ we get the famous "factor of three" pricing for restaurants.

2.1 Multiple qualities

Suppose a seller has the ability to produce the good at different quality levels, and in fact can produce it at infinitely many quality levels. Then one strategy the seller can use is to produce goods at *all* quality levels, pricing a good at twice its quality. (A more intuitive interpretation is that buyers can select any quality the buyer wishes and pay the seller twice the quality it selected.)

In that case, a buyer with intrinsic value v will choose the item of quality q maximizing $\sqrt{vq} - 2q$, which solves to $q = v/16$ and $p = v/8$. In this case, the buyer will gain a utility of $v/8$ and the seller will gain a profit of $v/16$.

▶ **Theorem 4.** *Consider a seller which offers goods at all quality levels with price $p = 2q$. For any buyer with geometric mean valuation of intrinsic value v, the utility-maximizing price will be $p = v/8$, quality $q = v/16$ and the seller's revenue $v/16$.*

This is interesting in three respects. First of all, notice that the seller cannot hope to get profit greater than $v/4$ even if the seller knows v, since in that case its optimal strategy is to sell one good of price $v/2$ and quality $v/4$ (this is $\tau = v$ in the previous analysis). So, the seller is within a factor of 4 of the best it could possibly hope for, without requiring any information about the buyers distribution. Secondly, the buyer and seller are splitting the surplus roughly equally, which is interesting. Finally, price is still a signal for quality in this case.

Open Question: Out of all possible pricing functions, is $p(q) = 2q$ optimal for the seller in a minimax sense? More specifically, consider a game where the seller selects a pricing function $p(q)$, the adversary selects an intrinsic value v for the buyer, and then the seller's payoff in the game is the fraction of v that it makes in profit when the buyer chooses quality q and price $p(q)$ to maximize its own utility. I.e., the seller's payoff is $(p(q) - q)/v$ where $q = \arg\max_{q'} \sqrt{vq'} - p(q')$. The above analysis shows that $p(q) = 2q$ guarantees the seller a payoff at least $1/16$ in this game. Additionally, we know the value of this game to the seller is at most $1/4$ since even if it sees v in advance and then gets to best-respond, it cannot make more than $1/4$ of v in profit. It is not hard to show that $p(q) = 2q$ is optimal out of possible *linear* functions, but the open question is whether this is optimal out of all possible pricing functions.

3 Unit Demand buyers

In this section we consider a market with multiple goods and multiple buyers. The buyers are unit-demand, which implies that there is always a singleton (or empty) set which is their

best response. Our main goal is to show that for buyers with geometric mean valuations we maintain the "nice" equilibrium properties that exist in the standard Walrasian equilibrium setting.

In Section 3.1 we show that for buyers with geometric mean valuations, there always exist prices and qualities that guarantee (a) that each buyer is allocated a best response set, (b) that unsold items have price 0, and (c) that the social welfare is maximized. (Note that for geometric mean valuations, satisfying only (a) and (b) is trivial because we can can set all the prices and qualities to zero, and not allocate any good to any buyer.)

The proof in Section 3.1 does not provide a natural dynamics. In Section 3.2 we improve on this by defining a rather natural two stage dynamics, which we show reaches the desired equilibrium. In this dynamics the buyers compete for the right to receive a good and set its quality.

3.1 Market clearing prices and qualities

Consider n_b unit-demand buyers with geometric-mean valuations and n_g goods. Formally, each buyer i has an intrinsic value $v_{i,j}$ for good j. Given prices p and qualities q, the utility of buyer i for a subset S of goods is,

$$u_i(S) = \max_{j \in S} \sqrt{v_{i,j} q_j} - \sum_{j \in S} p_j.$$

Note that buyer i's utility is always maximized on a set S_i of size at most one.

We would like to consider market clearing prices and qualities. That is, prices and qualities for which each buyer receives a utility-maximizing set S_i, unsold goods have price and quality zero, and overall social welfare is maximized over possible allocations $\{S_i\}$ and qualities $\{q_j\}$. Formally, for each item j either $p_j = q_j = 0$ or $j \in S_i$ for exactly one buyer i, and moreover the social welfare of the allocation is as high as possible where social welfare is defined as

$$\sum_{i=1}^{n_b} u_i(S_i) - \sum_{j=1}^{n_g} q_j = \sum_{i=1}^{n_g} \max_{j \in S_i} \sqrt{v_{i,j} q_j} - \sum_{j=1}^{n_g} q_j \ ,$$

and the maximization is both over the allocations S_i and the qualities q_j.

One useful fact to note is that a necessary condition for maximizing social welfare is that if $S_i = \{j\}$ then we must have $q_j = v_{i,j}/4$. This implies that the maximum social welfare is well defined as a function of the intrinsic values.

▶ **Theorem 5.** *For any set of n_b unit-demand buyers, with geometric-mean valuations, there are prices p and qualities q which clear the market with an allocation that maximizes social welfare.*

Proof. Consider any matching between buyers and goods. If buyer i is matched to good j, then social welfare is maximized at $q_j = v_{i,j}/4$, and the contribution to social welfare of this pair is $v_{i,j}/4$.

This implies that for any matching, the maximum welfare of that matching is exactly the weight of the matching divided by 4. Therefore, to solve for the maximum welfare solution, we can find the maximum weighted matching and then set qualities equal to the buyer's value divided by 4.

Now, we need to define prices that make this an equilibrium. We will show a reduction to a standard unit-demand valuations and solve for prices in that market, i.e., a Walresian

Algorithm 1: Two phase dynamics

phase 1: Tatonnement process:
  Each buyer competes for goods (where the valuation of buyer i for good j is $v_{i,j}/4$)
outcome: each good j is allocated to at most one buyer $b(j)$ with a price p_j.
phase 2: Setting the qualities:
  For each good j, buyer $b(j)$ sets the quality $q_j = v_{i,j}/4$, pays $p_j + q_j = p_j + v_{i,j}/4$ and receives good j with quality q_j.

equilibrium. Consider the maximum matching between buyers and goods, ignoring qualities. We will define for each good j a quality q_j. For any unallocated item we set the quality to 0 (which will imply later price 0). For good j, which the maximum matching allocated to buyer i we set $q_j = v_{i,j}/4$. After we fix the qualities, we define n_b standard unit demand buyers, where the valuation of buyer k for good j is $\widehat{v}_{k,j} = \sqrt{v_{k,j}q_j}$. We now have a market with unit demand buyers, and therefore there exists a Walresian equilibrium with price $\widehat{p}_j$ for each good j (see, [5]).

Therefore, in our unit demand buyers with geometric mean valuation, the market clearing prices are $\langle \widehat{p}_1, \ldots, \widehat{p}_{n_g} \rangle$ and qualities $\langle q_1, \ldots, q_{n_g} \rangle$ ◀

The above proof does not give a "natural" process to determine the prices, but shows that market clearing prices and qualities always exist for unit demand buyers. Below we modify the above construction to produce a more natural tatonnement-like process.

3.2 A natural dynamics process

We consider the following process to set prices and qualities for each good. In this process, buyers are bidding for the "right to control" each good. That is, prices on each good begin at zero and are raised as in the usual tatonnement process, where the quality of each good will later be *determined and paid for by the winning buyer for it.* In particular, if a buyer wins a good j for a price p_j, she can then afterwards set the quality of the good to any desired q_j and pay a total of $p_j + q_j$.

One way to think of this is like an auction by oil companies for drilling rights on land. Once a company i pays p_j for the rights to drill on land j, it then can decide the amount q_j that it will invest to drill, and it will then receive revenue of $\sqrt{v_{i,j}q_j}$, having paid a total of $p_j + q_j$.

Note that once buyer i purchases (the right to control) good j it will then set a quality $q_j = v_{i,j}/4$ in order to maximize its own utility, which will be $\sqrt{v_{i,j}q_j} - q_j - p_j = v_{i,j}/4 - p_j$. Therefore, when buyer i is considering whether it prefers to pay p_j for item j or $p_{j'}$ for item j', it is comparing $v_{i,j}/4 - p_j$ to $v_{i,j'}/4 - p_{j'}$ just as in the usual Walrasian market setting except the values v have been divided by 4.

So, by allowing buyers to bid on the "right to control" each good, we now have a Walrasian market, where the fundamental valuation of buyer i for good j is $v_{i,j}/4$. This will result in the same outcome as having valuations $v_{i,j}$ (with just the clearing prices a factor of 4 lower). This means it is an equilibrium and social welfare maximizing allocation in our setting as well.

Thus we have the following theorem.

▶ **Theorem 6.** *For unit-demand buyers with geometric mean valuation, the two-phase dynamics (of Algorithm 1) converges to market clearing prices which maximize the social*

welfare.

4 Two populations: Local and Tourists

In this section we go back to the setting of a single good considered in Section 2, but now assume that the populations of buyers is composed of two sub-populations. One sub-population, which we call *locals*, are aware of the quality of the good. The other sub-population, which we call *tourists*, are unaware of the quality of the good. The tourist sub-population uses the price as a proxy for quality, and uses the naive assumption that the quality is half the price (which would be the case under optimal sellers in a market of only locals). We assume the seller is aware of the fraction of tourists and locals, but has to post a single price for both populations. Our goal is to investigate the effect of tourists on the quality and prices that the seller chooses. For example, we would expect to observe effects similar to tourist traps, where the quality is low and the price is high.

Formally, assume that the population is λ fraction locals and $1-\lambda$ fraction tourists. The tourists have a "one time" experience. They assume that the quality of the good is $q = p/2$. (We assume that the behavior of the tourists is non-strategic in this aspect. We leave for future research considering the case that the tourists are aware of λ , the fraction of locals, and consider an equilibrium of price and quality. The main challenge would be to have a model where the price will not completely reveal the quality.)

We will start with the two extreme cases, $\lambda = 1$ and $\lambda = 0$. The first case is *no tourists*, i.e., $\lambda = 1$. We saw that in this case we have $q = p/2$ and the seller selects a price p that maximizes

$$(p-q)\Pr[\sqrt{vq} \geq p] = (p/2)\Pr[v \geq 2p].$$

In the second case we have *only tourists*, i.e., $\lambda = 0$, then we have the seller selecting a price that maximizes

$$p\Pr[\sqrt{vp/2} \geq p] = p\Pr[v \geq 2p].$$

Clearly in both cases we have the same optimal price p^*, but a different seller quality and revenue. In the no tourists case the quality is $p^*/2$ and the revenue per good is $p^*/2$, while in the only tourists case the quality is 0 and the revenue per good is p^*.

4.1 Price at intermediate values of λ

We saw above that for any $\mathcal{F}$, the optimal price at $\lambda = 1$ equals the optimal price at $\lambda = 0$. This brings up the natural question: is that also true for intermediate values of λ?

The answer to this question is "not necessarily". In particular, consider a distribution $\mathcal{F}$ on intrinsic values v that is uniform over $[0, 1]$. In the case of $\lambda \in \{0, 1\}$, profit is optimized when we maximize $p\Pr[v \geq 2p] = p(1-2p)$, which occurs at $p = 1/4$ (with $q = 1/8$ when $\lambda = 1$ and $q = 0$ when $\lambda = 0$). However, at $\lambda = 0.5$, a calculation shows that profit is maximized at $p = 1/(3+\sqrt{3}) \approx 1/4.7$, and $q = p/(1+\sqrt{3})$. For example, with $\lambda = 0.5$, if the seller sets $p = 1/4$ then the maximum profit she can make is $1/16 = 0.0625$ (which occurs both at $q = 0$ and $q = 1/8$). But at the optimal p and q, she gets a (slightly) higher profit of 0.067.

Another natural qualitative question is: what happens to quality when the fraction of locals is small? When the fraction of locals is zero, then clearly the seller's optimal quality is zero. The question is whether or not any strictly positive fraction of locals is sufficient to drive the quality away from zero.

In the next section we show that the answer to this question is distribution-dependent. For many distributions the quality will remain zero, while for some distributions the quality will increase from zero.

4.2 Pareto distribution for intrinsic values

We consider a Pareto distribution for the intrinsic values. Recall that a Pareto distribution has two parameters, $\beta > 0$ and $x_{\min} > 0$. We will set $x_{\min} = 1$, so we have a single parameter. The density of the distribution is $f(x) = \beta/x^{1+\beta}$ and $\Pr[v \geq x] = (1/x)^{\beta}$.

We can now write the revenue R as a function of the price p, quality q and parameter β, i.e.,

$$R(p, q, \beta) = \lambda(p - q)(\frac{q}{p^2})^{\beta} + (1 - \lambda)(p - q)(\frac{1}{2p})^{\beta}$$

Clearly maximizing the revenue implies that $q \in [0, p]$. Consider the derivative of the revenue with respect to the quality,

$$\frac{\partial}{\partial q}R = -\lambda(\frac{q}{p^2})^{\beta} + \lambda(p - q)\frac{\beta}{p^2}(\frac{q}{p^2})^{\beta-1} - (1 - \lambda)(\frac{1}{2p})^{\beta}$$

For $\beta = 1$ we have

$$\frac{\partial}{\partial q}R = \frac{\lambda}{p} - \frac{2\lambda q}{p^2} - \frac{1 - \lambda}{2p}$$

We get that the optimal q^* is

$$q^* = \max\{0, \frac{p(3\lambda - 1)}{4\lambda}\}$$

which for $\lambda \leq 1/3$ implies that $q^* \leq 0$ and hence the optimal q for $\lambda \leq 1/3$ is $q^* = 0$.
For $\beta = 1/2$ we have

$$\frac{\partial}{\partial q}R = \frac{\lambda}{2\sqrt{q}} - \frac{3\lambda\sqrt{q}}{2p} - \frac{1 - \lambda}{\sqrt{2p}}$$

In this case we show that even for $\lambda \approx 0$ we will have $q^* > 0$. The optimal quality q^* is

$$\begin{aligned}\sqrt{q^*} &= \frac{-\sqrt{2}(1 - \lambda)/\sqrt{p} \pm \sqrt{2(1 - \lambda)^2/p + 12\lambda^2/p}}{6\lambda/p} \\ &= \frac{\sqrt{2p}}{6\lambda}(\sqrt{(1 - \lambda)^2 + 6\lambda^2} - (1 - \lambda)) \\ &= \frac{\lambda\sqrt{2p}}{\sqrt{(1 - \lambda)^2 + 6\lambda^2} + (1 - \lambda)}\end{aligned}$$

and for $\lambda \approx 0$ we have $q^* \approx \lambda^2 p/2 > 0$.

It is clear that the main issue is whether the derivative at $q = 0$ is infinite (as is the case for $\beta < 1$) or finite (as is the for $\beta \geq 1$). In the former case we will have $q^* > 0$ when $\lambda > 0$ and in the latter case we will have $q^* = 0$ for small values of λ.

▶ **Theorem 7.** *For Pareto distribution over intrinsic valuation: (1) for any $\beta \geq 1$ there is a constant $\gamma_\beta > 0$ such that for $\lambda \in [0, \gamma_\beta]$ the optimal seller quality is $q^* = 0$, and (2) for $0 < \beta < 1$, for any $\lambda > 0$ the optimal seller quality is $q^* > 0$.*

Proof. Consider the case that $\beta \geq 1$ we have

$$\frac{\partial}{\partial q}R = \lambda(\frac{q}{p^2})^{\beta-1}((p-q)\frac{\beta}{p^2} - \frac{q}{p^2}) - (1-\lambda)(\frac{1}{2p})^{\beta}$$

Clearly,

$$(p-q)\frac{\beta}{p^2} - \frac{q}{p^2} \leq \frac{\beta}{p}$$

Therefore,

$$\frac{\partial}{\partial q}R \leq \lambda(\frac{q}{p^2})^{\beta-1}\frac{\beta}{p} - (1-\lambda)(\frac{1}{2p})^{\beta}$$

Since $q < p$ (if $q = p$ then $R = 0$) we have

$$\frac{\partial}{\partial q}R < (\frac{1}{p})^{\beta}(\lambda\beta - (1-\lambda)(\frac{1}{2})^{\beta})$$

Therefore, for $\lambda < \frac{1}{1+\beta 2^{\beta}} = \gamma_{\beta}$ we have $\frac{\partial}{\partial q}R < 0$ and hence $q^* = 0$.

Consider the case that $0 < \beta < 1$.

$$\frac{\partial}{\partial q}R = \lambda(p-q)\frac{\beta}{p^2}\frac{p^2}{q}(\frac{q}{p^2})^{\beta} - \lambda(\frac{q}{p^2})^{\beta} - (1-\lambda)(\frac{1}{2p})^{\beta}$$

If we have $q^* > p/2 > 0$ we are done. Otherwise we have

$$\frac{\partial}{\partial q}R > \lambda(p/2)\frac{\beta}{q}(\frac{q}{p^2})^{\beta} - (\frac{1}{2p})^{\beta}(\lambda + (1-\lambda)) = \lambda(p/2)\frac{\beta}{q}(\frac{q}{p^2})^{\beta} - (\frac{1}{2p})^{\beta}$$

For $q < (\lambda\beta)^{1/(1-\beta)}p/2$ we have that $\frac{\partial}{\partial q}R > 0$, and hence the optimal quality q^* is at least $(\lambda\beta)^{1/(1-\beta)}p/2 > 0$. ◀

References

1. Kyle Bagwell and Michael H. Riordan. High and declining prices signal product quality. *The American Economic Review*, 81(1):224–239, 1991.
2. Gabrielle Demange, David Gale, and Marilda Sotomayor. Multi-item auctions. *Journal of Political Economy*, 94(4):863–872, 1986.
3. Bernd Fricka and Robert Simmons. The impact of managerial quality on organizational performance: Evidence from german soccer. *Managarial and Decision Economics*, 29:593–600, 2008.
4. Paul J. Gertler and Donald M. Waldman. Quality-adjusted cost functions and policy evaluation in the nursing home industry. *Journal of Political Economy*, 100(6):1232–1256, 1992.
5. Faruk Gul and Ennio Stacchetti. Walrasian equilibrium with gross substitutes. *Journal of Economic Theory*, 87(1):95–124, 1999.
6. Amna Kirmani and Akshay R. Rao. No pain, no gain: A critical review of the literature on signaling unobservable product quality. *Journal of Marketing*, 64(2):66–79, 2000.
7. Andreu Mas-Colell, Michael D. Whinston, and Jerry R. Green. *Microeconomic Theory*. Oxford University Press, 1995.
8. N. Nisan, T. Roughgarden, E. Tardos, and V. V. Vazirani. *Algorithmic Game Theory*. Cambridge University Press, 2007.

9 Ariely Dan Shiv Baba, Carmon Ziv. Placebo effects of marketing actions: Consumers may get what they pay for. *Journal of Marketing Research*, 42:383–393, 2005.

10 Steven M. Shugan. Price-quality relationships. *Advances in Consumer Research*, 11:627–632, 1984.

11 Surendra Rajiv Tridas Mukhopadhyay and Kannan Srinivasan. Information technology impact on process output and quality. *Management Science*, 43(12):1645–1659, 1997.

12 Valarie A. Zeithaml. Consumer perceptions of price, quality, and value: A means-end model and synthesis of evidence. *Journal of Marketing*, 52(3):2–22, 1988.

Pseudo-Deterministic Proofs

Shafi Goldwasser[1], Ofer Grossman[2], and Dhiraj Holden[3]

1 **MIT, Cambridge MA, USA**
`shafi@theory.csail.mit.edu`
2 **MIT, Cambridge MA, USA**
`ofer.grossman@gmail.com`
3 **MIT, Cambridge MA, USA**
`dholden@mit.edu`

Abstract

We introduce *pseudo-deterministic interactive proofs* (psdIP): interactive proof systems for search problems where the verifier is guaranteed with high probability to output the same output on different executions. As in the case with classical interactive proofs, the verifier is a probabilistic polynomial time algorithm interacting with an untrusted powerful prover.

We view pseudo-deterministic interactive proofs as an extension of the study of pseudo-deterministic randomized polynomial time algorithms: the goal of the latter is to *find* canonical solutions to search problems whereas the goal of the former is to *prove* that a solution to a search problem is canonical to a probabilistic polynomial time verifier. Alternatively, one may think of the powerful prover as aiding the probabilistic polynomial time verifier to find canonical solutions to search problems, with high probability over the randomness of the verifier. The challenge is that pseudo-determinism should hold not only with respect to the randomness, but also with respect to the prover: a malicious prover should not be able to cause the verifier to output a solution other than the unique canonical one.

The $IP = PSPACE$ characterization implies that psdIP $= IP$. The challenge is to find constant round pseudo-deterministic interactive proofs for hard search problems. We show a constant round pseudo-deterministic interactive proof for the graph isomorphism problem: on any input pair of isomorphic graphs (G_0, G_1), there exist a unique isomorphism ϕ from G_0 to G_1 (although many isomorphism many exist) which will be output by the verifier with high probability, regardless of any dishonest prover strategy. In contrast, we show that it is unlikely that psdIP proofs with constant rounds exist for NP-complete problems by showing that if any NP-complete problem has a constant round psdIP protocol, then the polynomial hierarchy collapses.

1998 ACM Subject Classification F.1.1. Models of Computation (Probabilistic Algorithms)

Keywords and phrases Pseudo-Determinism, Interactive Proofs

Digital Object Identifier 10.4230/LIPIcs.ITCS.2018.17

1 Introduction

In [6], Gat and Goldwasser initiated the study of probabilistic (polynomial-time) search algorithms that, with high probability, output the same solution on different executions. That is, for all inputs x, the randomized algorithm A satisfies $Pr_{r_1,r_2}(A(x,r_1) = A(x,r_2)) \geq 1 - 1/poly(n)$.

Another way of viewing such algorithms is that for a fixed binary relation R, for every x the algorithm associates a canonical solution $s(x)$ satisfying $(x, s(x)) \in R$, and on input x the algorithm outputs $s(x)$ with overwhelmingly high probability. Algorithms that satisfy this

condition are called *pseudo-deterministic*, because they essentially offer the same functionality as deterministic algorithms; that is, they produce a canonical output for each possible input (except with small error probability)[1]. In contrast, arbitrary probabilistic algorithms that solve search problems may output different solutions when presented with the same input (but using different internal coin tosses); that is, on input x, the output may arbitrarily distributed among all valid solutions for x (e.g. it may be uniformly distributed).

Several pseudo-deterministic algorithms have been found which improve (sometimes significantly) on the corresponding best known deterministic algorithm. This is the case for finding quadratic non-residues modulo primes, generators for certain cyclic groups, non-zeros of multi-variate polynomials, matchings in bipartite graphs in RNC, and sub-linear algorithms for several problems [8, 10, 6, 13]. For other problems, such as finding unique primes of a given length, pseudo-deterministic algorithms remain elusive (for the case of primes, it has been shown that there exists a subexponential time pseudo-deterministic algorithm which works on infinitely many input sizes [22]).

In this work we extend the study of pseudo-determinism in the context of probabilistic algorithms to the context of interactive proofs and non-determinism. We view pseudo-deterministic interactive proofs as a natural extension of pseudo-deterministic randomized polynomial time algorithms: the goal of the latter is to *find* canonical solutions to search problems whereas the goal of the former is to *prove* that a solution to a search problem is canonical to a probabilistic polynomial time verifier. This naturally models the cryptographic setting when an authority generates system-wide parameters (e.g. an elliptic curve for all to use or a generator of a finite group) and it must prove that the parameters were chosen properly.

1.1 Our Contribution

Consider the search problem of finding a large clique in a graph. A nondeterministic efficient algorithm for this problem exists: simply guess a set of vertices C, confirm in polynomial time that the set of vertices forms a clique, and either output C or reject if C is not a clique. Interestingly, in addition to being nondeterministic, there is another feature of this algorithm; on the same input graph there may be many possible solutions to the search problem and any one of them may be produced as output. Namely, on different executions of the algorithm, on the same input graph G, one execution may guess clique C and another execution may guess clique $C' \neq C$, and both are valid accepting executions.

A natural question is whether for each graph with a large clique, there exists a unique canonical large clique C which can be verified by a polynomial time verifier: that is, can the verifier V be convinced that the clique C is the canonical one for the input graph? Natural candidates which come to mind, such as being the lexicographically smallest large clique, are not known to be verifiable in polynomial time (but seem to require the power of Σ_2 computation).

In this paper, we consider this question in the setting of interactive proofs and ask whether the interactive proof mechanism enables provers to convince a probabilistic verifier of the "uniqueness of their answer" with high probability.

[1] In fact, by amplifying the success probability, one can ensure that as black boxes, pseudo-deterministic algorithms are indistinguishable from deterministic algorithms by a polynomial time machine.

Pseudo-deterministic Interactive Proofs: We define *pseudo-deterministic interactive proofs* for a search problem R (consisting of pairs $(instance, solution)$) with associated function s as a pair of interacting algorithms: a probabilistic polynomial time verifier and a computationally unbounded prover which on a common input instance x engage in rounds of interaction at the end of which with high probability the verifier output a canonical solution $y = s(x)$ for x if any solution exists and otherwise rejects x. Analogously to the case of completeness in interactive proofs for languages, we require *Canonical Completeness*: that for every input x, there exists an honest prover which can send the correct solution $s(x)$ to the verifier when one exists. Analogously to the case of soundness, we require *Canonical Soundness*: no dishonest prover can cause the verifier to output a solution other than $s(x)$ (the canonical one) (except with very low probability).

One may think of the powerful prover as aiding the probabilistic polynomial time verifier to find canonical solutions to search problems, with high probability over the randomness of the verifier. The challenge is that pseudo-determinism should hold not only with respect to the randomness, but also with respect to the prover: a malicious prover should not be able to cause the verifier to output a solution other than the canonical unique one. In addition to the intrinsic complexity theoretic interest in this problem, *consistency* or *predictability* of different executions on the same input are natural requirements from protocols.

We define *pseudo-deterministic IP* (psdIP) to be the class of search problems R (relation on inputs and solutions) for which there exists a pseudo-deterministic polynomial round interactive proof for R.

Theorem: For any problem L in NP, there is a pseudo-deterministic polynomial round interactive proof for the search problem R consisting of all pairs (x, w) where $x \in L$ and w is a witness for x.

One can prove the above theorem by noting that finding the lexicographically first witness w for x is a problem in PSPACE. Then, since IP = PSPACE [24], we know an interactive proof for finding the lexicographically first witness w exists. More formally we have:

Proof: Let us consider the function $f(x)$ which outputs the lexicographically first witness that $x \in L$ if $x \in L$ or $\perp$ otherwise. It is easy to see that determining whether $f(x) = y$ is in PSPACE. As a result, there is an polynomial-round IP protocol to determine whether $f(x) = y$. Then, the psdIP protocol is as follows; the prover gives the verifier y and then they run the protocol, and the verifier accepts and outputs y if the protocol accepts. This satisfies the conditions for pseudo-determinism because of the completeness and soundness properties of the IP protocol.

In light of the above, we ask: do **constant-round** *pseudo-deterministic interactive proofs exist for hard problems in* NP *for which many witnesses exist?* We let psdAM refer to those pseudo-deterministic interactive proofs in which a constant number of rounds is used. [2].

Graph Isomorphism is in pseudo-deterministic AM: Theorem 7: *There exists a pseudo-deterministic constant-round Arthur-Merlin protocol for finding an isomorphism between two given graphs.*

Recall that the first protocol showing graph non-isomorphism is in constant round IP was shown by [9] and later shown to be possible using public coins via the general transformation

[2] We note that historically, the class AM referred to protocols in which the verifiers' messages consisted of his coin tosses, namely public-coin protocols. In this work, we use AM to refer to constant round interactive proofs

of private to public coins [11]. Our algorithm finds a unique isomorphism by producing the lexicographically first isomorphism. In order to prove that a particular isomorphism between input graph pairs is lexicographically smallest, the prover will prove in a sequence of sub-protocols to the verifier that a sequence of graphs suitably defined are non-isomorphic. In an alternative construction, we exhibit an interactive protocol that computes the automorphism group of a graph in a verifiable fashion.

SAT is not in pseudo-deterministic AM: Theorem 9: *if any* NP*-complete problem has a a pseudo-deterministic constant round* AM *protocol, then,* NP $\subseteq$ coNP/*poly and the polynomial hierarchy collapses to the third level, showing that it is unlikely that NP complete problems have pseudo-deterministic constant round* AM *protocols.*

This result extends the work of [16] which shows that if there are polynomial time unique verifiable proofs for SAT, then the polynomial hierarchy collapses. Essentially, their result held for deterministic interactive proofs (i.e., NP), and we extend their result to probabilistic interactive proofs with constant number of rounds (i.e., AM).

Every problem in search-BPP is in subexponential-time pseudo-deterministic MA: Theorem 13: *For every problem in search-*BPP*, there exists a pseudo-deterministic* MA *protocol where the verifier takes subexponential time on infinitely many input lengths.*

The idea of the result is to use known circuit lower bounds to get pseudo-deterministic subexponential time MA protocols for problems in search-BPP for infinitely many input lengths. We remark that recently Oliveira and Santhanam [22] showed a subexponential time pseudo-deterministic algorithm for infinitely many input lengths for all properties which have inverse polynomial density and are testable in probabilistic polynomial time. (An example of such a property is the property of being prime, as the set of primes has polynomial density.) In their construction, the condition of high density is required in order for the property to intersect with their subexponential-size hitting set. (Subsequent work in [17] also drops this requirement but only results in an average-case pseudo-deterministic algorithm.) In the case of MA, unconditional circuit lower bounds for MA with a verifier which runs in exponential time have been shown by Miltersen et al [20], which allows us to no longer require inverse polynomial density. Hence, we can obtain a pseudo-deterministic MA algorithm from circuit lower bounds. Thus, compared to [22], our result shows a pseudo-derandomization (for a subexponential verifier and infinitely many input sizes n) for all problems in search-BPP (and not just those with high density), but requires a prover.

Pseudo-deterministic NL equals search-NL: Theorem 15: *For every search problem in search-*NL*, there exists a pseudo-deterministic* NL *protocol.*

We define *pseudo-deterministic NL* to be the class of search problems R (a relation on inputs and solutions) for which there exists log-space non-deterministic algorithm M (Turing machines) such that for every input x, there exists a unique $s(x)$ such that $R(x, s(x)) = 1$ and $M(x)$ outputs $s(x)$ or rejects x. Namely, there are no two accepting paths for input x that result in different outputs.

To prove the above theorem, we look at the problem of directed connectivity (that is, given a directed graph G with two vertices s and t, we find a path from s to t), and we show that it is possible to find the lexicographically first path of shortest length in NL. To do so, we first find the length d of the shortest path, which can be done in NL. Then, we find the lexicographically first outneighbor u of s such that there is a path of length $d-1$ from u to t. This can be done by going in order over all outneighbors of s, and for each of them checking

if there is a path of length $d-1$ to t (if there is not such a path, that can be demonstrated in NL since NL = coNL [18, 25]). By recursively applying this protocol to find a path from u to t, we end up obtaining the lexicographically first path of shortest length, which is unique.

Structural Results: We show a few structural results regarding pseudo-deterministic interactive proofs In Section 7. Specifically, we show that psdAM equals to the class search$-\mathrm{P}^{\mathrm{promise}-(\mathrm{AM}\cap\mathrm{coAM})}$, where for valid inputs x, all queries to the oracle must be in the promise. We show similar results in the case of pseudo-deterministic MA and pseudo-deterministic NP.

1.2 Other Related Work

In their seminal paper on NP with unique solutions, Valiant and Vazirani asked the following question: is the inherent intractability of NP-complete problems caused by the fact that NP-complete problems have many solutions? They show this is not the case by exhibiting a problem – SAT with unique solutions – which is NP-hard under randomized reductions. They then showed how their result enables to show the NP-hardness under randomized reductions for a few related problems such as parity-SAT. We point out that our question is different. We are not restricting our study to problems (e.g. satisfiable formulas) with unique solutions. Rather, we consider hard problems for which there may be exponentially many solutions, and ask if one can focus on one of them and verify it in polynomial time. In the language of satisfiability, ϕ can be any satisfiable formula with exponentially many satisfying assignments; set $s(\phi)$ to be a unique valued function which outputs a satisfying assignment for ϕ. We study whether there exists an s which can be efficiently computed, or which has an efficient interactive proof.

The question of computing canonical labellings of graphs was considered by Babai and Luks [3] in the early eighties. Clearly graph isomorphism is polynomial time reducible to computing canonical labellings of graphs (compute the canonical labeling for your graphs and compare), however it is unknown whether the two problems are equivalent (although finding canonical labellings in polynomial time seems to be known for all classes of graphs for which isomorphism can be computed in polynomial time). The problem of computing a set of generators (of size $O(\log n)$) of the automorphism group of a graph G was shown by Mathon [19] (among other results) to be polynomial-time reducible to the problem of computing the isomorphism of a graph. We use this in our proof that graph isomorphism is in psdAM.

A line of work on search vs decision and hierarchy collapses, some in the flavor of our result of Section 4, have appeared in [16, 15, 14, 4].

Finally, we mention that recently another notion of uniqueness has been studied in the context of interactive proofs by Reingold et al [23], called *unambiguous interactive proofs* where the prover has a unique successful strategy. This again differs from pseudo-deterministic interactive proofs, in that we don't assume (nor guarantee) a unique strategy by the successful prover, we only require that the prover proves that the solution (or witness) the verifier receives is unique (with high probability).

1.3 Subsequent Work

In [17], inspired by this work, Holden shows that for every BPP search problem there exists an algorithm A which for infinitely many input lengths n and for every polynomial-time samplable distribution over inputs of length n runs in subexponential time and produces

a unique answer with high probability on inputs drawn from the distribution and over A's random coins.

[17] expands on the work of Oliveira and Santhanam [22] in several ways. Whereas the latter give a pseudo-deterministic algorithm for estimating the acceptance probability of a circuit on inputs of a given length, the former applies to general search-BPP problems, where the input is a string of a given length over some alphabet and algorithm's A goal is to output a solution that satisfies a BPP testable relation with the input string. Holden [17] shows that for infinitely many input lengths, average-case (over the input distribution) pseudo-deterministic algorithms are possible for problems in search-BPP.

2 Definitions of Pseudo-deterministic Interactive Proofs

In this section, we define pseudo-determinism in the context of nondeterminism and interactive proofs. We begin by defining a search problem.

▶ **Definition 1** (Search Problem)**.** A *search problem* is a relation R consisting of pairs (x, y). We define L_R to be the set of x's such that there exists a y satisfying $(x, y) \in R$. An algorithm solving the search problem is an algorithm that, when given $x \in L_R$, finds a y such that $(x, y) \in R$. When L_R contains all strings, we say that R is a *total* search problem. Otherwise, we say R is a *promise* search problem.

We now define pseudo-determinism in the context of interactive proofs for search problems. Intuitively speaking, we say that an interactive proof is pseudo-deterministic if an honest prover causes the verifier to output the same unique solution with high probability (canonical completeness), and dishonest provers can only cause the verifier to output either the unique solution or $\perp$ with high probability (canonical soundness). In other words, dishonest provers cannot cause the verifier to output an answer which is not the unique answer. Additionally, we have the condition that for an input x with no solutions, for all provers the verifier will output $\perp$ with high probability (standard soundness). We note that we use psdIP, psdAM, psdNP, psdMA, and so on, to refer to a class of promise problems, unless otherwise stated.

▶ **Definition 2** (Pseudo-deterministic IP)**.** A search problem R is in *pseudo-deterministic* IP (often denoted psdIP) if there exists a function s mapping inputs to the search problem to solutions (i.e., all $x \in L_R$ satisfy $(x, s(x)) \in R$), and there is an interactive protocol between a probabilistic polynomial time verifier algorithm V and a prover (unbounded algorithm) P such that for every $x \in L_R$:

1. (Canonical Completeness) There exists a P such that $\Pr_r[(P, V)(x, r) = s(x)] \geq \frac{2}{3}$. (We use $(P, V)(x, r)$ to denote the output of the verifier V when interacting with prover P on input x using randomness r).
2. (Canonical Soundness) For all P', $\Pr_r[(P', V)(x, r) = s(x) \text{ or } \perp] \geq \frac{2}{3}$.

And (Standard Soundness) for every $x \notin L_R$, for all provers P', $\Pr_r[(P', V)(x, r) \neq \perp] \leq \frac{1}{3}$.

One can similarly define pseudo-deterministic MA, and pseudo-deterministic AM, where MA is a 1-round protocol, and AM is a 2-round protocol. One can show that any constant-round interactive protocol can be reduced to a 2-round interactive protocol [2]. Hence, the definition of pseudo-deterministic AM captures the set of all search problems solvable in a constant number of rounds of interaction.

Historical Note: Historically, AM referred to public coin protocols, whereas IP referred to private coin protocols. In this work, we use AM to refer to constant round protocols, and IP

to refer to polynomial round protocols (unless explicitly stated otherwise). By the result of [11], we know that when the prover is all-powerful, a private coin protocol can be simulated using private coins, so in this setting the distinction between private and public coins does not matter.

▶ **Definition 3** (Pseudo-deterministic AM)**.** A search problem R is in *pseudo-deterministic* AM (often denoted psdAM) if there exists a function s where all $x \in L_R$ satisfy $(x, s(x)) \in R$, a probabilistic polynomial time verifier algorithm V, and polynomials p and q, such that for every $x \in L_R$:

1. (Canonical Completeness) $\Pr_{r\in\{0,1\}^{p(n)}}(\exists z \in \{0,1\}^{q(n)}\, V(x,r,z) = s(x)) \geq \frac{2}{3}$
2. (Canonical Soundness) $\Pr_{r\in\{0,1\}^{p(n)}}(\forall z \in \{0,1\}^{q(n)}\, V(x,r,z) \in \{s(x), \perp\}) \geq \frac{2}{3}$.

And (Standard Soundness) for every $x \notin L_R$, we have $\Pr_{r\in\{0,1\}^{p(n)}}(\forall z \in \{0,1\}^{q(n)}\, V(x,r,z) = \{\perp\}) \geq \frac{2}{3}$.

▶ **Definition 4** (Pseudo-deterministic MA)**.** A search problem R is in *pseudo-deterministic* MA (often denoted psdMA) if there exists a function s where all $x \in L_R$ satisfy $(x, s(x)) \in R$ and $|s(x)| \leq poly(x)$, a probabilistic polynomial time verifier V such that for every $x \in L_R$[3]:

1. (Canonical Completeness) There exists a message M of polynomial size such that $\Pr_r[V(x, M, r) = s(x)] \geq \frac{2}{3}$.
2. (Canonical Soundness) For all M', $\Pr_r[V(x, M', r) = s(x) \text{ or } \perp] > \frac{2}{3}$.

And (Standard Soundness) for every $x \notin L_R$, for all M', $\Pr_r[V(x, M', r) \neq \perp] \leq \frac{1}{3}$.

Pseudo-determinism can similarly be defined in the context of NP (which can be viewed as a specific case of an interactive proof):

▶ **Definition 5** (Pseudo-deterministic NP)**.** A search problem R is in *pseudo-deterministic* NP (often denoted psdNP) if there exists a function s where all $x \in L_R$ satisfy $(x, s(x)) \in R$ and $|s(x)| \leq poly(x)$, and there is a deterministic polynomial time verifier V such that for every $x \in L_R$:

1. There exists a message M of polynomial size such that $V(x, M) = s(x)$.
2. For all M', $V(x, M') = s(x)$ or $V(x, M') = \perp$.

And for every $x \notin L_R$, for all M', we have $V(x, M') = \perp$.

A similar definition for pseudo-deterministic NL follows naturally:

▶ **Definition 6** (Pseudo-deterministic NL)**.** A search problem R is in *pseudo-deterministic* NL (often denoted psdNL) if there exists a function s where all $x \in L_R$ satisfy $(x, s(x)) \in R$ and $|s(x)| \leq poly(x)$, there is a nondeterministic log-space machine V such that for every $x \in L_R$:

1. There exist nondeterministic choices N for the machine such that such that $V(x, N) = s(x)$.
2. For all possible nondeterministic choices N', $V(x, N') = s(x)$ or $V(x, N') = \perp$.

And for every $x \notin L_R$, for all nondeterministic choices N', $V(x, N') = \perp$.

[3] We remark that we use M to denote the proof sent by the prover Merlin, and not the algorithm implemented by the prover.

3 Pseudo-deterministic-AM algorithm for graph isomorphism

In this section we give an algorithm for finding an isomorphism between two graphs in AM that outputs the same answer with high probability. The way this algorithm works is that the prover will send the lexicographically first isomorphism to the verifier and then prove that it is the lexicographically first isomorphism. To prove that the isomorphism is the lexicographically first isomorphism, we label the graph and run a sequence of graph non-isomorphism protocols to show no lexicographically smaller isomorphism exists. We present an alternate proof of the same result in the appendix (the proof in the appendix is more group theoretic, whereas the proof below is more combinatorial).

▶ **Theorem 7.** *Finding an isomorphism between graphs can be done in* psdAM.

Proof. Let the vertices of G_1 be $v_1, v_2, \ldots, v_n$, and the vertices of G_2 be $u_1, u_2, \ldots, u_n$. We will show an AM algorithm which outputs a unique isomorphism ϕ. Our algorithm will proceed in n stages (which we will later show can be parallelized). After the kth stage, the values $\phi(v_1), \phi(v_2), \ldots, \phi(v_k)$ will be determined.

Suppose that the values $\phi(v_1), \phi(v_2), \ldots, \phi(v_k)$ have been determined. Then we will determine the smallest r such that there exists an isomorphism ϕ^* such that for $1 \leq i \leq k$, we have $\phi^*(v_i) = \phi(v_i)$, and in addition, $\phi^*(v_{k+1}) = u_r$. If we find r, we can set $\phi(v_{k+1}) = \phi^*(v_{k+1})$ and continue to the $k + 1^{th}$ stage.

To find the correct value of r, the (honest) prover will tell the verifier the value of r and ϕ. Then, to show that the prover is not lying, for each $r' < r$, the prover will prove that there exists no isomorphism ϕ' such that for $1 \leq i \leq k$, we have $\phi'(v_i) = \phi(v_i)$, and in addition, $\phi'(v_{k+1}) = u_{r'}$. To prove this, the verifier will pick G_1 or G_2, each with probability $1/2$. If the verifier picked G_1, he will randomly shuffle the vertices $v_{k+2}, \ldots, v_n$, and send the shuffled graph to the prover. If the verifier picked G_2, he will set $u'_i = \phi(v_i)$ for $1 \leq i \leq k$, and $u'_{k+1} = u_{r'}$, and shuffle the rest of the vertices. If the prover can distinguish between whether the verifier initially picked G_1 or G_2, then that implies there is no isomorphism sending v_i to $\phi(v_i)$ for $1 \leq i \leq k$, and sending v_{k+1} to $u_{r'}$. The prover now can show this for all $r' \leq r$ (in parallel), as well as exhibit the isomorphism ϕ, thus proving that r is the minimum value such that there is an isomorphism sending v_i to $\phi(v_i)$ for $1 \leq i \leq k$, and sending v_{k+1} to u_r.

The above n stages can be done in parallel in order to achieve a constant round protocol. To do so, in the first stage, the prover sends the isomorphism ϕ to the verifier. Then, the verifier can test (in parallel) for each k whether under the assumption that $\phi(v_1), \phi(v_2), \ldots, \phi(v_k)$ are correct, $\phi(v_{k+1})$ is the lexicographically minimal vertex which v_{k+1} can be sent to. The correctness of this protocol follows from the fact that multiple AM interactive proofs can be performed in parallel while maintaining soundness and completeness for all of the interactive proofs performed (as shown in appendix C.1 of [7]).

We note that in the above protocol, the prover only needs to have the power to solve graph isomorphism (and graph non-isomorphism). Also, we note that the above protocol uses private coins. While the protocol can be simulated with a public coin protocol [11], the simulation requires the prover to be very powerful. ◀

4 Lower bound on pseudo-deterministic AM algorithms

In this section, we establish that if any NP-complete problem has an AM protocol that outputs a unique witness with high probability, then the polynomial hierarchy collapses. To do this we show the analog of AM $\subseteq$ NP/*poly* for the pseudo-deterministic setting, and then

use this fact to get a NP/*poly* algorithm with a unique witness. We can then use [16] to show that NP $\subseteq$ coNP/*poly*, which obtains the hierarchy collapse.

We begin by proving that psdAM $\subseteq$ psdNP/*poly*:

▶ **Lemma 8.** *Suppose that there is a* psdAM *protocol for a search problem R, which on input $x \in L_R$, outputs $s(x)$. Then, the search problem R has a* psdNP/*poly algorithm which, on input x, outputs $s(x)$.*

Proof. Consider a psdAM protocol, and suppose that on input $x \in L_R$, it outputs $s(x)$.

Since we are guaranteed that when the verifier of the the psdAM accepts, it will output $s(x)$ with high probability, we can use standard amplification techniques to show that the verifier will output $s(x)$ with probability $1 - o(\exp(-n))$, assuming an honest prover, and will output anything other than $s(x)$ with probability $o(\exp(-n))$, even with a malicious prover. Then, by a union bound, there exists a choice of random string r that makes the verifier output $s(x)$ for all inputs $x \in \{0,1\}^n$ of size n with an honest prover, and that for malicious provers, the verifier will either reject or output $s(x)$. We encode this string r as the advice string for the NP/*poly* machine.

The NP/*poly* machine computing s can read r off the advice tape and then guess the prover's message, and whenever the verifier accepts, $s(x)$ will be output by that nondeterministic branch. Thus $s(x)$ can be computed by an NP/*poly* machine. ◀

Next, we show that if an NP-complete problem has a pseudo-deterministic-NP/*poly* algorithm, then the polynomial hierarchy collapses.

▶ **Theorem 9.** *Let $L \in$ NP be an* NP*-complete problem. Let R be a polynomial time algorithm such that there exists a polynomial p so that $x \in L$ if and only if $\exists y \in \{0,1\}^{p(|x|)} R(x,y)$. Suppose that there is a* psdAM *protocol that when given some $x \in L$, outputs a unique $s(x) \in \{0,1\}^{p(|x|)}$ such that $R(x, s(x)) = 1$. Then,* NP $\subseteq$ coNP/*poly and the polynomial hierarchy collapses to the third level.*

Proof. Assume that there is a psdAM protocol that when given some $\phi \in L$, outputs a unique $s(\phi) \in \{0,1\}^{p(|\phi|)}$ such that $R(\phi, s(\phi)) = 1$. From Lemma 8, we have that there exists psdNP/*poly* algorithm A that given $\phi \in L$, outputs a unique witness $s(\phi)$ for ϕ. Given such an algorithm A, we can construct a function g computable in psdNP/*poly* that on two inputs ϕ_1 and ϕ_2, $g(\phi_1, \phi_2)$ is one of either ϕ_1 or ϕ_2 with the condition that if either ϕ_1 or ϕ_2 is in L, then $g(\phi_1, \phi_2)$ is satisfiable. If neither ϕ_1 nor ϕ_2 are in L, then $g(\phi_1, \phi_2) = \perp$.

To construct such a g, define a function g' where $g'(\phi_1, \phi_2) = \{\phi_1, \phi_2\} \cap L$ (i.e., $g'(\phi_1, \phi_2)$ is the subset of $\{\phi_1, \phi_2\}$ consisting of satisfiable formulas). We construct g by reducing the language $L' = \{(\phi_1, \phi_2) | g'(\phi_1, \phi_2) \neq \emptyset\}$ (which is in NP, and hence reducible to L, since L is NP-complete) to L and running A to find a unique witness for g, which we can then turn into a witness for L'. Note that a witness for L' is either a witness for ϕ_1 or for ϕ_2. We can then check whether this unique witness is a witness for ϕ_1 or ϕ_2, and output the ϕ_i for which it is a witness (in the case that the witness works for both of the ϕ_i, we output the lexicographically first ϕ_i).

We note that we view g as a function on the set $\{\phi_1, \phi_2\}$. That is, we set $g(\phi_1, \phi_2) = g(\phi_2, \phi_1)$ (if a function g does not satisfy this property, we can create a g^* satisfying this property by setting $g^*(\phi_1, \phi_2) = g(min(\phi_1, \phi_2), max(\phi_2, \phi_1))$).

Now, our goal is to use g, which we know is computable in psdNP/*poly* to construct an NP/*poly* algorithm for $\bar{L}$ (the complement of L).

We construct the advice string for L for length n as follows. Our advice string will be a set S consisting of strings ϕ_i. Start out with $S = \emptyset$. We know that there exists a $\phi_1 \in \{0,1\}^n \cap L$

such that $g(\phi, \phi_1) = x$ for at least half of the set $\{\phi \in \{0,1\}^n \cap L | g(\phi, s) = s \forall s \in S\}$. Such an s exists since in expectation, when picking a random s, half of the ϕ's will satisfy $g(\phi, s) = x$. If we keep doing this, we get a set S with $|S| \leq poly(n)$ such that for every $\phi \in L$ of length n, there exists an $s \in S$ such that $g(\phi, s) = x$.

Now, to check that $\phi \in \bar{L}$ in NP/*poly* (where S as defined above is the advice), we compute $g(\phi, s)$ for every $s \in S$, and check that $g(\phi, s) = s$ for every $s \in S$ which is possible because $|S|$ is polynomial in n. It is clear that this algorithm accepts if $\phi \notin L$ and rejects if $\phi \in L$, so therefore $L \in$ coNP/*poly*, which implies that NP $\subseteq$ coNP/*poly*. Furthermore, NP $\subseteq$ coNP/*poly* implies that the polynomial hierarchy collapses to the third level. ◀

5 Pseudo-deterministic derandomization for BPP in subexponential time MA

In this section, we prove the existence of pseudo-deterministic subexponential time (time $O(2^{n^\epsilon})$ for every ϵ) MA protocols for problems in search-BPP for infinitely many input lengths.

In this section, we prove that every problem R in search-BPP has an MA proof where the verifier takes subexponential time (and the prover is unbounded). For completeness, we define search-BPP below:

▶ **Definition 10** (Search-BPP)**.** A binary relation R is in *search-BPP* if there exist probabilistic polynomial-time algorithms A, B such that

1. Given $x \in R_L$, A outputs a y such that with probability at least $2/3$, $(x, y) \in R$.
2. If y is output by A when run on x, and $(x, y) \notin R$, then B rejects on (x, y) with probability at least $2/3$. Furthermore, for all $x \in L_R$, with probability at least $1/2$ B accepts on (x, y) with probability at least $1/2$.

When $x \notin R_L$, A outputs $\perp$ with probability at least $2/3$.

The intuition of the above definition is that A is used to find an output y, and then B can be used to verify y, and amplify the success probability.

A main component of our proof will be the Nisan-Wigderson pseudo-random generator, which shows a way to construct pseudorandom strings given access to an oracle solving a problem of high circuit complexity.

To obtain the best running time for our pseudo-deterministic algorithm, we will need the iterated exponential functions first used in complexity theory by [20]. We will be considering functions with half-exponential growth, i.e. functions f such that $f(f(n)) \in O(2^{n^k})$ for some k.

▶ **Definition 11** (Fractional exponentials [20])**.** The fractional exponential function $e_\alpha(x)$ will be defined as $A^{-1}(A(x) + \alpha)$, where A is the solution to the functional equation $A(e^x - 1) = A(x) + 1$. In addition, we can construct such functions so that $e_\alpha(e_\beta(x)) = e_{\alpha+\beta}(x)$. It is clear from this definition that $e_1(n) = O(2^n)$, and that $e_{1/2}(e_{1/2}(x)) = O(2^n)$. We call a function f satisfying $f(x) = \Theta(e_{1/2}(x))$ a *half-exponential* function.

▶ **Definition 12** (Half-Exponential Time MA)**.** We define a *half-exponential time* MA *proof* to be an interactive MA proof in which the verifier runs in half-exponential time.

▶ **Theorem 13.** *Given a problem R in search-*BPP*, it is possible to obtain a pseudo-deterministic* MA *algorithm for R where the verifier takes subexponential time for infinitely many input lengths.*

Proof. From [20], we see that MA ∩ coMA where the verifier runs in half-exponential time cannot be approximated by polynomial-sized circuits. By Nisan-Wigderson [21], it follows that in half-exponential time MA, one can construct a pseudorandom generator with half-exponential stretch which is secure against any given polynomial-size circuit for infinitely many input lengths. We provide more details below.

Let T be the truth-table of a hard function in MA ∩ coMA. Then, let R be a relation in search-BPP. Recall from Definition 10 that there is an algorithm A that given x, produces y such that $(x, y) \in R$ with high probability if $x \in R_L$.

We will now describe the MA protocol. First, the prover sends T to the verifier and proves that it is indeed the truth table of the hard function in half-exponential time MA (which can be done in half-exponential time). With T in hand, the verifier can then compute the output of the Nisan-Wigderson pseudorandom generator. The verifier loops through the seeds in lexicographic order and uses the pseudorandom generator on each seed to create pseudo-random strings, which the verifier then uses as the randomness for A. Each time, the verifier tests whether $(x, A(x, r)) \in R$ (which can be done in BPP, and hence also in MA) and returns the first such valid output.

This will output the same solution whenever the verifier both gets the correct truth-table for the PRG, and succeeds in testing for each PRG output whether the output it provides is valid. Both of these happen with high probability, and thus this is a pseudo-deterministic subexponential-time MA algorithm for any problem in search-BPP which succeeds on infinitely many input lengths. ◀

6 Uniqueness in NL

In this section, we prove that every problem in search-NL can be made pseudo-deterministic. For completeness we include a definition of search-NL:

▶ **Definition 14** (search-NL)**.** A search problem R is in *search*-NL if there is a nondeterministic log-space machine V such that for every $x \in L_R$,

1. There exist nondeterministic choices N for the machine such that such that $V(x, N) = y$, and $(x, y) \in R$.
2. For all possible nondeterministic choices N', $(x, V(x, N')) \in R$, or $V(x, N') = \perp$.

And for every $x \notin L_R$, for all nondeterministic choices N', $V(x, N') = \perp$.

▶ **Theorem 15** (Pseudo-determinism NL)**.** *Every search problem in search-*NL *is in* psdNL*.*

One can think of the complete search problem for NL as: given a directed graph G, and two vertices s and t such that there is a path from s to t, find a path from s to t. Note that the standard nondeterministic algorithm of simply guessing a path will result in different paths for different nondeterministic guesses. Our goal will be to find a unique path, so that on different nondeterministic choices, we will not end up with a path which is not the unique one.

The idea will be to find the lexicographically first shortest path (i.e, if the min-length path from s to t is of length d, we will output the lexicographically first path of length d from s to t). To do so, first we will determine the length d of the min-length path from s to t. Then, for each neighbor of s, we will check if it has a path of length $d - 1$ to t, and move to the first such neighbor. Now, we have reduced the problem to finding a unique path of length $d - 1$, which we can do recursively.

The full proof is given below:

Proof. Given a problem in search-NL, consider the set of all min-length computation histories. We will find the lexicographically first successful computation history in this set.

To do so, we first (nondeterministically) compute the length of the min-length computation history. This can be done because coNL = NL (so if the shortest computation history is of size T, one can show a history of size T. Also, because it is coNL to show that there is no history of size up to $T-1$, we can show that there is no history of size less than T in NL).

In general, using the same technique, given a state S of the NL machine, we can tell what is the shortest possible length for a successful computation history starting at S.

Our algorithm will proceed as follows. Given a state S (which we initially set to be the initial configuration of the NL machine), we will compute T, the length of the shortest successful computation path starting at S. Then, for each possible nondeterministic choice, we will check (in NL) whether there exists a computation history of length $T-1$ given that nondeterministic choice. Then, we will choose the lexicographically first such nondeterministic choice, and recurse.

This algorithm finds the lexicographically first computation path of minimal length which is unique. Hence, the algorithm will always output the same solution (or reject), so the algorithm is pseudo-deterministic. ◀

7 Structural Results

In [8], Goldreich et al showed that the set of total search problems solved by pseudo-deterministic polynomial time randomized algorithms equals the set of total search problems solved by deterministic polynomial time algorithms, with access to an oracle to decision problems in BPP. In [10], this result was extended to the context of RNC. We show analogous theorems here. In the context of MA, we show that for total search problems, psdMA = search$-\mathrm{P}^{\mathrm{MA}\cap\mathrm{coMA}}$.[4] In other words, any pseudo-deterministic MA algorithm can be simulated by a polynomial time search algorithm with an oracle solving decision problems in MA $\cap$ coMA, and vice versa.

In the case of search problems that are not total, we show that psdMA equals to the class search$-\mathrm{P}^{\mathrm{promise}-(\mathrm{MA}\cap\mathrm{coMA})}$, where when the input x is in L_R, all queries to the oracle must be in the promise. We note that generally, when having an oracle to a promise problem, one is allowed to query the oracle on inputs not in the promise, as long as the output of the algorithm as a whole is correct for all possible answers the oracle gives to such queries. In our case, we simply do not allow queries to the oracle to be in the promise. Such reductions have been called *smart* reductions [12].

We show similar theorems for AM, and NP. Specifically, we show psdAM = search$-\mathrm{P}^{\mathrm{promise}-(\mathrm{AM}\cap\mathrm{coAM})}$ and psdNP = search$-\mathrm{P}^{\mathrm{promise}-(\mathrm{NP}\cap\mathrm{coNP})}$, where the reductions to the oracles are smart reductions.

In the case of total problems, one can use a similar technique to show psdAM = search$-\mathrm{P}^{\mathrm{AM}\cap\mathrm{coAM}}$ and psdNP = search$-\mathrm{P}^{\mathrm{NP}\cap\mathrm{coNP}}$, where the oracles can only return answers to total decision problems.

▶ **Theorem 16.** *The class* psdMA *equals the class* search$-\mathrm{P}^{\mathrm{promise}-(\mathrm{MA}\cap\mathrm{coMA})}$*, where on any input $x \in L_R$, the all queries to the oracle are in the promise.*

Proof. The proof is similar to the proofs in [8] and [10] which show similar reductions to decision problems in the context of pseudo-deterministic polynomial time algorithms and

[4] What we call search$-$P is often denoted as FP.

pseudo-deterministic NC algorithms.

First, we show that a polynomial time algorithm with an oracle for promise$-(\mathrm{MA} \cap \mathrm{coMA})$ decision problems which only asks queries in the promise has a corresponding pseudo-deterministic MA algorithm. Consider a polynomial time algorithm A which uses an oracle for promise$-(\mathrm{MA} \cap \mathrm{coMA})$. We can simulate A by an MA protocol where the prover sends the verifier the proof for every question which A asks the oracle. Then, the verifier can simply run the algorithm from A, and whenever he accesses the oracle, he instead verifies the proof sent to him by the prover.

We note that the condition of a smart reduction is required in order for the prover to be able to send to the verifier the list of all queries A will make to the oracle. If A can ask the oracle queries not in the promise, it may be that on different executions of A, different queries will be made to the oracle (since A is a adaptive, and the queries A makes may depend on the answers returned by the oracle for queries not in the promise), so the prover is unable to predict what queries A will need answered.

We now show that a pseudo-deterministic MA algorithm B has a corresponding polynomial time algorithm A that uses a promise$-(\mathrm{MA} \cap \mathrm{coMA})$ oracle while only querying on inputs in the promise. On input $x \in L_R$, the polynomial time algorithm can ask the promise$-(\mathrm{MA} \cap \mathrm{coMA})$ oracle for the first bit of the unique answer given by B. This is a decision problem in promise$-(\mathrm{MA} \cap \mathrm{coMA})$ since it has a constant round interactive proof (namely, run B and then output the first bit). Similarly, the algorithm A can figure out every other bit of the unique answer, and then concatenate those bits to obtain the full output.

Note that it is required that the oracle is for promise$-(\mathrm{MA} \cap \mathrm{coMA})$, and not just for promise$-$MA, since if one of the bits of the output is 0, the verifier must be able to convince the prover of that (and this would require a promise$-$coMA protocol). ◀

A very similar proof shows the following:

▶ **Theorem 17.** *The class* psdNP *equals the class* search$-\mathrm{P}^{\mathrm{promise}-(\mathrm{NP} \cap \mathrm{coNP})}$*, where on any input $x \in L_R$, all queries to the oracle are in the promise.*

We now prove a similar theorem for the case of AM protocols. We note that this is slightly more subtle, since it's not clear how to simulate a search$-\mathrm{P}^{\mathrm{promise}-(\mathrm{AM} \cap \mathrm{coAM})}$ protocol using only a constant number of rounds of interaction, since the search-P algorithm may ask polynomial many queries in an adaptive fashion.

▶ **Theorem 18.** *The class* psdAM *equals the class* search$-\mathrm{P}^{\mathrm{promise}-(\mathrm{AM} \cap \mathrm{coAM})}$*, where on any input $x \in L_R$, the all queries to the oracle are in the promise.*

Proof. First, we show that a polynomial time algorithm with an oracle for promise$-(\mathrm{AM} \cap \mathrm{coAM})$ decision problems where the queries are all in the promise has a corresponding pseudo-deterministic AM algorithm. We proceed similarly to the proof of Theorem 16. Consider a polynomial time algorithm A which uses an oracle for promise$-(\mathrm{AM} \cap \mathrm{coAM})$. The prover will internally simulate that algorithm A, and then send to the verifier a list of all queries that A makes to the promise$-(\mathrm{AM} \cap \mathrm{coAM})$ oracle. Then, the prover can prove the answer (in parallel), to all of those queries.

To prove correctness, suppose that the prover lies about at least one of the oracle queries. Then, consider the first oracle query to which the prover lied. Then, by a standard simulation argument, one can show that it can be made overwhelmingly likely that the verifier will discover that the prover lied on that query.

Once all queries have been answered by the verifier the algorithm B can run like A, but instead of querying the oracle, it already knows the answer since the prover has proved it to him.

The proof that a pseudo-deterministic MA algorithm B has a corresponding polynomial time algorithm A that uses an promise$-(\mathrm{AM} \cap \mathrm{coAM})$ oracle is identical to the proof of Theorem 16 ◀

As a corollary of the above, we learn that private coins are no more powerful than public coins in the pseudo-deterministic setting:

▶ **Corollary 19.** *A pseudo-deterministic constant round interactive proof using private coins can be simulated by a pseudo-deterministic constant round interactive proof using public coins.*

Proof. By Theorem 18, we can view the algorithm as an algorithm in search$-\mathrm{P}^{\mathrm{AM} \cap \mathrm{coAM}}$.

By a similar argument to that in Theorem 18, one can show that psdIP = search$-\mathrm{P}^{\mathrm{IP} \cap \mathrm{coIP}}$, where in this context IP refers to *constant* round interactive proofs using *private coins*, and AM refers to constant round interactive proofs using *public coins*. Since promise$-(\mathrm{AM} \cap \mathrm{coAM}) =$ promise$-(\mathrm{IP} \cap \mathrm{coIP})$, since every constant round *private coin* interactive proof for decision problems can be simulated by a constant round interactive proof using *public coins* [11], we have:

$$\mathrm{psdAM} = \text{search}-\mathrm{P}^{\text{promise}-(\mathrm{AM} \cap \mathrm{coAM})} = \text{search}-\mathrm{P}^{\text{promise}-(\mathrm{IP} \cap \mathrm{coIP})} = \mathrm{psdIP}.$$ ◀

8 Discussion and Open Problems

Pseudo-determinism and TFNP: The class of total search problems solvable by pseudo-deterministic NP algorithms is a very natural subset of TFNP, the set of all total NP search problems. It is interesting to understand how the set of total psdNP problems fits in TFNP. For example, it is not known whether TFNP = psdNP. It would be interesting either to show that every problem in TFNP has a pseudo-deterministic NP algorithm, or to show that under plausible assumptions there is a problem in TFNP which does not have a pseudo-deterministic NP algorithm.

Similarly, it is interesting to understand the relationship of psdNP to other subclasses of TFNP. For example, one can ask whether every problem in PPAD has a pseudo-deterministic NP algorithm (i.e., given a game, does there exists a pseudo-deterministic NP or AM algorithm which outputs a Nash Equilibrium), or whether under plausible assumptions this is not the case. Similar questions can be asked for CLS, PPP, and so on.

Pseudo-determinism in Lattice problems: There are several problems in the context of lattices which have NP (and often also NP $\cap$ coNP) algorithms [1]. Notable examples include gap-SVP and gap-CVP, for certain gap sizes. It would be interesting to show pseudo-deterministic interactive proofs for those problems. In other words, one could ask: does there exists an AM protocol for gap-SVP so that when a short vector exists, the *same* short vector is output every time. Perhaps more interesting would be to show, under plausible cryptographic assumptions, that certain such problems *do not* have psdAM protocols.

Pseudo-determinism and Number Theoretic Problems: The problem of generating primes (given a number n, output a prime greater than n), and the problem of finding primitive roots (given a prime p, find a primitive root mod p) have efficient randomized algorithms, and

have been studied in the context of pseudo-determinism [13, 6, 22], though no polynomial time pseudo-deterministic algorithms have been found. It is interesting to ask whether these problems have polynomial time psdAM protocols.

The Relationship between psdAM **and** search−BPP**:** One of the main open problems in pseudo-determinism is to determine whether every problem in search−BPP also has a polynomial time pseudo-deterministic algorithm. This remains unsolved. As a step in that direction (and as an interesting problem on its own), it is interesting to determine whether search−BPP $\subseteq$ psdAM. In this paper, we proved a partial result in this direction, namely that search−BPP $\subseteq i.o.\text{psdMA}_{\text{SUBEXP}}$.

Zero Knowledge Proofs of Uniqueness: The definition of pseudo-deterministic interactive proofs can be extended to the context of Zero Knowledge. In other words, the verifier gets no information other than the answer, and knowing that it is the unique/canonical answer. It is interesting to examine this notion and understand its relationship to psdAM.

The Power of the Prover in pseudo-deterministic interactive proofs: Consider a search problem which can be solved in IP where the prover, instead of being all-powerful, is computationally limited. We know that such a problem can be solved in psdIP if the prover has unlimited computational power (in fact, one can show it is enough for the prover to be in PSPACE). In general, if the prover can be computationally limited for some IP protocol, can it also be computationally limited for a psdIP protocol for the same problem? It is also interesting in general to compare the power needed for the psdIP protocol compared to the power needed to solve the search problem non-pseudo-deterministically. Similar questions can be asked in the context of AM.

The Power of the Prover in pseudo-deterministic private vs public coins proofs: In our psdAM protocol for Graph Isomorphism, the verifier uses private coins, and the prover is weak (it can be simulated by a polynomial time machine with an oracle for graph isomorphism). If using public coins, what power would the prover need? In general, it is interesting to compare the power needed by the prover when using private coins vs public coins in psdAM and psdIP protocols.

Pseudo-deterministic interactive proofs for setting cryptographic global system parameters: Suppose an authority must come up with global parameters for a cryptographic protocol (for instance, a prime p and a primitive root g of p, which would be needed for a Diffie-Hellman key exchange). It may be important that other parties in the protocol know that the authority did not come up with these parameters because he happens to have a trapdoor to them. If the authority proves to the other parties that the parameters chosen are canonical, the other parties now know that the authority did not just pick these parameters because of a trapdoor (instead, the authority had to pick those parameters, since those are the canonical ones). It would be interesting to come up with a specific example of a protocol along with global parameters for which there is a pseudo-deterministic interactive proof showing the parameters are unique.

References

1 Dorit Aharonov and Oded Regev. Lattice problems in NP ∩ coNP. *Journal of the ACM (JACM)*, 52(5):749–765, 2005.

2 László Babai. Trading group theory for randomness. In *Proceedings of the seventeenth annual ACM symposium on Theory of computing*, pages 421–429. ACM, 1985.

3 László Babai and Eugene M Luks. Canonical labeling of graphs. In *Proceedings of the fifteenth annual ACM symposium on Theory of computing*, pages 171–183. ACM, 1983.

4 J. Cai, V. Chakaravarthy, L. Hemaspaandra, and M. Ogihara. Competing provers yield improved Karp–Lipton collapse results. *Information and Computation*, 198(1):1–23, 2005.

5 John J Cannon. Construction of defining relators for finite groups. *Discrete Mathematics*, 5(2):105–129, 1973.

6 Eran Gat and Shafi Goldwasser. Probabilistic search algorithms with unique answers and their cryptographic applications. In *Electronic Colloquium on Computational Complexity (ECCC)*, volume 18, page 136, 2011.

7 Oded Goldreich. *Modern cryptography, probabilistic proofs and pseudorandomness*, volume 17. Springer Science & Business Media, 1998.

8 Oded Goldreich, Shafi Goldwasser, and Dana Ron. On the possibilities and limitations of pseudodeterministic algorithms. In *Proceedings of the 4th conference on Innovations in Theoretical Computer Science*, pages 127–138. ACM, 2013.

9 Oded Goldreich, Silvio Micali, and Avi Wigderson. Proofs that yield nothing but their validity or all languages in NP have zero-knowledge proof systems. *Journal of the ACM (JACM)*, 38(3):690–728, 1991.

10 Shafi Goldwasser and Ofer Grossman. Perfect bipartite matching in pseudo-deterministic RNC. In *Electronic Colloquium on Computational Complexity (ECCC)*, volume 22, page 208, 2015.

11 Shafi Goldwasser and Michael Sipser. Private coins versus public coins in interactive proof systems. In *Proceedings of the eighteenth annual ACM symposium on Theory of computing*, pages 59–68. ACM, 1986.

12 Joachim Grollmann and Alan L Selman. Complexity measures for public-key cryptosystems. *SIAM Journal on Computing*, 17(2):309–335, 1988.

13 Ofer Grossman. Finding primitive roots pseudo-deterministically. In *Electronic Colloquium on Computational Complexity (ECCC)*, volume 22, page 207, 2015.

14 E. Hemaspaandra, L. Hemaspaandra, and C. Menton. Search versus decision for election manipulation problems. In *Proceedings of the 30th Annual Symposium on Theoretical Aspects of Computer Science*, pages 377–388. Leibniz International Proceedings in Informatics (LIPIcs), 2013.

15 L. Hemaspaandra and D. Narváez. The opacity of backbones. In *AAAI-2017*, pages 3900–3906. AAAI Press, 2017.

16 Lane A Hemaspaandra, Ashish V Naik, Mitsunori Ogihara, and Alan L Selman. Computing solutions uniquely collapses the polynomial hierarchy. *SIAM Journal on Computing*, 25(4):697–708, 1996.

17 Dhiraj Holden. A note on unconditional subexponential-time pseudo-deterministic algorithms for BPP search problems. *arXiv preprint arXiv:1707.05808*, 2017.

18 Neil Immerman. Nondeterministic space is closed under complementation. *SIAM Journal on computing*, 17(5):935–938, 1988.

19 Rudolf Mathon. A note on the graph isomorphism counting problem. *Information Processing Letters*, 8(3):131–136, 1979.

20 Peter Bro Miltersen, N Variyam Vinodchandran, and Osamu Watanabe. Super-polynomial versus half-exponential circuit size in the exponential hierarchy. In *International Computing and Combinatorics Conference*, pages 210–220. Springer, 1999.

21 Noam Nisan and Avi Wigderson. Hardness vs randomness. *Journal of computer and System Sciences*, 49(2):149–167, 1994.

22 Igor C Oliveira and Rahul Santhanam. Pseudodeterministic constructions in subexponential time. *arXiv preprint arXiv:1612.01817*, 2016.
23 Omer Reingold, Guy N Rothblum, and Ron D Rothblum. Constant-round interactive proofs for delegating computation. In *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing*, pages 49–62. ACM, 2016.
24 Adi Shamir. IP=PSPACE. *Journal of the ACM (JACM)*, 39(4):869–877, 1992.
25 Róbert Szelepcsényi. The method of forced enumeration for nondeterministic automata. *Acta Informatica*, 26(3):279–284, 1988.

A Alternate Algorithm for Graph Isomorphism in pseudo-deterministic AM

In this section, we present another psdAM algorithm for Graph Isomorphism, this one more group theoretic (as opposed to the more combinatorial approach of the algorithm in Section 3). The method we use to do this involves finding the lexicographically first isomorphism using group theory. In particular, the verifier will obtain the automorphism group of one of the graphs from the prover and verify that it is indeed the automorphism group, and then the verifier will convert an isomorphism obtained from the prover into the lexicographically first isomorphism between the two graphs. We will define the group-theoretic terms used below.

▶ **Definition 20** (Automorphism Group)**.** The *automorphism group* $Aut(G)$ of a graph is the set of permutations $\phi : G \to G$ such that for every $u, v \in V(G)$, $(u, v) \in E(G) \iff (\phi(u), \phi(v)) \in E(G)$ (i.e., ϕ is an automorphism of G).

▶ **Definition 21** (Stabilize)**.** Given a set S and elements $\alpha_1, \alpha_2, ..., \alpha_i \in S$, we say that a permutation $\phi : S \to S$ *stabilizes* $\{\alpha_1, \alpha_2, ..., \alpha_k\}$ iff $\phi(\alpha_i) = \alpha_i$ for $i \in \{1, ..., k\}$. We also say that a group G *stabilizes* $\{\alpha_1, \alpha_2, ..., \alpha_k\}$ when every $\phi \in G$ stabilizes $\{\alpha_1, \alpha_2, ..., \alpha_k\}$.

▶ **Definition 22** (Stabilizer)**.** The *stabilizer* of an element s in S for a group G acting on S is the set of elements of G that stabilize s.

▶ **Lemma 23.** *Suppose that we are given a tuple (G_1, G_2, H, ϕ) where G_1 and G_2 are graphs, $H = Aut(G_1)$ is represented as a set of generators, and ϕ an isomorphism between G_1 and G_2. Then, in polynomial time, we can compute a unique isomorphism ϕ^* from G_1 to G_2 independent of the choice of ϕ and the representation of H.*

Proof. We use the algorithm given in [5] to compute a canonical coset representative, observing that the set of isomorphisms between G_1 and G_2 is a coset of the automorphism group of G_1. Let $\alpha_1, ..., \alpha_t$ be a basis of H, i.e., a set such that any $h \in H$ fixing $\alpha_1, ..., \alpha_t$ is the identity. Let H_i be the subgroup of H that stabilizes $\alpha_1, ..., \alpha_{i-1}$. Now, let U_i be a set of coset representatives of H_{i+1} in H_i. Given the generators of H_i, we can calculate U_i, and by Schreier's theorem we can calculate the generators for H_{i+1}. In this fashion, we can get generators and coset representatives for all the H_i. To produce ϕ^*, we do the following.

FIND-FIRST-ISOMORPHISM
1 $\phi^* = \phi$
2 For $i = 1, ..., t$
3 Let $P_i = \{\phi^* u | u \in U_i\}$.
4 Set $\phi^* = \arg\min_{\phi \in P_i}(\phi(\alpha_i))$.

To see that this produces a unique isomorphism that does not depend on ϕ, observe that $\phi^*(\alpha_1)$ is the minimum possible value of $\phi(\alpha_1)$ over all isomorphisms of G_1 to G_2 as U_1 is a set of coset representatives for the stabilizer of α_1 over H. Also, if $\phi^*(\alpha_i)$ is fixed for $i \in \{1, ..., k\}$, then $\phi^*(\alpha_{k+1})$ is the minimum possible value of $\phi(\alpha_{k+1})$ over all isomorphisms which take α_1 to $\phi^*(\alpha_1)$, α_2 to $\phi^*(\alpha_2)$,..., and α_k to $\phi^*(\alpha_k)$, as U_{i+1} stabilizes $\alpha_1, ..., \alpha_k$, so everything in P_{i+1} takes α_1 to $\phi^*(\alpha_1)$, α_2 to $\phi^*(\alpha_2)$,..., and α_k to $\phi^*(\alpha_k)$. This implies that ϕ^* does not depend on ϕ and is unique. ◀

Given this result, this means that it suffices to show a protocol that lets the verifier obtain a set of generators for the automorphism group of G_1 and an isomorphism that are correct with high probability, as by the above lemma this can be used to obtain a unique isomorphism between G_1 and G_2 independent of the isomorphism or the generators.

▶ **Theorem 24.** *There exists an interactive protocol for graph isomorphism such that with high probability, the isomorphism that is output by the verifier is unique, where in the case of a cheating prover the verifier fails instead of outputting a non-unique isomorphism. In other words, finding an isomorphism between graphs can be done in* psdAM.

Proof. From Lemma 23, it suffices to show an interactive protocol that computes the automorphism group of a graph in a verifiable fashion. [19] reduces the problem of computing the generators of the automorphism group to the problem of finding isomorphisms. Using this reduction, we can make a constant-round interactive protocol to determine the automorphism group by finding the isomorphisms in parallel. The reason we can do this in parallel is that [19] implies that there are $O(n^4)$ different pairs of graphs to check and for each pair of graphs we either run the graph isomorphism protocol or the graph non-isomorphism protocol. In the case of the graph isomorphism protocol, the verifier need only accept with an isomorphism in hand; for graph non-isomorphism, the messages sent to the prover are indistinguishable between the two graphs when they are isomorphic, so since the graphs and permutations are chosen independently, there is no way for the prover to correlate their answers to gain a higher acceptance probability for isomorphic graphs. Thus this means that the verifier can determine the automorphism group of a graph and verify that it is indeed the entire automorphism group. Using Lemma 23 we then see that the prover just has to give the verifier an isomorphism, and verifier can compute a unique isomorphism using the automorphism group. ◀

Simple Doubly-Efficient Interactive Proof Systems for Locally-Characterizable Sets*

Oded Goldreich[1] **and Guy N. Rothblum**[†2]

1 **Weizmann Institute of Science, Rehovot, Israel**
oded.goldreich@weizmann.ac.il
2 **Weizmann Institute of Science, Rehovot, Israel**
rothblum@alum.mit.edu

Abstract

A proof system is called doubly-efficient if the prescribed prover strategy can be implemented in polynomial-time and the verifier's strategy can be implemented in almost-linear-time.

We present direct constructions of doubly-efficient interactive proof systems for problems in $\mathcal{P}$ that are believed to have relatively high complexity. Specifically, such constructions are presented for t-CLIQUE and t-SUM. In addition, we present a generic construction of such proof systems for a natural class that contains both problems and is in $\mathcal{NC}$ (and also in $\mathcal{SC}$). The proof systems presented by us are significantly simpler than the proof systems presented by Goldwasser, Kalai and Rothblum (*JACM*, 2015), let alone those presented by Reingold, Rothblum, and Rothblum (*STOC*, 2016), and can be implemented using a smaller number of rounds.

1998 ACM Subject Classification F.0 Theory of Computation: General

Keywords and phrases Interactive Proofs, Fine-Grained Complexity

Digital Object Identifier 10.4230/LIPIcs.ITCS.2018.18

1 Introduction

The notion of interactive proof systems, put forward by Goldwasser, Micali, and Rackoff [16], and the demonstration of their power by Lund, Fortnow, Karloff, and Nisan [18] and Shamir [23] are among the most celebrated achievements of complexity theory. Recall that an interactive proof system for a set S is associated with an interactive verification procedure, V, that can be made to accept any input in S but no input outside of S. That is, there exists an interactive strategy for the prover that makes V accepts any input in S, but no strategy can make V accept an input outside of S, except with negligible probability. (See [12, Chap. 9] for a formal definition as well as a wider perspective.)

The original definition does not restrict the complexity of the strategy of the prescribed prover and the constructions of [18, 23] use prover strategies of high complexity. This fact limits the applicability of these proof systems in practice. (Nevertheless, such proof systems may be actually applied when the prover knows something that the verifier does not know, such as an NP-witness to an NP-claim, and when the proof system offers an advantage such as zero-knowledge [16, 13].)

Seeking to make interactive proof systems available for a wider range of applications, Goldwasser, Kalai and Rothblum put forward a notion of *doubly-efficient* interactive proof systems (also called *interactive proofs for muggles* [15] and *interactive proofs for delegating*

* A full version of the paper is available at https://eccc.weizmann.ac.il/report/2017/018/
† This research was supported by the ISRAEL SCIENCE FOUNDATION (grant No. 529/17).

computation [22]). In these proof systems the prescribed prover strategy can be implemented in polynomial-time and the verifier's strategy can be implemented in almost-linear-time. (We stress that unlike in *argument systems*, the soundness condition holds for all possible cheating strategies, and not only for feasible ones.) Restricting the prescribed prover to run in polynomial-time implies that such systems may exist only for sets in $\mathcal{BPP}$, and thus a polynomial-time verifier can check membership in such sets by itself. However, restricting the verifier to run in almost-linear-time implies that something can be gained by interacting with a more powerful prover, even though the latter is restricted to polynomial-time.

The potential applicability of doubly-efficient interactive proof systems was demonstrated by Goldwasser, Kalai and Rothblum [15], who constructed such proof systems for any set that has log-space uniform circuits of bounded depth (e.g., log-space uniform $\mathcal{NC}$). A recent work of Reingold, Rothblum, and Rothblum [22] provided such (constant-round) proof systems for any set that can be decided in polynomial-time and a bounded amount of space (e.g., for all sets in $\mathcal{SC}$).

1.1 The current work

In this work, we aim to develop a more "algorithmic" understanding of doubly-efficient interactive proofs. Studying $\mathcal{BPP}$ problems on a case-by-case basis, our goal is to identify structures and patterns that facilitate the design of efficient proof systems. Towards this goal, our main contributions are *identifying a rich and natural class of polynomial-time computations*, and *constructing far simpler doubly-efficient proof systems for this class*. The aforementioned class consists of all sets that can be locally-characterized by the conjunction of polynomially many local conditions, each of which can be expressed by Boolean formulae of polylogarithmic size (see definition in Section 4.1). The class of locally-characterizable sets is believed not to be in $\text{Dtime}(p)$ for any fixed polynomial p, and contains natural problems of interest such as determining whether a given graph *does not* contain a clique of constant size t. In particular, several problems in this class have played a central role in the recent theory of "hardness within $\mathcal{P}$" [27]. We note that the class of locally-characterizable sets is a sub-class of $\mathcal{NC} \cap \mathcal{SC}$, yet the interactive proofs we present are significantly simpler than those in [15, 22].

▶ **Theorem 1** (main result, loosely stated)**.** *Every locally-characterizable set has a simple doubly-efficient interactive proof system. Specifically, on input of length n, the verifier runs in $\widetilde{O}(n)$-time and the strategy of the prescribed prover can be implemented in $\widetilde{O}(n^{c+1})$-time, where n^c denotes the number of local conditions in the characterization. Furthermore, the interactive proof system uses public coins, has logarithmic round-complexity, and uses polylogarithmic communication.*

Studying proof systems for locally-characterizable sets has also shed light on the complexity of problems in this class. In our subsequent work [14], building on the work of [5], we leverage techniques developed in the current work to show worst-case to average-case reductions between problems in a closely related class.[1] These reductions run in nearly-linear time.

[1] The class we study in [14] contains problems that *count* the number of local conditions that are violated by the input, rather than simply checking their conjunction. The interactive proofs we construct in this work can readily be modified to verify these "local-counting" problems, see Remark 4. In addition, the size of formulae considered in [14] is slightly larger, but the interactive proofs presented here apply to this too (except that the verification time becomes slightly larger (i.e., $2^{\widetilde{O}(\log \log n)} \cdot n$ rather than $\widetilde{O}(n)$)).

Thus, if the class contains problems that are hard to solve in the worst case, then it also contains problems that are hard to solve on-average.

High-level overview of our proof systems.

Analogously to [18, 23], our first step is recasting membership in a locally-characterizable set as an algebraic problem. Specifically, the algebraic problem consists of computing the sum of polynomially many evaluations of a low-degree polynomial, where the particular polynomial is derived from the description of the locally-characterizable set. The interactive proof uses the sum-check protocol [18] to verify the correctness of the sum. We stress that we check a sum with polynomially many terms, and so the sum-check protocol uses logarithmically many rounds, whereas the celebrated results of [18, 23] deal with a sum having exponentially (or more) many terms, and so the sum-check protocol requires a linear (or more) number of rounds.

1.1.1 Extensions

Round/communication trade-offs

Using a different setting of parameters, the interactive proofs constructed in Theorem 1 can also give new trade-offs between the number of rounds and communication/verification complexity (i.e., we obtain $O(r)$-round doubly-efficient interactive proofs with $\widetilde{O}(n^{1+(1/r)})$ verification time). In particular, the number of rounds can be significantly smaller than the protocols of [15, 22] (see Theorem 5 and the comparison in Section 1.2).

A richer class

Theorem 1 can be extended to a richer class of sets. This extended class also checks a polynomial number of local conditions, but rather than checking their conjunction, we allow a constant number of alternations between conjunctions and disjunctions. For example, this extended class contains the problem of determining whether a given graph contains no dominating set of constant size t (i.e., whether for every set D of t vertices, there exists a vertex v that is not adjacent to at least one member of D). This extension corresponds to the levels of a known hierarchy of *parameterized complexity classes* [11]. See the discussion following Remark 8.

This non-trivial extension is our most technically involved contribution. The proof system we construct leverages ideas from the proof of Toda's Theorem [26], where we need to take care and use a small bias space with additional algebraic structure. See the details and digest in Section 4.4.

1.1.2 Two cases of interest

We also present direct constructions of proof systems for verifying two particularly appealing t-parameterized problems which are locally-characterizable. Both of these problems are widely believed not to be in, say, Dtime($n^{t/5}$):[2]

[2] Currently, t-CLIQUE is conjectured to require time $n^{c' \cdot t}$, where c' is any constant smaller than one third of the Boolean matrix multiplication exponent (see, e.g., [1]). Recall that t-CLIQUE is $\mathcal{W}[1]$-complete [10] and that solving it in time $n^{o(t)}$ refutes the ETH [9]. The 3-SUM conjecture was popularized in [20], and lower bounds for t-SUM were shown to follow from lower bounds for t-CLIQUE (see [2]).

t-no-CLIQUE: The set of n-vertex graphs that do not contain a clique of size t.

t-no-SUM: The set of n-long sequences of integers that contain no t elements that sum-up to zero.[3]

(Indeed, the aforementioned sets are the complements of the NP-sets called t-CLIQUE and t-SUM.)

We present direct constructions of proof systems for these two sets. The corresponding prover strategies can be implemented in time $\widetilde{O}(n^t)$ and $\widetilde{O}(n^{t+1})$, resp. Although the generic construction for the class of "locally-characterizable" sets, which is presented in Section 4.2, almost meets the parameters of constructions tailored for t-no-CLIQUE and t-no-SUM, we believe that direct constructions for natural problems are of interest. In particular, the construction tailored for t-no-CLIQUE, which is presented in Section 3, is simpler and more efficient than the general construction. It can also be viewed as a warm-up for the more general result. The construction tailored for t-no-SUM makes use of technical ideas that may be of independent interest, it is presented in Section 5 (see the digest at the end of that section).

1.2 Relation to prior work

We first compare our results to the interactive proofs obtained by [15, 22], as well as an interactive proof system of Thaler [25] for counting the number of t-cliques in a graph. We then discuss the relationship to the non-interactive proofs presented in [8, 28, 7].

Complexity comparison to [15, 22]

On top of yielding simpler proof systems, the round complexity of our system is smaller than those of [15, 22]. This is most striking in the case of [15], which uses a protocol of $O(\log^2 n)$ rounds (whereas Theorem 1 uses $O(\log n)$ rounds). As for [22], its constant round-complexity is exponential in the degree of the polynomial bounding the complexity of the set, whereas the constant round-complexity in Theorem 5 is linear in the latter constant.

Comparison to [25]

The recent work [25] gives an interactive proof system for counting the number of t-cliques in a graph.[4] His system uses a constant number of rounds, with $\widetilde{O}(n)$ communication, $O(|E|+n)$ verification time, and $O(|E| \cdot n^{t-2})$ proving time. The system that we present for the t-no-CLIQUE problem can also be used to verify the number of t-cliques (see Remark 4). In that system the verification time is $\widetilde{O}(|E|+n)$, the number of rounds is logarithmic, the communication is polylogarithmic, and the prover work is $\widetilde{O}(n^t)$. As remarked above, we can also trade off the communication complexity and the number of rounds (see Section 4.3, where this is done for the generic protocol for "locally-characterizable" sets). That would result in parameters similar to those obtained in [25], except that the prover complexity is $\widetilde{O}(n^t)$ rather than $O(|E| \cdot n^{t-2})$.

---

[3] Alternatively, we may consider sequences of positive integers and ask if they contain a t-subset that sums-up to target integer, which is also given as part of the input.

[4] We remark that the protocol of [25] operates in a more challenging streaming setting, which we do not consider or elaborate on in this work.

Comparison to [8, 28, 7]

Several recent works [8, 28, 7] constructed *non-interactive* proof systems for problems in $\mathcal{P}$. The main distinction with our work is that we focus on *interactive* proofs, and obtain proof systems with *faster verification* (and smaller amounts of communication). Similarly to our work, the proofs in the systems of [8, 28] can be produced in polynomial-time.

Carmosino *et al.* [8] construct $\mathcal{NP}$ certificates for 3-no-SUM. The certificates are of length $\widetilde{O}(n^{1.5})$ and can be verified in deterministic $\widetilde{O}(n^{1.5})$-time. Our proof systems for this problem are interactive, and the verification is probabilistic, but the communication is only polylogarithmic, and the verifier's work is almost-linear (indeed, this remains true for the t-no-SUM problem, for any constant t).

Williams [28] constructs $\mathcal{MA}$ proof systems for problems in $\mathcal{P}$. An $\mathcal{MA}$ proof system is one in which the prover sends a single message to the verifier, who runs a probabilistic verification procedure. Focusing on problems that have been at the center of a recent theory of "hardness within $\mathcal{P}$" [27], he constructs $\mathcal{MA}$ proof systems for counting the number of orthogonal vectors within a collection of n input vectors and for counting the number of t-cliques in a given graph. The $\mathcal{MA}$ proof for counting the number of t-cliques has length and verification time $\widetilde{O}(n^{\lfloor t/2 \rfloor + 2})$. Björklund and Kaski [7] construct an $\mathcal{MA}$ proof with length and verification time $\widetilde{O}(n^{(\omega+\epsilon)t/6})$, where ω is the exponent of square matrix multiplication over the integers and $\epsilon > 0$ is an arbitrarily small constant. The time to construct their proof is $\widetilde{O}(n^{(\omega+\epsilon)t/3})$, matching the best sequential algorithm known for solving the problem. Comparing these works with our *interactive* proof for t-no-CLIQUE (which can also be used to verify the number of cliques), the interactive proof has poly-logarithmic communication, the verifier's work is almost-linear, and the prover's work is $\widetilde{O}(n^t)$.

Inspired by Williams [28] and using one of his results, we also present an $\mathcal{MA}$ proof system of verification complexity $\widetilde{O}(n^{(c+1)/2})$ for every locally-characterizable set (i.e., the class considered in Theorem 1). This proof system is presented in the appendix. For a more restricted subclass of locally-characterizable sets, which includes the c-no-CLIQUE problem, we construct an $\mathcal{MA}$ proof with improved length and verification time $\widetilde{O}(n^{c/2})$.

Relation to [6]

Subsequently to our work, Ball *et al.* [6] propose an *interactive* proof system for the generalized orthogonal vectors problem (a problem in $\mathcal{BPP}$), where the verification can be performed in nearly-linear time. They note that their approach can be extended to other problems studied in [5], as well as to our notion of locally-characterizable sets and a more general algebraic class of problems. We remark that the generalized orthogonal vectors problem lies in the class of locally-characterizable sets, and thus our general construction also applies to this problem, and achieves similar performance to the protocol of [6]. We note also that their earlier independent work [5] constructs (building on [28]) an $\mathcal{MA}$ proof system for the generalized orthogonal vectors problem, with polynomial communication and verification time.

1.3 Organization and conventions

Section 3 presents our simplest proof system, which is for t-no-CLIQUE. Our generic construction for any locally-characterizable set is presented in Section 4: The corresponding class is defined in Section 4.1, the basic construction is in Section 4.2, extensions and ramifications are in Sections 4.3 and 4.4. We present our proof system for t-no-SUM in Section 5. This

proof system is not significantly simpler than the generic construction but uses an idea that may be of independent interest.

Sections 3, 4 and 5 can be read independently of one another, and without reading Section 2 in which we merely review the celebrated sum-check protocol. (We assume that the reader is familiar with the definition of an interactive proof system.) Brief conclusions are in Section 6.

Conventions

We assume that the verifier (resp., prover) has *direct access* to the common input; that is, each bit in the input can be read in unit cost. Unless stated explicitly differently, all logarithms are to base 2.

2 Preliminaries: The sum-check protocol

Fixing a finite field $\mathcal{F}$ and a set $H \subset \mathcal{F}$ (e.g., H may consist of the 0 and 1 elements of $\mathcal{F}$), we consider an m-variate polynomial $P : \mathcal{F}^m \to \mathcal{F}$ of individual degree d. Given a value v, the sum-check protocol is used to prove that

$$\sum_{\sigma_1,...,\sigma_m \in H} P(\sigma_1, ..., \sigma_m) = v. \tag{1}$$

The sum-check protocol (of [18]) proceeds in m iterations, starting with $v_0 = v$, such that in the i^{th} iteration the parties act as follows.

Prover's move: The prove computes a univariate polynomial of degree d

$$P_i(z) \stackrel{\text{def}}{=} \sum_{\sigma_{i+1},...,\sigma_m \in H} P(r_1, ..., r_{i-1}, z, \sigma_{i+1}, ..., \sigma_m) \tag{2}$$

where $r_1, ..., r_{i-1}$ are as determined in prior iterations, and sends P_i to the verifier.

Verifier's move: The verifier checks that $\sum_{\sigma \in H} P_i(\sigma) = v_{i-1}$ and rejects if inequality holds.

Otherwise, it selects r_i uniformly in $\mathcal{F}$, and sends it to the prover, while setting $v_i \leftarrow P_i(r_i)$. If all iteration are completed, the verifier conducts a final check. It computes the value of $P(r_1, ..., r_m)$ and accepts if and only if this value equals v_m.

Clearly, if (1) holds (and the prover acts according to the protocol), then the verifier accepts with probability 1. Otherwise, no matter what the prover does, the verifier accepts with probability at most $m \cdot d/|\mathcal{F}|$. The complexity of verification is dominated by the complexity of evaluating P (on a single point). As for the prescribed prover, it may compute the relevant P_i's by interpolation, which is based on computing the value of P at $(d+1) \cdot 2^{m-i}$ points, for $i \in [m]$.

3 The case of t-CLIQUE

For a parameter $t \in \mathbb{N}$, given a graph $G = ([n], E)$, the task is determining whether there exist t vertices such that the subgraph induced by them is a clique; that is, does there exist distinct $v_1, ..., v_t \in [n]$ such that $\bigwedge_{j,k \in [t]: j<k} \chi_{v_j,v_k}$, where $\chi_{u,v} = 1$ if and only if $\{u, v\} \in E$. (Our focus is on simple graphs, and so we assume that $\chi_{v,v} = 0$ for every $v \in [n]$.)

The set of YES-instances (i.e., having a t-clique) has an NP-proof system that uses proofs of length $t \log n$. We shall present a doubly-efficient interactive proof for the set of NO-instances. This system is based on the observation that membership of an n-vertex

graph in t-no-CLIQUE can be captured by an explicit low degree polynomial having n^t terms. Furthermore, each term of this polynomial can be evaluated in almost-linear time (in the size of the graph). Hence, applying the Sum-Check protocol (reviewed in Section 2) yields the desired proof system.

Letting $\ell = \log n$, consider a finite field $\mathcal{F}$ of prime size greater than n^t, and identify $\{0,1\}$ with the set H containing the zero and one elements of $\mathcal{F}$. Using this identification, we define a polynomial $P : (\mathcal{F}^\ell)^t \to \mathcal{F}$ such that

$$P(z^{(1)},...,z^{(t)}) = \prod_{j,k\in[t]:j<k} \sum_{\alpha,\beta\in H^\ell} \mathtt{EQ}(\alpha\beta, z^{(j)}z^{(k)}) \cdot \chi_{\alpha,\beta} \tag{3}$$

$$\text{where } \mathtt{EQ}(\overline{\gamma},\overline{z}) = \textstyle\prod_{i\in[2\ell]}(z_i\gamma_i + (1-z_i)(1-\gamma_i)). \tag{4}$$

(There is some abuse of notation in (3): In the first two occurrences, α and β are viewed as an elements of $H^t \subset \mathcal{F}^\ell$, whereas in the last occurrence they viewed as elements of $[n] \equiv \{0,1\}^\ell$.)

Note that P has individual degree $O(t^2)$, and that it is straightforward to evaluate it in time $O(t^2 \cdot 2^{2\ell} \cdot \ell) = \widetilde{O}(t^2 \cdot n^2)$. Also, for $\overline{\gamma},\overline{z} \in H^{2\ell}$, it holds that $\mathtt{EQ}(\overline{\gamma},\overline{z}) = 1$ if $\overline{\gamma} = \overline{z}$ and $\mathtt{EQ}(\overline{\gamma},\overline{z}) = 0$ otherwise. Hence, for $\overline{z} \in H^{t\ell}$, it holds that $P(\overline{z}) = \prod_{j,k\in[t]:j<k} \chi_{z^{(j)},z^{(k)}}$.

The key observation is that the graph G is a NO-instances if and only if for all $\overline{z} \in (H^\ell)^t$ it holds that $P(\overline{z}) = 0$. (This holds since, for distinct $v^{(1)},...,v^{(t)} \in H^\ell$, it holds that $P(v^{(1)},...,v^{(t)}) = 1$ if the subgraph induced by $v^{(1)},...,v^{(t)}$ is a clique, and $P(v^{(1)},...,v^{(t)}) = 0$ otherwise.)[5] Hence, *we obtain an interactive proof system* (for the set of NO-instances) *by applying the sum-check protocol to the claim* $\sum_{\overline{z}\in(H^\ell)^t} P(\overline{z}) = 0$.

The complexity of the verifier's strategy is dominated by the evaluation of P (as defined in (3)), which reduces to $\binom{t}{2}$ computations of sums having the form

$$\sum_{\alpha,\beta\in H^\ell} \mathtt{EQ}(\alpha\beta, r^{(j)}r^{(k)}) \cdot \chi_{\alpha,\beta} \tag{5}$$

where $j < k \in [t]$ and $r^{(1)} \cdots r^{(t)} \in (\mathcal{F}^\ell)^t$ are determined in the execution of the sum-check protocol. Writing (5) as $\sum_{(\alpha,\beta)\in H^{2\ell}\cap E} \mathtt{EQ}(\alpha\beta, r^{(j)}r^{(k)})$, we can evaluate this sum in time $O(|E| \cdot \ell) = \widetilde{O}(|E| + n)$.

Turning our attention to the complexity of proving, we observe that the prover has to compute the polynomials that arise in each of the iterations of the sum-check protocol. The prover can do so by computing partial sums with at most $|H|^{t\ell} = n^t$ terms, where computing each such term amounts to $\mathrm{poly}(t)$ evaluations of P. Hence, the prover's complexity is definitely bounded by $\widetilde{O}(n^t \cdot |E|)$. A closer inspection reveals that we can do better. Specifically, in the i^{th} iteration of the sum-check, the prover has to provide the univariate polynomial (in z)

$$\sum_{\overline{s}\in H^{t\ell-i}} P(\overline{r}, z, \overline{s}) \tag{6}$$

where $\overline{r} \in \mathcal{F}^{i-1}$ was determined in the previous iteration. This univariate polynomial can be found by interpolation, and so the complexity of finding it is $\mathrm{poly}(t) \cdot 2^{t\ell-i} \cdot O(|E| \cdot \ell)$, which is $\widetilde{O}(n^t)$ for $i \geq 2\ell$. Thus, we focus on the case of $i \in [2\ell - 1]$. In that case, the complexity is $n^t/2^i$ times the complexity of evaluating P on $\overline{r}u\overline{s}$, where $(\overline{r},u) \in \mathcal{F}^{i-1} \times \mathcal{F}$ and $\overline{s} \in H^{t\ell-i}$ (for $O(t^2)$ values of $u \in \mathcal{F}$). So the question is what is the complexity of evaluating P on such a $t\ell$-element argument (which has a $(t\ell - i)$-long suffix in $H^{t\ell-i}$).

[5] If $v^{(j)} = v^{(k)}$ for some $j < k$, then $P(v^{(1)},...,v^{(t)}) = 0$, since $\chi_{v^{(j)},v^{(k)}} = 0$.

Focusing on evaluating each of the inner sums (captured by (5)), we consider evaluating the sum $\sum_{\alpha,\beta\in H^\ell} \mathtt{EQ}(\alpha\beta, v)\cdot\chi_{\alpha,\beta}$, when given $v = uw \in \mathcal{F}^{2\ell}$ such that $w \in H^{2\ell-p}$, where p is determined by i and j,k (indeed, $p = i \in [2\ell]$ if $(j,k) = (1,2)$, but $p < i$ otherwise). Letting $\chi_\gamma = \chi_{\alpha,\beta}$, where $\gamma = \alpha\beta$ such that $|\alpha| = |\gamma|/2$, we have

$$\begin{aligned}\sum_{\gamma\in H^{2\ell}} \mathtt{EQ}(\gamma, uw)\cdot\chi_\gamma &= \sum_{(\gamma',\gamma'')\in H^p\times H^{2\ell-p}} \mathtt{EQ}(\gamma', u)\cdot\mathtt{EQ}(\gamma'', w)\cdot\chi_{\gamma'\gamma''}\\ &= \sum_{\gamma'\in H^p} \mathtt{EQ}(\gamma', u)\cdot\chi_{\gamma' w}\end{aligned}$$

where the second equality holds because for $\gamma'', w \in H^{2\ell-p}$ it holds that $\mathtt{EQ}(\gamma'', w) = 1$ if $\gamma'' = w$ and $\mathtt{EQ}(\gamma'', w) = 0$ otherwise. Hence, evaluating this sum takes time $O(2^p\cdot\ell)$. The final observation is that, in our application, it holds that $p \le i$, since the p values in $\mathcal{F}\setminus H$ can only arise from the values determined in the previous $i-1$ iterations and the single value used for interpolation in the current iteration. It follows that the complexity of implementing the prover's strategy in the i^{th} iteration is $\mathrm{poly}(t)\cdot 2^{t\ell-i}\cdot O(2^p\cdot\ell) = \widetilde{O}(n^t)$.

4 The general result

In this section we prove Theorem 1; that is, we present a simple doubly-efficient interactive proof system for any locally-characterizable set. The latter class is defined next.

4.1 A natural class: locally-characterizable sets

The following definition is related but different from the definition of "local characterization" that is often used in the property testing literature (see, Sudan's survey [24]). Most importantly, the latter definitions do not specify the complexity of the functions ϕ_n and $\pi_{n,j}$, and typically take p to be a constant.[6]

▶ **Definition 2** (locally-characterizable sets)**.** A set S is locally-characterizable if there exist a constant c, a polynomial p and a polynomial-time algorithm that on input n outputs $\mathrm{poly}(\log n)$-sized formulae $\phi_n : [n]^{p(\log n)}\times\{0,1\}^{p(\log n)}\to\{0,1\}$ and $\pi_{n,1},...,\pi_{n,p(\log n)} : \{0,1\}^{c\log n}\to[n]$ such that, for every $x\in\{0,1\}^n$, it holds that $x\in S$ if and only if for all $w\in\{0,1\}^{c\log n}$

$$\Phi_x(w) \stackrel{\text{def}}{=} \phi_n(\pi_{n,1}(w),...,\pi_{n,p(\log n)}(w), x_{\pi_{n,1}(w)},...,x_{\pi_{n,p(\log n)}(w)}) \tag{7}$$

equals 0.[7]

That is, each value of $w\in\{0,1\}^{c\log n}$ yields a local condition that refers to polylogarithmically many locations in the input (i.e., the locations $\pi_{n,1}(w),...,\pi_{n,p(\log n)}(w)\in[n]$). This local condition is captured by ϕ_n, and in its general form it depends both on the selected locations and on the value on the input in these locations. A simplified form, which suffices in many case, uses a local condition that only depends on the values of the input in these locations (i.e., $\phi_n : [n]^{p(\log n)}\times\{0,1\}^{p(\log n)}\to\{0,1\}$ only depends on the $p(\log n)$-bit suffix).

[6] In addition, the notion used in property testing does not restrict the domain of Φ_x to have size $\mathrm{poly}(|x|)$, although this can be assumed without loss of generality.

[7] To simplify our exposition, we require that in case of inputs in S, the predicate ϕ_n evaluates to 0 (rather than to 1).

The simplified form (in which $\phi_n : \{0,1\}^{p(\log n)} \to \{0,1\}$) suffices for capturing the specific problems studied in the previous two sections. Specifically, for fixed $t \in \mathbb{N}$, when representing n-vertex graphs by their adjacency matrix, denoted $x = (x_{r,c})_{r,c\in[n]}$, the t-CLIQUE problem is captured by $\Phi_x(i_1,...,i_t) = \wedge_{j<k} x_{i_j,i_k}$; that is, we use $\phi_{n^2} : \{0,1\}^{t^2} \to \{0,1\}$ and $\pi_{n^2,(j,k)} : \{0,1\}^{t\log n} \to [n^2]$ (for $j,k \in [t]$) such that $\phi_{n^2}(\sigma_{1,1},...,\sigma_{t,t}) = \wedge_{j<k}\sigma_{j,k}$ and $\pi_{n^2,(j,k)}(i_1,...,i_t) = (i_j,i_k)$. Likewise, with some abuse of notation, the t-SUM problem over $[m]$, where $x = (a_1,...,a_n,b) \in [m]$ (and $m = \mathrm{poly}(n)$), is captured by $\Phi_x(i_1,...,i_t) = 1$ if and only if $\sum_{j\in[t]} x_{i_j} = x_{n+1}$; that is, we use $\phi_n : [m]^{t+1} \to \{0,1\}$ such that $\phi_n(z_1,...,z_t,z_{t+1}) = \mathtt{TruthValue}(\sum_{j\in[t]} z_j \neq z_{t+1})$ and $\pi_{n,j} : \{0,1\}^{t\log n} \to [n+1]$ such that $\pi_{n,j}(i_1,...,i_t) = i_j$ if $j \in [t]$ and $\pi_{n,t+1}(i_1,...,i_t) = n+1$.

Note that the complement of every locally-characterizable set has a doubly-efficient interactive proof system. In this proof system, on input $x \in \{0,1\}^n$, letting $\ell' = c\ell = c\log n$, the prover finds an adequate $w \in \{0,1\}^{\ell'}$, sends it to the verifier, who retrieves the bits $x_{\pi_{n,1}(w)},...,x_{\pi_{n,p(\ell)}(w)}$ and evaluates ϕ_n on them. Indeed, in this NP-proof system, the prover runs in time $2^{\ell'}\cdot\widetilde{O}(|x|) = \mathrm{poly}(|x|)$, whereas the verifier runs in $\mathrm{poly}(\log|x|)$-time (given direct access to the input). On the other hand, the set of YES-instances of this set has a doubly-efficient interactive proof systems (since computing the function $\sum_{w\in\{0,1\}^{\ell'}}\Phi_x(w)$, where Φ_x is as in (7), is in $\mathcal{NC}$, and so the proof system of [15] can be used).[8] Here we present a direct construction.

4.2 Proof of Theorem 1

Letting $\ell = \log n$, we associate $[n]$ with $\{0,1\}^\ell$, and derive from each $\pi_{n,j} : \{0,1\}^{c\ell} \to [n]$ Boolean formulae $\pi_{n,j,1},...,\pi_{n,j,\ell} : \{0,1\}^{c\ell} \to \{0,1\}$ such that $\pi_{n,j,k}(w)$ is the k^{th} bit of $\pi_{n,j}(w)$. We may assume, without loss of generality, that the depth of each of the formulae (i.e., ϕ_n and the $\pi_{n,j,k}$'s) is logarithmic in their size (which is $\mathrm{poly}(\ell)$).[9] Next, for a finite field $\mathcal{F}$ of size $\mathrm{poly}(n)$, we construct arithmetic formula $\widehat{\phi}_n : \mathcal{F}^{(\ell+1)\cdot p(\ell)} \to \mathcal{F}$ and $\widehat{\pi}_{n,j,k} : \mathcal{F}^{c\ell} \to \mathcal{F}$ such that $\widehat{\phi}_n$ (resp., $\widehat{\pi}_{n,j,k}$) agrees with ϕ_n on $H^{\ell\cdot p(\ell)+p(\ell)}$ (resp., with $\pi_{n,j,k}$ on $H^{c\ell}$). The crucial point is that these arithmetic formulae preserve the depth of the Boolean counterparts, and so the degrees of the functions that they compute is upper bounded by $D = \mathrm{poly}(\log n) \ll |\mathcal{F}|$. (Note: $\mathcal{F}$ is chosen so that the latter inequality holds.) Now, letting $\widehat{\pi}_{n,j} : \mathcal{F}^{c\ell} \to \mathcal{F}^\ell$ such that $\widehat{\pi}_{n,j}(\overline{z}) = (\widehat{\pi}_{n,j,1}(\overline{z}),...,\widehat{\pi}_{n,j,\ell}(\overline{z}))$, we consider the function $\widehat{\Phi}_x : \mathcal{F}^{c\ell} \to \mathcal{F}$ (i.e., an extension of Φ_x) such that

$$\widehat{\Phi}_x(\overline{z}) \stackrel{\text{def}}{=} \widehat{\phi}_n(\widehat{\pi}_{n,1}(\overline{z}),...,\widehat{\pi}_{n,p(\ell)}(\overline{z}),X_1,...,X_{p(\ell)}) \tag{8}$$

$$\text{where } X_i = \textstyle\sum_{\alpha\in\{0,1\}^\ell} \mathtt{EQ}(\widehat{\pi}_{n,i}(\overline{z}),\alpha)\cdot x_\alpha \tag{9}$$

and, as in (4), $\mathtt{EQ}(\overline{y},\alpha) = \prod_{i\in[\ell]}(y_i\alpha_i + (1-y_i)(1-\alpha_i))$. That is, the value of $\widehat{\Phi}_x(\overline{z})$ is obtained by feeding to $\widehat{\phi}_n : \mathcal{F}^{(\ell+1)\cdot p(\ell)} \to \mathcal{F}$ the $(p(\ell)\cdot\ell + p(\ell))$-sequence consisting of $(\widehat{\pi}_{n,1}(\overline{z}),...,\widehat{\pi}_{n,p(\log n)}(\overline{z})) \in (\mathcal{F}^\ell)^{p(\ell)}$ and the $p(\ell)$-long sequence whose j^{th} location contains the field element $\sum_{\alpha\in\{0,1\}^\ell}\mathtt{EQ}(\widehat{\pi}_{n,j}(\overline{z}),\alpha)\cdot x_\alpha$.

We invoke the sum-check protocol on the claim that the sum $\sum_{w\in\{0,1\}^{c\ell}}\widehat{\Phi}_x(w)$ equals 0, relying on the fact that $\mathcal{F}$ is larger than $2^{c\ell}$. This protocol performs $c\ell$ iterations (i.e., it is applied only to the outer sum), and the verifier evaluates the residual expression, which

[8] Alternatively, one observes that this computation is in $\mathcal{SC}$ and use the proof system of [22].

[9] Note that the transformation to this form can be performed in polynomial time, whereas the relevant formulae are of $\mathrm{poly}(\log n)$-size.

amounts to evaluating the $\widehat{\pi}_{n,j}$'s and $\widehat{\phi}_n$ as well as computing $\text{poly}(\ell)$ sums of 2^ℓ terms each. The prover's computation is dominated by computing a sum of $2^{c\ell}$ terms, where each term requires a computation of the type conducted by the verifier. Recalling that $2^\ell = n$, it follows that the verifier runs in almost-linear-time (i.e., it runs in $\widetilde{O}(n)$-time), whereas the prover runs in polynomial-time (i.e., it runs in $\widetilde{O}(n^{c+1})$-time). ◀

Comment

Applied to the t-no-CLIQUE problem, the foregoing generic construction yields a verifier that runs in time that is almost-linear in the size of the adjacency matrix, whereas the tailored proof system (presented in Section 3) yields a verifier that runs in time that is almost linear in the number of edges. Furthermore, the prover's complexity in this generic construction is n times slower than that of the tailored proof system. Looking ahead, we note that when applied to the t-no-SUM problem, the generic construction yields a proof system of complexity that is comparable to that of the tailored proof system presented in Section 5.

▶ **Remark 3** (a relaxation of Definition 2)**.** *Theorem 1 holds also when relaxing the notion of locally-characterizable sets such that the* $\text{poly}(\log n)$*-sized formulae are required to be generated in* $\widetilde{O}(n)$*-time, rather than in* $\text{poly}(\log n)$*-time. Actually, the foregoing proof remains intact even if the said formulae may depend on* x *itself, but we consider the class as defined in Definition 2 more natural.*

▶ **Remark 4** (counting the number of violated condition)**.** *The interactive proof system presented above is applicable also to the task of verifying the number of violated conditions. The same applies also to the problem-tailored proof systems presented in Section 3 and (looking ahead) in Section 5. Note that the number of violated conditions in* t-no-CLIQUE *is the number of* t*-cliques in the graph. Similarly, the number of violated conditions for* t-no-SUM *is the number of* t*-tuples that sum to the target.*

4.3 Generalization: round versus computation trade-off

By using a set H of arbitrary size (rather than $H \equiv \{0,1\}$), we obtain a general trade-off between the computational complexity and the number of communication rounds. Specifically, the computational complexity increases by a factor of $\widetilde{O}(|H|)$, whereas the number of rounds is decaresed by a factor of $\log |H|$.

▶ **Theorem 5** (main result, restated)**.** *Every locally-characterizable set has a simple doubly-efficient interactive proof system. Specifically, on input of length* n *and using a parameter* $h \leq n$*, we get public-coin interactive proof systems of round-complexity* $O(\log_h n)$*, verification time* $\widetilde{O}(h \cdot n)$*, and proving time* $\widetilde{O}(h \cdot n^{c+1})$*.*

In particular, setting $h = n^\epsilon$ for any constant $\epsilon > 0$, yields a constant-round interactive proof of verification time $\widetilde{O}(n^{1+\epsilon})$. On the other hand, using $h = \log n$ maintains the computational complexity bounds of Theorem 1 (i.e., $\widetilde{O}(n)$-time verification and $\widetilde{O}(n^{c+1})$-time prover strategy) while using $o(\log n)$ rounds of communication.

Proof. We use $H = [h]$ but maintain $\ell = \log n$. Defining the $\pi_{n,j,k}$'s as above and letting $d = \log h$, we associate $\{0,1\}^{c\ell}$ with $H^{c\ell/d}$ and consider $\pi_{n,j,k} : H^{c\ell/d} \to \{0,1\}$. Now, we let $\widehat{\pi}_{n,j,k} : \mathcal{F}^{c\ell/d} \to \mathcal{F}$ be the low degree extension of $\pi_{n,j,k}$, and define $\widehat{\pi}_{n,j} : \mathcal{F}^{c\ell/d} \to \mathcal{F}^\ell$ as

before. The definition of $\widehat{\Phi}_x : \mathcal{F}^{c\ell/d} \to \mathcal{F}$ is analogous to (8) except that (9) is replaced by

$$X_i = \sum_{\alpha \in H^{\ell/d}} \left(\prod_{i \in [\ell/d]} \prod_{\beta \in H \setminus \{\alpha_i\}} \frac{\beta - z_i}{\beta - \alpha_i} \right) \cdot x_\alpha \tag{10}$$

Invoking the sum-check protocol (w.r.t the non-binary H) on the claim that the sum $\sum_{w \in H^{c\ell/d}} \widehat{\Phi}_x(w)$ equals 0, yields a protocol that performs $c\ell/d$ iterations. Again, the verifier evaluates the residual expression, which amounts to evaluating the $\widehat{\pi}_{n,j}$'s and $\widehat{\phi}_n$ as well as computing $\mathrm{poly}(\ell)$ sums of $h^{\ell/d} = 2^\ell$ terms each, but here each term calls for evaluating a polynomial that has an arithmetic formula of size $O(\ell \cdot h/d)$. The prover's computation is dominated by computing a sum of $h^{c\ell/d} = 2^{c\ell}$ terms, where each term requires a computation of the type conducted by the verifier. ◀

4.4 Extension to a wider class

As a motivation towards the following extension, we mention the problem of finding a dominating set of constant size t. This problem does not seem to be locally-characterizable in the sense of Definition 2 (cf. [21]), but it definitely resides in the following class.

▶ **Definition 6** (locally ∀∃-characterizable sets)**.** A set S is locally ∀∃-characterizable if there exist constants c, c', a polynomial p and an almost-linear time algorithm that on input 1^n outputs $\mathrm{poly}(\log n)$-sized formulae $\phi_n : [n]^{p(\log n)} \times \{0,1\}^{p(\log n)} \to \{0,1\}$ and $\pi_{n,1}, ..., \pi_{n,p(\log n)} : \{0,1\}^{(c+c')\log n} \to [n]$ such that, for every $x \in \{0,1\}^n$, it holds that $x \in S$ if and only if for all $w \in \{0,1\}^{c \log n}$ there exists $w' \in \{0,1\}^{c' \log n}$ such that

$$\Phi_x(w, w') \stackrel{\text{def}}{=} \phi_n(\pi_{n,1}(w, w'), ..., \pi_{n,p(\log n)}(w, w'), x_{\pi_{n,1}(w,w')}, ..., x_{\pi_{n,p(\log n)}(w,w')}) \tag{11}$$

equals 0.

Like in Definition 2, sometimes one may use a simplified form in which ϕ_n only depends on its $p(\log n)$-bit long suffix. This simplification suffices for the charcaterizing the set of graphs not having a dominating set of size $t = O(1)$. Specifically, when representing n-vertex graphs by their adjacency matrix x (augmented with 1's on the diagonal), the t-dominating set problem is captured by $\Phi_x(i_1, ..., i_t, i_{t+1}) = \vee_{j \in [t]} x_{i_j, i_{t+1}}$ (i.e., x has no t-dominating set iff for every $w = (i_1, ..., i_t) \in [n]^t$ there exists $w' = i_{t+1} \in [n]$ such that $\Phi_x(w, w') = 0$); that is, we used $\phi_{n^2} : \{0,1\}^t \to \{0,1\}$ and $\pi_{n^2,j} : \{0,1\}^{(t+1)\log n} \to [n^2]$ such that $\phi_{n^2}(\sigma_1, ..., \sigma_t) = \vee_j \sigma_j$ and $\pi_{n^2,j}(i_1, ..., i_t, i_{t+1}) = (i_j, i_{t+1})$ for $j \in [t]$.

Every locally ∀∃-characterizable set is in $\mathcal{NC}$ (resp., in $\mathcal{SC}$) and so the existence of a doubly-efficient interactive proof system for it (and its complement) is guaranteed by [15] (resp., [22]). Here we present a direct construction.

▶ **Theorem 7** (main result, extended)**.** *Every locally ∀∃-characterizable set has a simple doubly-efficient interactive proof system. Specifically, on input of length n, the verifier runs in $\widetilde{O}(n)$-time and the strategy of the prescribed prover can be implemented in $\widetilde{O}(n^{c+c'+1})$-time, where $n^{c+c'}$ denotes the number of local conditions in the characterization. Furthermore, the interactive proof system uses public coins and has logarithmic round complexity.*

Note that the complement of every locally ∀∃-characterizable set also has a doubly-efficient interactive proof system. In this proof system, on input $x \in \{0,1\}^n$, letting $\ell = \log n$, the prover finds an adequate $w \in \{0,1\}^{c\ell}$, sends it to the verifier, and the parties engage in a doubly-efficient interactive proof of the residual claim that *for every $w \in \{0,1\}^{c'\ell}$ it holds that $\Phi_{x,w}(w') \stackrel{\text{def}}{=} \Phi_x(w, w')$ evaluates to 0.* (Such a proof system is provided by Theorem 1.)

Proof Sketch. We extend the proof of Theorem 1, as presented in Section 4.2. Specifically, letting $\ell = \log n$, we derive a low degree extension, $\widehat{\Phi}_x : \mathcal{F}^{(c+c')\ell} \to \mathcal{F}$, of Φ_x by following the same steps as in the former proof.[10] Here we have to provide a proof system for establishing that for every w there exists a w' such that $\widehat{\Phi}_x(w, w')$ equals 0, and the problem is converting this claim to one that can be handled by the sum-check protocol. We do so by mimicking the proof of Toda's Theorem [26]. Specifically, for $\ell' = O(\ell)$ and $H \equiv \{0,1\}$, we consider the following arithmetic expression

$$\sum_{w \in H^{c\ell}} r_w \cdot \prod_{i \in [\ell']} \left(1 - \sum_{w' \in H^{c'\ell}} r^{(i)}_{w'} \cdot (1 - \widehat{\Phi}_x(w, w')) \right) \tag{12}$$

where the r_w's and $r^{(i)}_{w'}$'s are selected at random by the verifier at the very beginning of the interaction. Actually, the verifier will select the sequence $(r_w)_{w \in H^{c\ell}}$ from a small biased sample space (over GF(2))[11], and ditto for the sequences $(r^{(i)}_{w'})_{w' \in H^{c'\ell}}$ (which are selected independently for each $i \in [\ell']$). In particular, we shall use $\mathcal{F}$ that is an extension field of GF(2), and view the r_w's (resp., $r^{(i)}_{w'}$'s) as elements of the base field GF(2).

Note that if x does not belong to the set (i.e., $\exists w \forall w' \; \Phi_x(w, w') = 1$), then there exists a w such that for every $i \in [\ell']$ and for every choice of the $r^{(i)}_{w'}$'s it holds that $\Psi^{(i)}_x(w) = 1$, where

$$\Psi^{(i)}_x(z) \stackrel{\text{def}}{=} 1 - \sum_{w' \in H^{c'\ell}} r^{(i)}_{w'} \cdot (1 - \widehat{\Phi}_x(z, w')). \tag{13}$$

Hence, when the r_w's are selected from a small bias set, it holds that

$$\Pr\left[\sum_{w \in H^{c\ell}} r_w \cdot \prod_{i \in [\ell']} \Psi^{(i)}_x(w) = 0 \right] \approx 1/2.$$

On the other hand, if x belongs to the set (i.e., $\forall w \exists w'$ s.t. $\Phi_x(w, w') = 0$), then for every w and i it holds that $\Pr[\Psi^{(i)}_x(w) = 1] < 0.51$, where the probability is taken over the choice of the $r^{(i)}_{w'}$'s (since $(1 - \widehat{\Phi}_x(w, w')) = 1$ for some w' and it follows that $\Pr[\sum_{w' \in H^{c'\ell}} r^{(i)}_{w'} \cdot (1 - \widehat{\Phi}_x(w, w'))) = 0]$ is approximately $1/2$). Hence, $\Pr[\prod_{i \in [\ell']} \Psi^{(i)}_x(w) = 0] < 0.51^{\ell'}$ and so, for every choice of r_w's, it holds that

$$\Pr\left[\sum_{w \in H^{c\ell}} r_w \cdot \prod_{i \in [\ell']} \Psi^{(i)}_x(w) = 0 \right] > 1 - 2^{c\ell} \cdot 0.51^{\ell'} \approx 1,$$

where the probability is taken over the choice of the $r^{(i)}_{w'}$'s (and the approximation assumes $\ell' \gg c \log \ell$ (e.g., $\ell' = 2c\ell$ will do)).

In order to prepare for an application of the sum-check, we need to replace the sequences $(r_w)_{w \in H^{c\ell}}$ and $(r^{(i)}_{w'})_{w' \in H^{c'\ell}}$ (for each $i \in [\ell']$) by the evaluation of low degree polynomials in w (resp., w') (which are defined over $\mathcal{F}^{c\ell}$ (resp., over $\mathcal{F}^{c'\ell}$) and agree with the said sequences on $H^{c\ell}$ (resp., on $H^{c'\ell}$)). (That is, for each fixing of the seed for a small bias generator,

[10] We stress that we use $H \equiv \{0,1\}$ and the corresponding function `EQ` as in Section 4.2 (rather than the settings used in Section 4.3).

[11] See [19] or [12, Sec. 8.5.2]. A seed length of $O(\ell)$ will do.

we consider the function that maps a location in the output sequence to a value of the corresponding bit.) Fortunately, the LFSR construction of [3] is suitable for that purpose, since the j^{th} bit in the corresponding sequence is produced by raising a matrix R to the power j and multiplying the first row of the resulting matrix by a vector s, where R and s are determined by the seed of this pseudorandom generator.[12] Specifically, the j^{th} bit is the top element of the vector $R^j s$, where matrix R and the vector s have dimension that is linear in the seed length (which in turn is logarithmic in the length of the produced sequence). Hence, we may replace r_w, where $w \equiv (w_1, ..., w_{c\ell})$, by a polynomial that computes the top bit of the vector $R^{\sum_{j\in[c\ell]} w_j 2^{j-1}} s$, by precomputing $R_j = R^{2^{j-1}}$ and using

$$R^{\sum_{j\in[c\ell]} w_j 2^{j-1}} = \prod_{j\in[c\ell]} R_j^{w_j} = \prod_{j\in[c\ell]} (w_j R_j + (1-w_j)I),$$

where $I = R^0$ is the identity matrix. Thus, r_w will be replaced by $\widehat{r}(w)$, where $\widehat{r} : \mathcal{F}^{c\ell} \to \mathcal{F}$ is such that $\widehat{r}(z)$ equals the top element of $(\prod_{j\in[c\ell]}(z_j R_j + (1-z_j)I))s$, and ditto for each $(r^{(i)}_{w'})_{w'\in H^{c'\ell}}$ (via the corresponding $\widehat{r}^{(i)} : \mathcal{F}^{c'\ell} \to \mathcal{F}$). We stress that $\widehat{r}$ (resp., $\widehat{r}^{(i)}$) is a polynomial of degree $c\ell$ (resp., $c'\ell$) and it can be evaluated in time $\text{poly}(\ell)$. Hence, the claim that (12) evaluates to 0 can be replaced by the claim

$$\sum_{w\in H^{c\ell}} \widehat{r}(w) \cdot \prod_{i\in[\ell']} \left(1 - \sum_{w'\in H^{c'\ell}} \widehat{r}^{(i)}(w') \cdot (1 - \widehat{\Phi}_x(w,w'))\right) = 0 \tag{14}$$

We outline two ways of handling this claim. The first way consists of invoking the generalized sum-check protocol, which can also handle products, on (14). Pursuing this approach requires identifying $[\ell']$ with $H^{\log \ell'}$ and introducing a low degree polynomial $\widehat{r}' : \mathcal{F}^{(\log \ell')+c'\ell} \to \mathcal{F}$ such that for every $i \in [\ell']$ it holds that $\widehat{r}'(i, z') = \widehat{r}^{(i)}(z')$.

Alternatively, we can apply the sum-check protocol to the claim $\sum_{w\in H^{c\ell}} \widehat{r}(w) \cdot \Psi_x(w) = 0$, where $\Psi_x(w) \stackrel{\text{def}}{=} \prod_{i\in[\ell']} \Psi^{(i)}_x(w)$. This involves $c\ell$ rounds of interactions, and leaves us with verifying a claim of the form $\Psi_x(r) = v$, where $r \in \mathcal{F}^{c\ell}$ and $v \in \mathcal{F}$ are determined by the said execution. At this point, the prover is asked to present the values of $\Psi^{(i)}_x(r)$ for each $i \in [\ell']$, the verifier checks that their products equals v, and the parties involve the sum-check protocol to each of the claimed values. That is, in the i^{th} execution, the prover proves that $1 - \sum_{w'\in H^{c'\ell}} \widehat{r}^{(i)}(w') \cdot (1 - \widehat{\Phi}_x(r, w'))$ equals v_i, where v_i is the value provided for $\Psi^{(i)}_x(r)$. (Note that these ℓ' executions can be performed in parallel.)[13]

This protocol performs $c\ell + c'\ell$ iterations, and the verifier evaluates the residual expression, which (as in the proof of Theorem 1) amounts to evaluating the $\widehat{\pi}_{n,j}$'s and $\widehat{\phi}_n$ as well as computing $\text{poly}(\ell)$ sums of 2^ℓ terms each. The prover's computation is dominated by computing a sum of $2^{c\ell+c'\ell}$ terms, where each term requires a computation of the type conducted by the verifier. ◀

[12] Alternatively, we can use the "powering" (in finite field) construction of [3].

[13] Alternatively, the verifier can select at random $r'_1, .., r'_{\ell'} \in \mathcal{F}$, and ask the prover to prove that $\sum_{i\in[\ell']} r'_i \cdot \Psi^{(i)}_x(r)$ equals $\sum_{i\in[\ell']} r'_i \cdot v_i$. Note that $\sum_{i\in[\ell']} r'_i \cdot \Psi^{(i)}_x(r) = \sum_{i\in[\ell']} r'_i - \sum_{w'\in H^{c'\ell}} \sum_{i\in[\ell']} r'_i \cdot \widehat{r}^{(i)}(w') \cdot (1 - \widehat{\Phi}_x(r, w'))$, so we can apply the sum-check to the outer sum (of $w' \in H^{c'\ell}$) and let the verifier evaluate the residual expression (which has ℓ' terms) by itself.

Digest

The interactive proof presented in the proof of Theorem 7 uses a randomized reduction of evaluating (11) to evaluating (12). In a straightforward implementation, this reduction calls upon the verifier to toss $\text{poly}(n)$ coins and send the outcome to the prover, whereas we aim at verifiers that run in time $\widetilde{O}(n)$. Hence, we use an adequate pseudorandom generator, and let the verifier select a (much shorter) random seed and send it to the prover. For this to work, we need the function that describes the pseudorandom sequence that corresponds to a fixed seed to have low complexity (in an adequate sense). That is, the relevant complexity measure here refers to the function that maps possible locations in a fixed sequence to the value of the corresponding bits, whereas the standard complexity measures refer to the mapping of possible seeds to the value of a fixed location in the corresponding output sequence.

▶ **Remark 8** (beyond $\forall\exists$-characterization)**.** *The notion of a locally $\forall\exists$-characterizable set can be further extended to allow a constant number of* (alternating) *quantifiers; for example, a $\forall\exists\forall$-characterization corresponds to the case that for all $w \in \{0,1\}^{c \log n}$ there exists $w' \in \{0,1\}^{c' \log n}$ such that for all $w'' \in \{0,1\}^{c'' \log n}$ it holds that $\Phi_x(w,w',w'') = 0$. The proof of Theorem 7 extends naturally to that case* (cf. the proof of Toda's Theorem [26]).

Lastly, we note the correspondence between the foregoing local characterizations and the levels of a known hierarchy of *parameterized complexity classes* [11]. In particular, Definition 2 corresponds to a class denoted $\mathcal{W}[1]$, and Definition 6 corresponds to $\mathcal{W}[2]$. (In terms of the $\mathcal{W}$-hierarchy, our definitions are restricted in requiring that the "defining circuits" be more uniform.)

5 The case of t-SUM

For a parameter $t \in \mathbb{N}$, given $(a_1, ..., a_n, b) \in [m]^{n+1}$, the problem is determining whether there exists t indices $i_1, ..., i_t \in [n]$ such that $\sum_{j\in[t]} a_{i_j} = b$. We shall assume, without loss of generality, that $m = \text{poly}(n)$ and that if $\sum_{j\in[t]} a_{i_j} = b$ then $|\{i_j : j \in [t]\}| = t$. These assumptions are justified as follows.

- Given an arbitrary instance $(a_1, ..., a_n, b)$, we consider the instance $(a'_{1,1}, ..., a'_{n,t}, b')$ such that $a'_{i,j} = (t+1)^t \cdot a_i + (t+1)^{j-1}$ and $b' = (t+1)^t \cdot b + \sum_{j\in[t]}(t+1)^{j-1}$. Hence, if $\sum_{(i,j)\in T} a'_{i,j} = b'$ for some $|T| \leq t$, then for every $j \in [t]$ there exists a unique $i \in [n]$ such that $(i,j) \in T$.
- Starting with the case of $m = \exp(\text{poly}(n))$, we reduce to the case of $m = \text{poly}(n)$ as follows. We pick uniformly at random a prime p in $S \stackrel{\text{def}}{=} [O(n^t \cdot \log m)]$ and reduces all integers modulo p.
  Observe that if $\sum_{j\in[t]} a_{i_j} \neq b$, then equality modulo p may hold for at most $\frac{\log tm}{\log\log tm}$ primes $p > \log m$, whereas the number of primes in S is n^t times larger.
  To get back to a problem over the integers (rather than over $\mathbb{Z}_p$), we reduce the modular problem to t instances of the integral problem. Specifically, we use the fact that $\sum_{j\in[t]} a_{i_j} \equiv b \pmod p$ if and only if for some $i \in [t]$ it holds that $\sum_{j\in[t]} a_{i_j} = b + (i-1)\cdot p$.

Our goal is to present an interactive proof for proving that for every $i_1, ..., i_t \in [n]$ it holds that $\sum_{j\in[t]} a_{i_j} \neq b$.

Letting $\chi\colon \mathbb{Z} \to \{0,1\}$ denote the predicate that returns 0 only on 0, we wish to prove that for all $i_1, ..., i_t \in [n]$ it holds that $\chi(b - \sum_{j\in[t]} a_{i_j}) = 1$. Letting B denote the set of primes in $[m']$, where $m' \stackrel{\text{def}}{=} \log(tm)$, we may prove instead that for all $i_1, ..., i_t \in [n]$ it holds that $\prod_{p\in B}\left(1 - \chi\left(b - \sum_{j\in[t]} a_{i_j} \bmod p\right)\right) = 0$, since $|b - \sum_{j\in[t]} a_{i_j}| < tm$ and $\prod_{p\in B} p > tm$.

Letting $[a]_p$ denote the value of a mod p, we can rewrite the above as

$$\prod_{p\in B}\left(1-\chi\left([b]_p-\sum_{j\in[t]}[a_{i_j}]_p \bmod p\right)\right)=0. \tag{15}$$

Observing that $a \stackrel{\text{def}}{=} [b]_p-\sum_{j\in[t]}[a_{i_j}]_p$ resides in $[-tp+t,p-1]$, it follows that $\sum_{i=0}^{t-1}\chi(a+i\cdot p) = t-(1-\chi(a \bmod p))$. Hence, we replace $1-\chi(a \bmod p)$ by $t-\sum_{i=0}^{t-1}\chi(a+i\cdot p)$, and rewrite (15) as

$$\prod_{p\in B}\left(t-\left(\sum_{i=0}^{t-1}\chi\left([b]_p+ip-\sum_{j\in[t]}[a_{i_j}]_p\right)\right)\right)=0. \tag{16}$$

Since the arguments to χ resides in $[-tp+t,tp-1]\subset[-tm'+1,tm'-1]$, reducing it modulo any prime $q>tm'$ does not change the outcome. We shall do so next, while replacing the condition that all (0-1) terms (which correspond to the various $(i_1,...,i_t)\in[n]^t$) evaluate to 1 by the condition that for a random prime $q\in[O(\log n)]$ it holds that

$$\sum_{i_1,...,i_t\in[n]}\ \prod_{p\in B}\left(t-\left(\sum_{i=0}^{t-1}\chi\left([b]_p+ip-\sum_{j\in[t]}[a_{i_j}]_p \bmod q\right)\right)\right)\equiv 0 \pmod q. \tag{17}$$

(Recall that if each of the terms equals 0 then (17) holds, whereas otherwise with high probability over the choice of q (say $q\in[2\log(n^tm),3\log(n^tm)]$) (17) does not holds.) At this point we can implement χ arithmetically (by just raising the argument to power $q-1$). This yields the condition

$$\sum_{i_1,...,i_t\in[n]}\ \prod_{p\in B}\left(t-\left(\sum_{i=0}^{t-1}\left([b]_p+ip-\sum_{j\in[t]}[a_{i_j}]_p \bmod q\right)^{q-1}\right)\right)\equiv 0 \pmod q. \tag{18}$$

Towards applying the sum-check protocol, using $\ell=\log n$ and $\mathcal{F}=\mathrm{GF}(q)$, we define $P:(\mathcal{F}^\ell)^t\to\mathcal{F}$ such that

$$P(z^{(1)},...,z^{(t)})=\prod_{p\in B}\left(t-\left(\sum_{i=0}^{t-1}\left(b'_{p,i}-\sum_{j\in[t]}\sum_{\alpha\in H^\ell}\mathtt{EQ}(\alpha,z^{(j)})\cdot a'_{\alpha,p}\right)^{q-1}\right)\right) \tag{19}$$

where $\{0,1\}\equiv H\subset\mathcal{F}$, $\mathtt{EQ}:\mathcal{F}^\ell\times\mathcal{F}^\ell\to\{0,1\}$ is the indentity indicator (i.e., $\mathtt{EQ}(\overline{\gamma},\overline{z})=\prod_{i\in[\ell]}(z_i\gamma_i+(1-z_i)(1-\gamma_i))$), and $b'_{p,i}=[b]_p+ip$ and $a'_{\alpha,p}=[a_\alpha]_p$.

We wish to use the sum-check protocol in order to verify that $\sum_{\overline{z}\in(H^\ell)^t}P(\overline{z})$ equals 0 mod q, but the problem is that P is a ($t\ell$-variate) polynomial over $\mathcal{F}=\mathrm{GF}(q)$ with individual degree $|B|\cdot(q-1)$. This is a problem because, when running the sum-check protocol, the field size must be larger than the product of the individual degree of the polynomial and the number of variables in the polynomial. The solution is to run the sum-check protocol over an extension field. Specifically, it suffices to use the extension field $\mathcal{K}=\mathcal{F}^3$, since in this case we have $t\ell\cdot(q-1)|B|<q^3/4$, provided that $q\geq\log(n^tm)$ (whereas $|B|<\log(tm)$ and $\ell=\log n$). We thus consider (19) as an expression over $\mathcal{K}$, while noting that its value is in the base field $\mathcal{F}$, and that this value indicates whether the original instance is a YES-instance or a NO-instance (provided that we were not unlucky in our choice of the random prime $q\in[O(\log n)]$).

To wrap-up. The interactive proof starts with the verifier selecting uniformly a random prime $q \in [2\log(n^t m), 3\log(n^t m)]$, and expecting the prover to prove that $\sum_{\overline{z}\in(H^\ell)^t} P(\overline{z}) = 0$, where this expression as well as the definition of P (in (19)) are considered over the extension field $\mathcal{K} = \mathrm{GF}(q)^3 = \mathrm{GF}(q^3)$. The parties then run the sum-check protocol for $t \cdot \ell$ rounds. At the end of the interaction, the verifier evaluates the residual condition (i.e., evaluates P on a single point). Hence, the verifier's computation is dominated by the evaluation of the multi-linear polynomial $\mathtt{EQ}$ on $t \cdot 2^\ell = tn$ points, which means that its complexity is $\widetilde{O}(tn)$. The prover's complexity is $2^{t\ell} = n^t$ times larger.

Digest

One interesting aspect of the foregoing proof system is that it applies to asserting the value of $\sum_{\overline{z}\in H^{t\ell}} P(\overline{z})$, where $P : \mathcal{F}^{t\ell} \to \mathcal{F}$ is a polynomial over $\mathcal{F}$. But since we had no good upper bound on the degree of P, the sum-check was invoked over an extension field of $\mathcal{F}$, denoted $\mathcal{K}$. That is, we actually considered a polynomial over $\mathcal{K}$ that agrees with P on inputs that reside in $\mathcal{F}^{t\ell}$. We note that a similar idea was used by Gur and Raz [17] in their Arthur-Merlin streaming algorithm.

6 Conclusions

Our goal in this work was identifying structures and patterns that facilitate the design of efficient proof systems. Towards this goal, we view the identification of the class of locally-characterizable sets as one of our primary contributions. This is a large and natural class that permits simple and efficient interactive proof systems. Building on the identification of this class and its proof systems, our subsequent work [14], which also builds on the work of [5], shows worst-case to average-case reductions between problems in a closely related (and also natural) class (see the discussion following Theorem 1). We hope that future work will further explore classes and problems that permit efficient proof systems, and that this exploration will contribute to our understanding of these problems' computational complexity.

References

1. Amir Abboud, Arturs Backurs, and Virginia Vassilevska Williams. If the Current Clique Algorithms are Optimal, So is Valiant's Parser. In *46th IEEE Symposium on Foundations of Computer Science*, pages 98–117, 2015.
2. Amir Abboud, Kevin Lewi, and Ryan Williams. Losing Weight by Gaining Edges. In *22nd ESA*, pages 1–12, 2014
3. Noga Alon, Oded Goldreich, Joahn Håstad, and Rene Peralta. Simple Constructions of Almost k-wise Independent Random Variables. *Journal of Random Structures and Algorithms*, Vol. 3, No. 3, pages 289–304, 1992. Preliminary version in *31st FOCS*, 1990.
4. Laszlo Babai. Trading Group Theory for Randomness. In *17th ACM Symposium on the Theory of Computing*, pages 421–429, 1985.
5. Marshall Ball, Alon Rosen, Manuel Sabin and Prashant Nalini Vasudevan. Average-case fine-grained hardness. In *49th ACM Symposium on the Theory of Computing*, pages 483–496, 2017.
6. Marshall Ball, Alon Rosen, Manuel Sabin and Prashant Nalini Vasudevan. Proofs of Useful Work. IACR Cryptology ePrint Archive, Report 2017/203, 2017.
7. Andreas Björklund and Petteri Kaski. How Proofs are Prepared at Camelot. In *Proceedings of the 2016 ACM Symposium on Principles of Distributed Computing*, pages 391–400, 2016.

8 Marco L. Carmosino, Jiawei Gao, Russell Impagliazzo, Ivan Mihajlin, Ramamohan Paturi, and Stefan Schneider. Nondeterministic Extensions of the Strong Exponential Time Hypothesis and Consequences for Non-reducibility. In *2016 ACM Conference on Innovations in Theoretical Computer Science*, pages 261–270, 2016.

9 Jianer Chen, Benny Chor, Mike Fellows, Xiuzhen Huang, David W. Juedes, Iyad A. Kanj, Ge Xia. Tight lower bounds for certain parameterized NP-hard problems. *Inf. Comput.*, Vol. 201 (2), pages 216–231, 2005.

10 Rodney G. Downey and Michael R. Fellows. Fixed-parameter tractability and completeness II: On completeness for W[1]. *Theoretical Computer Science A*, Vol. 141 (1–2), pages 109–131, 1995.

11 Rodney G. Downey and Michael R. Fellows. *Parameterized Complexity*. Springer-Verlag Monographs in Computer Science, 1999.

12 Oded Goldreich. *Computational Complexity: A Conceptual Perspective*. Cambridge University Press, 2008.

13 Oded Goldreich, Silvio Micali, and Avi Wigderson. Proofs that Yield Nothing but their Validity or All Languages in NP Have Zero-Knowledge Proof Systems. *Journal of the ACM*, Vol. 38, No. 3, pages 691–729, 1991. Preliminary version in *27th FOCS*, 1986.

14 Oded Goldreich and Guy N. Rothblum. Worst-case to Average-case reductions for subclasses of P. *ECCC* TR17-130, 2017.

15 Shafi Goldwasser, Yael Tauman Kalai, and Guy N. Rothblum. Delegating Computation: Interactive Proofs for Muggles. *Journal of the ACM*, Vol. 62(4), Art. 27:1-27:64, 2015. Extended abstract in *40th STOC*, pages 113–122, 2008.

16 Shafi Goldwasser, Silvio Micali, and Charles Rackoff. The Knowledge Complexity of Interactive Proof Systems. *SIAM Journal on Computing*, Vol. 18, pages 186–208, 1989. Preliminary version in *17th STOC*, 1985. Earlier versions date to 1982.

17 Tom Gur and Ran Raz. Arthur-Merlin streaming complexity. *Information and Computation*, Vol. 243, pages 145–165, 2015.

18 Carsten Lund, Lance Fortnow, Howard Karloff, and Noam Nisan. Algebraic methods for interactive proof systems. *Journal of the ACM*, Vol. 39, No. 4, pages 859–868, 1992. Extended abstract in *31st FOCS*, 1990.

19 Joseph Naor and Moni Naor. Small-bias Probability Spaces: Efficient Constructions and Applications. *SIAM Journal on Computing*, Vol 22, pages 838–856, 1993. Preliminary version in *22nd STOC*, 1990.

20 Mihai Patrascu. Towards polynomial lower bounds for dynamic problems. In *42nd ACM Symposium on the Theory of Computing*, pages 603–610, 2010.

21 Mihai Patrascu and Ryan Williams. On the Possibility of Faster SAT Algorithms. In *21st SODA*, pages 1065–1075, 2010.

22 Omer Reingold, Guy N. Rothblum, Ron D. Rothblum. Constant-round interactive proofs for delegating computation. In *48th ACM Symposium on the Theory of Computing*, pages 49–62, 2016.

23 Adi Shamir. IP = PSPACE. *Journal of the ACM*, Vol. 39, No. 4, pages 869–877, 1992. Preliminary version in *31st FOCS*, 1990.

24 Madhu Sudan. Invariances in Property Testing. In *Property Testing: Current Research and Surveys*. Springer, Lecture Notes in Computer Science (Vol. 6390), pages 211–227, 2010.

25 Justin Thaler. Semi-Streaming Algorithms for Annotated Graph Streams. In *43rd International Colloquium on Automata, Languages, and Programming*, pages 59:1–59:14, 2016.

26 Seinosuke Toda. PP is as hard as the polynomial-time hierarchy. *SIAM Journal on Computing*, Vol. 20 (5), pages 865–877, 1991.

27 Virginia Vassilevska Williams. Hardness of Easy Problems: Basing Hardness on Popular Conjectures such as the Strong Exponential Time Hypothesis. In *10th International Symposium on Parameterized and Exact Computation*, pages 17–29, 2015.

28 Ryan Williams. Strong ETH Breaks With Merlin and Arthur: Short Non-Interactive Proofs of Batch Evaluation. In *31st Conference on Computational Complexity*, pages 2:1–2:17, 2016.

Appendix: An $\mathcal{MA}$ proof system for locally-chracterizable sets

We present first an $\mathcal{MA}$ proof system of verification complexity $\widetilde{O}(n^{(c+1)/2})$ for every locally-characterizable set, where n denotes the input length and the constant $c \geq 1$ is as in Definition 2. Recall that in the case of t-no-CLIQUE, the input length (for n-vertex graphs) is n^2 and $c = t/2$.

Our starting point is the claim $\sum_{w\in\{0,1\}^{c\ell}} \widehat{\Phi}_x(w) = 0$, where $\widehat{\Phi}_x$ is as in the proof of Theorem 1. Letting $\ell' = (c+1)\ell/2$ and $\ell'' = (c-1)\ell/2$, we write the claim $\sum_{w\in\{0,1\}^{c\ell}} \widehat{\Phi}_x(w) = 0$ as $\sum_{w'\in\{0,1\}^{\ell'}} P(w') = 0$, where

$$P(\overline{z}') = \sum_{w''\in\{0,1\}^{\ell''}} \widehat{\Phi}_x(\overline{z}'w'') \tag{20}$$

The key observation is that P is a multi-variate polynomial of degree $\mathrm{poly}(\ell)$ that can be computed by an arithmetic circuit of size $\widetilde{O}(2^{\ell''+\ell}) = \widetilde{O}(2^{\ell'})$. The size bound is due to summing over $2^{\ell''}$ summands in (20), where the summands are given by Eq. (8)-(9), and each summand is computed using a circuit of size $\widetilde{O}(2^{\ell})$ (the dominant part in computing each summand is computing the terms X_i). Thus, our $\mathcal{MA}$ proof system proceeds as follows.

1. The prover provides the verifier with $v_{w'} \leftarrow P(w')$, for every $w' \in \{0,1\}^{\ell'}$.
2. Using the $\mathcal{MA}$ system for "batch evaluation" of Williams [28], the prover proves to the verifier that $P(w') = v_{w'}$ for every $w' \in \{0,1\}^{\ell'}$.
   Recall that this $\mathcal{MA}$-proof can be verified in time that is almost linear in the sum of the number of evaluation points and the size of the circuit, where in our case each of these quantities is $\widetilde{O}(2^{\ell'})$. (The complexity is also linear in the degree of the computed polynomial, which is our case adds another $\mathrm{poly}(\ell)$ factor, and requires that the field is large enough (which holds too).)
3. Finally, the verifier checks that $\sum_{w'\in\{0,1\}^{\ell'}} v_{w'} = 0$.

Indeed, the non-obvious part is the $\mathcal{MA}$ system for "batch evaluation" of Williams [28], which is employed in Step 2.

Improvement for the case of $\pi_{n,i}$'s that are projections

We say that $\pi : \{0,1\}^{c\ell} \to [n]$ is a projection if there exists an ℓ-subset $I \subseteq [c\ell]$ such that $\pi(w) = w_I$ (where, as usual, $\{0,1\}^{\ell}$ is associated with $[n]$). For $c \geq 2$, in the special case that the $\pi_{n,i}$'s in Definition 2 are projections, we improve the verification time by a $\sqrt{n}$ factor (and the claim regarding t-no-CLIQUE follows). Letting $\ell' = \ell'' = c\ell/2$, observe that the polynomial P of (20) can be written as

$$P(\overline{z}') = \sum_{w''\in\{0,1\}^{\ell''}} Q(\overline{z}'w'', A_1(\overline{z}'w''), ..., A_{p(\ell)}(\overline{z}'w'')), \tag{21}$$

where $Q : \mathcal{F}^{c\ell+p(\ell)\cdot\ell} \to \mathcal{F}$ and $A_1, ..., A_{p(\ell)} : \mathcal{F}^{c\ell} \to \mathcal{F}$ are defined as

$$Q(\overline{z}, \overline{a}) = \widehat{\phi}_n(\widehat{\pi}_{n,1}(\overline{z}), ..., \widehat{\pi}_{n,p(\ell)}(\overline{z}), \overline{a}) \tag{22}$$

$$A_i(\overline{z}) = \sum_{\alpha\in\{0,1\}^\ell} \mathtt{EQ}(\widehat{\pi}_{n,i}(\overline{z}), \alpha) \cdot x_\alpha. \tag{23}$$

Observe that Q is a multi-variate polynomial of degree $\mathrm{poly}(\ell)$ that can be computed by an arithmetic of size $\mathrm{poly}(\ell)$, whereas the A_i's are multi-linear polynomials that can be computed by circuits of size $\widetilde{O}(2^\ell)$. Combining these circuits and summing over all $w'' \in \{0,1\}^{\ell''}$, as done above, yields a circuit of size $\widetilde{O}(2^{\ell''+\ell})$, whereas we aim at a circuit of size $\widetilde{O}(2^{\ell''} + 2^\ell)$. Towards this end, we use the hypothesis that the $\pi_{n,i}$'s are projections. Specifically, denoting the corresponding projections by I_i's, we observe that $A_i(\overline{z})$ actually depends only on $\overline{z}_{I_i}$. Furthermore, letting $I''_i = \{j - \ell' : j \in I_i \setminus [\ell']\}$ and $I'_i = I_i \cap [\ell']$, we can replace $A_i(\overline{z}'w'')$ by $C_{w''_{I''_i},i}(\overline{z}')$, where

$$C_{s,i}(\overline{z}') = \sum_{\alpha\in\{0,1\}^\ell} \mathtt{EQ}(\overline{z}'_{I'_i} s, \alpha) \cdot x_\alpha. \tag{24}$$

Hence, we obtain the circuit

$$P(\overline{z}') = \sum_{w''\in\{0,1\}^{\ell''}} Q(\overline{z}'w'', C_{w''_{I''_1},1}(\overline{z}'), ..., C_{w''_{I''_{p(\ell)}},p(\ell)}(\overline{z}')), \tag{25}$$

which has size $\widetilde{O}(2^{\ell'} + 2^{2\ell})$, where the size bound is due to the number of different circuits $C'_{s,i}$: for each $i \in [p(\ell)]$, there are $2^{|I''_i|} \leq 2^\ell$ possible values for s, and each circuit $C_{s,i}$ has size $\widetilde{O}(2^\ell)$. The key observation here is that the $2^{\ell''}$ terms in the main sum can reuse the values computed by the $\widetilde{O}(2^\ell)$ smaller circuits such that each term is fed by $p(\ell)$ small circuits (which are determined by its identity).

A closer inspection of these smaller circuits allows to upper bound their total size by $\widetilde{O}(2^\ell)$, instead of by $\widetilde{O}(2^{2\ell})$. Specifically, for each $i \in [p(\ell)]$, we have $2^{|I''_i|}$ different circuits but each of these circuits is a multilinear circuit in $|I'_i|$ bits (i.e., the bits $\overline{z}'_{I'_i}$ (see (24))), and so has size $\widetilde{O}(2^{|I'_i|})$. Hence, the circuit captured by (25) has size $\widetilde{O}(2^{\ell'}) + \widetilde{O}(2^\ell) = \widetilde{O}(n^{c/2} + n)$. Applying the foregoing $\mathcal{MA}$ proof system to the circuit captured by (25) (rather than to the circuit captured by (20) and Eq. (8)-(9)), yields a system with verification time $\widetilde{O}(n^{c/2} + n)$.

Zero-Knowledge Proofs of Proximity*†

Itay Berman[1], Ron D. Rothblum[2], and Vinod Vaikuntanathan[3]

1 **MIT, Cambridge MA, USA**
`itayberm@mit.edu`
2 **MIT, Cambridge MA, USA and Northeastern University, Boston MA, USA**
`ronr@mit.edu`
3 **MIT, Cambridge MA, USA**
`vinodv@mit.edu`

Abstract

Interactive proofs of proximity (IPPs) are interactive proofs in which the verifier runs in time *sub-linear* in the input length. Since the verifier cannot even read the entire input, following the property testing literature, we only require that the verifier reject inputs that are *far* from the language (and, as usual, accept inputs that are in the language).

In this work, we initiate the study of *zero-knowledge proofs of proximity* (ZKPP). A ZKPP convinces a sub-linear time verifier that the input is *close* to the language (similarly to an IPP) while simultaneously guaranteeing a natural zero-knowledge property. Specifically, the verifier learns nothing beyond (1) the fact that the input is in the language, and (2) what it could additionally infer by reading a few bits of the input.

Our main focus is the setting of *statistical* zero-knowledge where we show that the following hold *unconditionally* (where N denotes the input length):

- Statistical ZKPPs can be sub-exponentially more efficient than property testers (or even *non-interactive* IPPs): We show a natural property which has a statistical ZKPP with a $\mathsf{polylog}(N)$ time verifier, but requires $\Omega(\sqrt{N})$ queries (and hence also runtime) for every property tester.
- Statistical ZKPPs can be sub-exponentially less efficient than IPPs: We show a property which has an IPP with a $\mathsf{polylog}(N)$ time verifier, but cannot have a statistical ZKPP with even an $N^{o(1)}$ time verifier.
- Statistical ZKPPs for some graph-based properties such as promise versions of expansion and bipartiteness, in the bounded degree graph model, with $\mathsf{polylog}(N)$ time verifiers exist.

Lastly, we also consider the computational setting where we show that:

- Assuming the existence of one-way functions, every language computable either in (logspace uniform) NC or in SC, has a *computational* ZKPP with a (roughly) $\sqrt{N}$ time verifier.
- Assuming the existence of collision-resistant hash functions, every language in NP has a *statistical* zero-knowledge *argument* of proximity with a $\mathsf{polylog}(N)$ time verifier.

1998 ACM Subject Classification F.1.2 Interactive and reactive computation, Probabilistic computation

Keywords and phrases Property Testing, Interactive Proofs, Zero-Knowledge

Digital Object Identifier 10.4230/LIPIcs.ITCS.2018.19

* The first and third author were supported in part by NSF Grants CNS-1350619 and CNS-1414119, Alfred P. Sloan Research Fellowship, Microsoft Faculty Fellowship, the NEC Corporation, a Steven and Renee Finn Career Development Chair from MIT. This work was also sponsored in part by the Defense Advanced Research Projects Agency (DARPA) and the U.S. Army Research Office under contracts W911NF-15-C-0226 and W911NF-15-C-0236.
The second author was partially supported by NSF MACS - CNS-1413920, DARPA IBM - W911NF-15-C-0236, a SIMONS Investigator award Agreement Dated 6-5-12 and by the Cybersecurity and Privacy Institute at Northeastern University

† A full version [10] of the paper is available at `https://eprint.iacr.org/2017/114`

1 Introduction

Interactive proofs, introduced by Goldwasser, Micali and Rackoff [30] are protocols that allow a polynomial-time verifier to check the correctness of a computational statement, typically formulated as membership of an input x in a language $\mathcal{L}$, using an interactive protocol. Interactive proofs have had an incredible impact on theoretical computer science in general, and especially on cryptography and complexity theory.

Given the vast amounts of data that are available nowadays, and the ubiquity of cloud computing, in some applications polynomial-time or even *linear-time* verification may be too slow. A recent line of work, initiated by Rothblum, Vadhan and Wigderson [51] (following the earlier work of Ergün, Kumar and Rubinfeld [16]), asks whether we can construct interactive proofs in which the verifier runs in *sub-linear* time. Since the verifier cannot even read the entire input, we cannot hope to obtain sub-linear time verification in general (even for some very simple computations[1]). Thus, following the property testing literature [52, 21] (see also [20]), the verifier is given oracle access to the input, and soundness is relaxed. Namely, the verifier is only required to reject inputs that are *far* (in Hamming distance) from being in the language. Since the verifier is only assured that the input x is *close* to the language $\mathcal{L}$, these proof-systems are called *interactive proofs of proximity*, or IPPs for short. Recent results ([51, 34, 18, 41, 23, 50, 22, 35]) have demonstrated that many languages admit IPPs with sublinear-time verification.[2]

One of the main features of classical interactive proofs (over their non-interactive counterparts) is that they allow for proving statements in *zero-knowledge* [30, 24]: amazingly, it is possible to prove that $x \in \mathcal{L}$ without revealing anything other than that. Beyond being of intrinsic interest, zero-knowledge proofs have a multitude of applications, especially in cryptography.

In this work, we initiate the study of *zero-knowledge* proofs of proximity, or ZKPP for short. Specifically we ask:

> *Is it possible to prove the correctness of a computation to a verifier that reads only few bits of the input, without revealing any additional "non-local" information about the input?*

By non-local information, we mean any information that cannot be inferred by making only few queries to the input. In particular, and in contrast to the classical zero-knowledge setting, we want our notion of zero-knowledge to capture the fact that the verifier does not even learn the input string itself.

The Model of Zero-Knowledge Proofs of Proximity. As expected, we capture the desired zero-knowledge requirement using the simulation paradigm of [30].

▶ **Definition 1** (ZKPP, informally stated (see Section 2))**.** An IPP with prover $\mathbf{P}$ and verifier $\mathbf{V}$ is a ZKPP, if for any (possibly malicious) verifier $\widehat{\mathbf{V}}$, that given oracle access to input of length N runs in time $t(N) \ll N$, there exists a simulator $\mathbf{S}$ that runs in time roughly $t(N)$

[1] Consider for example verifying whether a given string has parity 0.

[2] Throughout this work we use the verification time as our primary complexity measure for IPPs. We could have alternatively chosen to view the total number of bits observed by the verifier (i.e., those read from the input and those communicated from the prover) as the main resource (note that the verification time is an upper bound on the latter). Focusing on verification time makes our upper bounds stronger, whereas our lower bounds also hold wrt the total number of bits observed by the verifier.

such that for every $x \in \mathcal{L}$ it holds that

$$\mathbf{S}^x \approx \left(\mathbf{P}(x), \widehat{\mathbf{V}}^x\right),$$

where $\left(\mathbf{P}(x), \widehat{\mathbf{V}}^x\right)$ denotes $\widehat{\mathbf{V}}$'s view when interacting with $\mathbf{P}$.

In particular, if the verifier cannot afford to read the entire input, then the simulator must successfully simulate the verifier's view even though it too cannot read the entire input. See Section 2 for the formal definition of the model and additional discussions.

Knowledge Tightness and Simulation Overhead. The above informal definition of zero-knowledge requires that for any possible cheating verifier that runs in (sublinear) time t, there exists a simulator, running in roughly the same time, that simulates the verifier's view. We call the running time of the simulator, viewed as a function of t, the *simulation overhead* of the protocol.[3]

In the zero-knowledge literature, the simulation overhead $s = s(t)$ is typically allowed to be any polynomially-bounded function. This is motivated by the fact that such polynomial-time simulation implies that every *polynomial-time* verifier strategy has a *polynomial-time* simulation.

In contrast, since in our setting of ZKPP the verifier runs in *sub-linear* time, we will sometimes need to be more precise. Suppose for example that we had a ZKPP with a $t = \sqrt{N}$ time verifier (where N is the input length) and with some unspecified polynomial simulation bound $s = s(t)$. In such a case, if for example $s(t) = \Omega(t^2)$, then the simulator would be able to read the *entire* input whereas the verifier clearly cannot. This leads to an undesirable gap between the power of the verifier and that of the simulator.

Thus, to obtain more meaningful results we will sometimes need to precisely specify the simulation overhead that is incurred. Nevertheless, since most (but not all) of our results deal with verifiers that run in poly-logarithmic time, unless we explicitly state otherwise, our default is to allow for *polynomial* simulation overhead. Indeed, in the poly-logarithmic regime, polynomial simulation implies that every poly-logarithmic time verifier strategy has a poly-logarithmic time simulation. In the few cases where we need to be more precise, the simulation overhead will be stated explicitly.

We remark that our quantification of the simulation overhead is closely related to Goldreich's [19, Section 4.4.4.2] notion of knowledge tightness of standard zero-knowledge proofs.

A Cryptographic Motivation from the 90's. Interestingly, the notion of ZKPP has already implicitly appeared in the cryptographic literature 20 years ago. Bellare and Yung [5] noticed that the soundness of the [17] construction of non-interactive zero-knowledge proof-system (NIZK) from trapdoor permutations breaks, if the cheating prover sends a description of a function that is not a permutation. [5] observed that to regain soundness in the [17] protocol, it suffices to verify that the given function is *close* to being a permutation.

Focusing on the case that the domain of the permutation[4] is $\{0,1\}^n$, [5] suggested the following natural non-interactive zero-knowledge proof for certifying that a function is *close*

[3] In our actual definition the simulation overhead may depend also on the input length (and proximity parameter). However, the more fundamental dependence is on the (possibly cheating) verifier's running time. Thus, we omit the dependence on these additional parameters from the current discussion.

[4] We remark that the general case (i.e., when the domain is not $\{0,1\}^n$) introduces significant difficulties. See [28] and [13] for details.

to a permutation: sufficiently many random elements $y_1, \ldots, y_k$ in $\{0,1\}^n$ are specified as part of a common random string[5] (CRS), and the prover is required to provide inverses $x_1, \ldots, x_k$ to all of these elements. Soundness follows from the fact that if the function is far from a permutation then, with high probability, one of these elements will simply not have an inverse. Zero-knowledge is demonstrated by having the simulator sample the x's at random and obtain the y's by evaluating the permutation.

Since the verifier in the [5] protocol is only assured that the function is close to a permutation, in our terminology, the [5] protocol is a non-interactive ZKPP. Notice that the verifier runs in time $\mathsf{poly}(n)$, which is *poly-logarithmic* in the input (i.e., the truth table of f).

1.1 Our Results

In this section we state our main results in an informal manner. See the full version [10] for the formal theorem statements.

As is the case for standard zero-knowledge, the results that we can obtain depend heavily on the specific notion of zero-knowledge. These notions depend on what exactly it means for the output of the simulator to be *indistinguishable* from a real interaction.

The main notion which we focus on in this work is that of *statistical* zero-knowledge proofs of proximity. Here, the requirement is that the distribution of the output of the simulator is statistically close[6] to that of the real interaction.

1.1.1 Statistical ZKPP

The first natural question to ask is whether this notion is meaningful – do there exist statistical ZKPPs?[7] More precisely, since every property tester is by itself a trivial ZKPP (in which the prover sends nothing), we ask whether statistical ZKPPs can outperform property testers.

We answer this question affirmatively. Moreover, we show that same natural problem considered by [5] (i.e., verifying that a function is a permutation) has a very efficient zero-knowledge proof of proximity. We emphasize that, in contrast to the protocol of [5] mentioned above, our protocol is zero-knowledge against arbitrary malicious verifiers (rather than only *honest-verifier* zero-knowledge as in the [5] protocol).

▶ **Theorem 2** (ZKPP for permutations)**.** *Let* $\mathsf{PERMUTATION}_n$ *be the set of all permutations on* n*-bit strings. Then:*

- **ZKPP Upper Bound:** $\mathsf{PERMUTATION}_n$ *has a* 4*-round statistical* ZKPP *in which the verifier runs in* $\mathsf{poly}(n)$ *time.*
- **Property Testing Lower Bound:** *Every tester for* $\mathsf{PERMUTATION}_n$ *must make at least* $\Omega(2^{n/2})$ *queries to the input (and in particular must run in time* $\Omega\left(2^{n/2}\right)$*).*

---

[5] Recall that NIZKs inherently require the use of a CRS.

[6] That is, the two distributions have negligible statistical distance. Negligible here refers to an auxiliary security parameter that is given to all parties, see further discussion in Section 2.2.1.

[7] Note that not every IPP is zero-knowledge. Suppose that we want to check whether a given input consists of two consecutive palindromes (of possibly different lengths) or is far from such. Alon *et al.* [2] showed that every tester for this property must make $\Omega(\sqrt{N})$ queries. However, Fischer *et al.* [18] observed that if the prover provides the index that separates the two palindromes, the property becomes easy to verify. The IPP of [18] is not zero-knowledge since any $o(\sqrt{N})$ time simulator can be transformed into an $o(\sqrt{N})$ time tester for the property, contradicting the [2] lower bound.

(Notice that $\mathsf{poly}(n)$ is poly-logarithmic in the input size, whereas $2^{n/2}$ is roughly the square root of the input size.)

Similarly to other results in the literature on constant-round statistical zero-knowledge (SZK), we can only bound the *expected* running time of our simulator (rather than giving a strict bound that holds with all but negligible probability). Using standard techniques, which introduce a super constant number of rounds, we can obtain a strict bound on the simulator's running time. However, in the interest of simplicity and since it is not our main focus, we avoid doing so. We also remark that Gur and Rothblum [33] give a lower bound on the complexity of *non-interactive* IPPs (i.e., IPP in which the entire interaction consists of a single message from the prover to the verifier, also known as MAPs) for PERMUTATION, and combining their result with ours yields a sub-exponential separation between the power of statistical ZKPP vs. MAPs. (Specifically, [33] show an MAP lower bound of roughly $\Omega(2^{n/4})$ for PERMUTATION.) Lastly, we mention that a variant of the permutation property was used by Aaronson [1] to give an oracle separation of SZK from QMA. However, the SZK protocol that he constructs (which is essentially the [5] protocol) is only *honest-verifier*[8] zero-knowledge.

Beyond the property of being a permutation, we also consider two additional *graph* problems, and show that they admit efficient *honest-verifier* ZKPP protocols. Both problems we consider are in the bounded degree graph model[9], which has been widely studied in the property testing literature [21, 26].

▶ **Theorem 3** (Honest Verifier ZKPP for Expansion and Bipartiteness). *There exist* honest-verifier *statistical* ZKPP *in which the verifier's running time is* $\mathsf{polylog}(N)$*, for input graphs of size* N*, for the following two promise problems:*

1. **Promise Expansion:** *Distinguish graphs with (vertex) expansion* $\alpha \in (0,1]$ *from graphs that are far from even having expansion roughly* $\beta = \alpha^2/\log(N)$*.*
2. **Promise Bipartiteness:** *Distinguish bipartite graphs from graphs that are both rapidly mixing and far from being bipartite.*

A few remarks are in order. We first note that the property testing complexity of both promise problems is known to be $\tilde{\Theta}(\sqrt{N})$ [26, 27, 15, 47, 42]. Second, the IPP for promise-bipartiteness is actually due to [51] and we merely point out that it is an honest-verifier ZKPP. In contrast, the promise-expansion property above was not previously known to admit an (efficient) IPP (let alone a zero-knowledge one). We also remark that both of the problems in Theorem 3 refer to *promise problems*. In particular, we leave open the possibility of a ZKPP for bipartiteness that also handles graphs that are not rapidly mixing, and a ZKPP for expansion that accepts graphs that are α-expanding and rejects graphs that are far from α-expanding (rather than just rejecting those that are far from being $\alpha^2/\log(N)$-expanding as in Theorem 3). Lastly, we also leave open the possibility of extending these protocols to be statistical ZKPP against arbitrary cheating verifiers (rather than just honest verifiers).[10]

[8] In an *honest-verifier* ZKPP, the simulator needs only to output an interaction that is indistinguishable from the interaction of the honest (original) verifier and the prover.

[9] In the bounded degree graph model we assume that the degree of all vertices is bounded by a parameter d and the input graph is represented by an adjacency list. In other words, one can request to see the i-th neighbor (for $i \in [d]$) of some vertex v using a single query.

[10] Since honest-verifier SZK protocols can be converted to be zero-knowledge against arbitrary malicious verifiers ([29], see also [53]), it is reasonable to wonder whether the same holds for statistical ZKPP. We conjecture that this is the case but leave the question of verifying this conjecture to future work.

Limitations of Statistical ZKPP. Given these feasibility results, one may wonder whether it is possible to obtain statistical ZKPP with poly-logarithmic complexity for large complexity classes (e.g., for any language in P), rather than just specific problems as in Theorems 2 and 3. The answer turns out to be negative since Kalai and Rothblum [41] constructed a language, computable in NC_1, for which every IPP (let alone a zero-knowledge one) requires $\Omega(\sqrt{N})$ verification time.[11]

Still, the latter observation raises the question of whether statistical ZKPP are as powerful as IPPs. That is, can every IPP be converted to be statistically zero-knowledge with small overhead? We show that this is not the case:

▶ **Theorem 4** (IPP $\not\subseteq$ SZKPP)**.** *There exists a property* Π *that has an* IPP *in which the verifier runs in* $\mathsf{polylog}(N)$ *time, where* N *is the input length, but* Π *does not have a statistical* ZKPP *in which the verifier runs even in time* $N^{o(1)}$*.*

We emphasize that Theorem 4 is unconditional (i.e., it does not rely on any unproven assumptions as is typically the case when establishing lower bounds in the classical setting). Interestingly, if we do allow for a (reasonable) assumption, we can obtain a stronger separation: namely, of MAP from SZKPP:

▶ **Theorem 5** (MAP $\not\subseteq$ SZKPP)**.** *Assuming suitable circuit lower bounds, there exists a property* Π *that has an* MAP *in which the verifier runs in* $\mathsf{polylog}(N)$ *time, where* N *is the input length, but* Π *does not have a statistical* ZKPP *in which the verifier runs even in time* $N^{o(1)}$*.*

The circuit lower bound that we assume follows from the plausible assumption that the Arthur-Merlin communication complexity of the set disjointness problem is $\Omega(n^{\varepsilon})$, where n is the input length and $\varepsilon > 0$ is some constant.

1.1.2 The Computational Setting

Unsurprisingly, we can obtain much stronger results if we relax some of our requirements to only be *computational* (rather than statistical). Specifically we will consider the following two relaxations:

1. (Computational Zero-Knowledge:) the simulated view is only required to be *computationally* indistinguishable from the real interaction.
2. (Computational Soundness aka Argument-System:) Here, we only require soundness against *efficient* cheating provers.

The following results show that under either one of these relaxations, and assuming reasonable cryptographic assumptions, we can transform many of the known results from the literature of IPPs to be zero-knowledge. Focusing on computational zero-knowledge, we can derive such protocols for any language computable in bounded-depth or in bounded-space, where the verifier runs in roughly $\sqrt{N}$ time.

▶ **Theorem 6** (Computational ZKPP for Bounded Depth)**.** *Assume that there exist one-way functions. Then, every language in logspace-uniform* NC*, has a computational* ZKPP*, where the verifier (and the simulator) run in time* $N^{\frac{1}{2}+o(1)}$ *and the number of rounds is* $\mathsf{polylog}(N)$*. The simulation overhead is roughly linear.*

[11] This still leaves open the possibility that statistical ZKPPs with $O(\sqrt{N})$ complexity exist for large complexity classes. Actually, in the computational setting we show such results, see further discussion in Section 1.1.2.

▶ **Theorem 7** (Computational ZKPP for Bounded Space). *Assume that there exist one-way functions. Then, every language computable in* poly(N)*-time and* $O(N^{\sigma})$*-space, for some sufficiently small constant* $\sigma > 0$*, has a computational* ZKPP*, where the verifier (and the simulator) run in time* $N^{\frac{1}{2}+O(\sigma)}$*. The simulation overhead is roughly linear.*

Note that in both results the simulation overhead is (roughly) linear which means that a verifier running in time t will be simulated in nearly the same time. See additional discussion on the notion of simulation overhead above.

Interestingly, if we only relax to *computational soundness*, we can do even better both in terms of expressive power and the running time of the verifier. The following result gives statistical zero-knowledge arguments of proximity for every language in NP, and with a verifier that runs in only *poly-logarithmic* time.

▶ **Theorem 8** (Statistical Zero-Knowledge Arguments for NP). *Assume that there exist collision-resistant hash functions. Then, every language in* NP*, has a constant-round* statistical *zero-knowledge argument of proximity, where the verifier runs in time* polylog(N)*.*

(Here, since the verifier runs in *poly-logarithmic* time, we can and we do allow for polynomial simulation overhead.)

1.2 Related Works

In this section we discuss some related notions (and results) that have previously appeared in the literature, and how they compare with our results.

Zero-Knowledge PCPs. Zero-knowledge PCPs, introduced by Kilian, Petrank and Tardos [44] (and further studied in [32, 38]), are similar to standard PCPs with an additional zero-knowledge requirement. Namely, the oracle access that the (potentially malicious) verifier has to the PCP should not reveal anything beyond the fact that the input is in the language. Note that the verifier in a zero-knowledge PCP is given full access to the input and oracle access to the proof. In contrast, in zero-knowledge proofs of proximity (studied in this paper) the situation is reversed: the verifier is given oracle access to the input but full access to the communication line with the prover.

A more closely related notion of *zero-knowledge* PCP*s of proximity* was considered by Ishai and Weiss [37]. These are PCP systems in which the verifier gets oracle access to *both* the input and to an alleged PCP-style proof. Similarly to our notion of ZKPP, the verifier runs in sublinear time and is assured (with high probability) that the input is close to the language. ZKPPs and zero-knowledge PCPPs are incomparable — soundness is harder to achieve in the interactive case (since the prover's answers may be adaptive) whereas zero-knowledge is harder to obtain in the PCP setting. Therefore, the difference between our model and that of [37] is that we consider *interactice* proofs, whereas [37] focus on PCP-style proofs: namely soundness is guaranteed only if the PCP proof string is written in advance.

Zero-Knowledge Communication Complexity. A model of zero-knowledge in *communication complexity* was recently proposed by Göös, Pitassi and Watson [31] and further studied by Applebaum and Raykov [3]. Since there are known connections between property testing and communication complexity [11] (which holds also in the interactive setting, see [34]), it is interesting to study whether such a connection can be fruitful also in the zero-knowledge setting. We leave the study of this possibility to future work.

Zero-Knowledge Interactive PCPs and Oracle Proofs. Recent works by Ben-Sasson *et al.* [7, 8] study zero-knowledge interactive oracle proofs – a model in which the verifier receives *oracle* access to the communication tape, but full access to the input.[12] Our model of ZKPP is reversed – the verifier has oracle access to the input but full access to the communication tape. Chiesa *et al.* [14] consider zero-knowledge in the context of interactive PCPs, a model introduced by Kalai and Raz [40].

Measures of Knowledge. The notion of "simulation overhead", similarly to that of "knowledge tightness" [19] mentioned above, can be viewed as a (quantitative) security measure for the zero-knowledge of a protocol. Both notions are *worst-case* and consider the verifier and simulator's running times. Micali and Pass [45] considered a similar measure, but in an *execution-to-execution* setting. Finally, Goldreich and Petrank [25] considered other, incomparable, security measures than the verifier's and simulator's running times.

1.3 Technical Overview

We provide overview for our main conceptual results. Overviews for our other results are given in the appropriate places in the body of the paper.

1.3.1 ZKPP for PERMUTATION (see Theorem 2)

Since it is easier to argue, we begin with showing that any *property tester* for PERMUTATION must make at least $\Omega(\sqrt{N})$ queries, where $N = 2^n$. To see this, consider the following two distributions: (1) a random permutation over $\{0,1\}^n$; and (2) a random function from $\{0,1\}^n$ to $\{0,1\}^n$. The first distribution is supported exclusively on YES instances whereas it can be shown that the second is, with high probability, far from a permutation. However, if a tester makes $q \ll \sqrt{N}$ queries, then in both cases, with high probability, its view will be the same: q distinct random elements. The property testing lower bound follows.

We now turn to show a statistical ZKPP in which the verifier runs in $\mathsf{poly}(n)$ time. Consider the following simple IPP for PERMUTATION (based on the [5] protocol). Given oracle access to a function $f : \{0,1\}^n \to \{0,1\}^n$, the verifier chooses a random $r \in \{0,1\}^n$ and sends r to the prover. The prover computes $z = f^{-1}(r)$ and sends it to the verifier. The verifier checks that indeed $f(z) = r$ and if so accepts.

Clearly if f is a permutation then the verifier in this protocol accepts with probability 1, whereas if f is *far* from a permutation, then with some non-negligible probability the verifier chooses r which does not have a pre-image under f. In such a case the prover cannot make the verifier accept and so the protocol is sound.

It is also not hard to see that this protocol is *honest-verifier* zero-knowledge.[13] However, it is not *cheating-verifier* zero-knowledge: a cheating verifier could learn the inverse of some arbitrary r of its choice.

In order to make the protocol zero-knowledge, intuitively, we would like to have a way for the prover and verifier to jointly sample the element r such that both are assured that it is uniform. For simplicity let us focus on the task of just sampling a single bit σ. The specific properties that we need are

[12] Interactive proofs in which the verifier is not charged for reading the entire communication tape are called either *probabilistically checkable interactive proofs* [50] or *interactive oracle proofs* [9] in the literature.

[13] As a matter of fact, this protocol can be viewed as a non-interactive statistical zero-knowledge protocol for PERMUTATION (and is used as such in [5]).

1. If f is a permutation then the *prover* is assured that σ is random.
2. If f is far from being a permutation then the *verifier* is assured that σ is random.

In fact, the transformation of general honest-verifier statistical zero-knowledge proofs to cheating-verifier ones (see [53, Chapter 6]) implements a sub-routine achieving a generalization of the above task, assuming *full* access to the input. We give a simple solution for our specific case. That is, using only oracle access to a function that is either a permutation or far from any permutation.

We proceed to describe a simple procedure for sampling such a random bit σ. First, the verifier chooses at random $x \in \{0,1\}^n$ and a pairwise independent hash function $h : \{0,1\}^n \to \{0,1\}$ and sends $y = f(x)$ and h to the prover. The prover now chooses a random bit $r \in \{0,1\}$ and sends r to the verifier. The verifier now sends x to the prover who checks that indeed $f(x) = y$. The random bit that they agree on is $\sigma = r \oplus h(x)$.

From the prover's perspective, if f is a permutation then y fully determines x and so r (which is chosen uniformly at random after y is specified) is independent of $h(x)$. Hence, $\sigma = r \oplus h(x)$ is a uniformly random bit. On the other hand, from the verifier's perspective, if f is far from being a permutation, then, intuitively, even conditioned on the value y there still remains some entropy in x (indeed, x is essentially uniform among all the pre-images of y).[14] Now, using a variant of the leftover hash lemma, we can argue that $h(x)$ is close to random. Actually, since the leftover hash lemma implies that pairwise independent hash functions are *strong* extractors, we have that $h(x)$ is close to random even conditioned on h and therefore also conditioned on r (which is a randomized function of h). Thus, we obtain that $\sigma = r \oplus h(x)$ is close to being uniformly random and so our procedure satisfies the desired properties.

A Different Perspective: Instance-Dependent Commitments. *Instance-dependent commitments* [4, 39] are commitment schemes that depend on a specific instance of some underlying language: if the instance is in the language, the commitment is guaranteed to be statistically binding; and if the instance is not in the language the commitment is guaranteed to be statistically hiding. Instance-dependent commitments are a central tool in the study of SZK (e.g., [49, 48, 46]).

We can use PERMUTATION to construct an instance-dependent commitment as follows. Given a function $f : \{0,1\}^n \to \{0,1\}^n$, a commitment to a bit b is a tuple $(f(x), h, h(x) \oplus b)$, for a random $x \in \{0,1\}^n$ and a pairwise independent hash function $h\colon \{0,1\}^n \to \{0,1\}$. Our arguments can be adapted to show that if f is a permutation, then this commitment is statistically binding, whereas if f is *far* from a permutation, then this commitment is (weakly) statistically hiding (to amplify, we can repeat by choosing many x's).

One way to view our protocol for sampling the random string r that was described above, is as an instantiation of Blum's coin flipping protocol [12] based on the foregoing instance-dependent commitment.[15]

[14] Actually, the amount of entropy can be fairly small (and depends on how far f is from being a permutation). To obtain a sufficient amount of entropy, in our actual protocol we generate many such y's.

[15] Recall that in Blum's coin-flipping protocol, one party sends a commitment to a random bit b and the other party replies with another random bit b'. Now, the first party decommits and the parties agree on the bit $b \oplus b'$.

1.3.2 Separating IPP from SZKPP (see Theorem 4)

The proof of Theorem 4 is done in two steps. The first step is to construct a property Π which has an interactive proof of proximity with a large number of rounds and $\mathsf{polylog}(N)$-time verifier, but such that in every 2-*message* interactive proof of proximity for Π, the verifier's running time must be N^{δ}, for some constant $\delta > 0$. Actually, such a result was recently established by Gur and Rothblum [35].

The second step in proving Theorem 4 is a general round reduction transformation for any honest-verifier statistical zero-knowledge proof of proximity. Namely, we would like a procedure that takes any *many-messages* honest-verifier zero-knowledge proof of proximity and turns it into a 2-*message* honest-verifier zero-knowledge proof of proximity while only slightly deteriorating the verifier's and simulator's running times.

To establish such a procedure we apply the proof that the promise problem Entropy Difference (ED) is complete for the class SZK (see [53]). That proof takes an instance x of any promise problem $\Pi = (\Pi_{\mathsf{YES}}, \Pi_{\mathsf{NO}}) \in \mathsf{SZK}$ and efficiently constructs two distributions X and Y such that if $x \in \Pi_{\mathsf{YES}}$ then $\mathrm{H}(X) \geq \mathrm{H}(Y) + 1$, and if $x \in \Pi_{\mathsf{NO}}$ then $\mathrm{H}(Y) \geq \mathrm{H}(X) + 1$. That proof goes on to show a zero-knowledge protocol to distinguish between the case that $\mathrm{H}(X) \geq \mathrm{H}(Y) + 1$ and the case that $\mathrm{H}(Y) \geq \mathrm{H}(X) + 1$. Two important points regarding that proof: (1) sampling from X and Y can be done by running (many times) the simulator for the original problem Π; (2) the protocol for ED consists of only two messages and requires only sample access to X and Y.

In our settings, we can view a property Π as a promise problem where functions possessing the property are in Π_{YES} and functions that are ε-far from possessing the property are in Π_{NO}. Then, we can have the verifier "run" the reduction to ED and apply the sample-access protocol for ED. The unbounded prover will behave as in the protocol for ED. Recall that the original simulator (i.e., the one for the property's IPP) required only oracle access to the input function. Since sampling from the distributions only requires running the original simulator, the new verifier can implement this step with only oracle access to the input function and with only polynomial overhead to the running time of the original simulator.

1.3.3 The Computational Setting (see Theorem 6-8)

The proofs of Theorem 6, Theorem 7 and Theorem 8 rely on the same basic idea: compiling existing public-coin protocols from the literature (specifically those of [51, 50, 43]) that are not zero-knowledge to ones that are. This step is based on the idea, which originates in the work of Ben-Or *et al.* [6], of having the prover commit to its messages rather than sending them in the clear. This ability to commit is where we use the assumption that one-way functions exist.

The compiler, which can only be applied to *public-coin* protocols is as follow. At every round, rather than sending its next message in the clear, the prover merely commits to the message that it would have sent in the protocol. Since the protocol is public-coin, the verifier can continue the interaction even though it does not see the actual contents of the prover's messages. After all commitments have been sent, the verifier only needs to check that there exist suitable decommitments that would have made the underlying IPP verifier accept. Since the commitment hides the contents of the messages, it cannot do so by itself and we would like to use the prover. At this point, one could try to naively argue that the residual statement is an NP statement, and so we can invoke a general purpose zero-knowledge protocol for NP (e.g., the classical [24] protocol or the more efficient [36] protocol).

Herein arises the main difficulty with this approach. While the statement that the verifier needs to check at the end of the interaction does consist of an existential quantifier applied to a polynomial-time computable predicate, the latter predicate makes oracle access to the input

x and so we do not know how to express it as an NP statement. To resolve this difficulty, we restrict our attention to verifiers that make *prover-oblivious queries*; that is, the queries that the verifier makes do not depend on messages sent by the prover. Luckily enough, in the IPPs that we rely on the verifier's queries are indeed prover-oblivious.

Thus, our verifier can actually make its queries after seeing only the commitments and we can construct an NP statement that refers to the actual values that it reads from the input. At this point we can indeed invoke a general purpose zero-knowledge protocol for NP and conclude the proof.

Lastly, we remark that the specific flavor of soundness and zero-knowledge that we obtain depends on the commitment scheme we use and the soundness of the protocol to which we apply the transformation. Loosely speaking, instantiating the above approach with a computationally hiding and statistically binding commitment scheme yields a *computational* zero-knowledge proof of proximity, whereas a statistically hiding and computationally binding one yields a statistical zero-knowledge *argument* of proximity.

Organization. In this extended abstract we include only the formal definitions of the ZKPP model, given in Section 2. See the full version [10] for the formal theorem statements and proofs.

2 ZKPP – Model and Definitions

A ZKPP is an interactive proof for convincing a *sub-linear* time verifier that a given input is close to the language, in *zero-knowledge*. Loosely speaking, by zero-knowledge we mean that if the (N-bit) input is in the language, the view of any (potentially malicious) verifier that runs in time $t \ll N$ can be simulated by reading not much more than t bits from the input.

The only non-trivial step in formalizing this intuition is in quantifying what we mean by "not much more". In the classical setting of zero-knowledge interactive proofs, we merely require that the simulator run in polynomial-time, and so "not much more" is interpreted as polynomial overhead. A natural adaptation for the sub-linear setting would therefore be to require that the running time of the simulator be polynomially related to that of the verifier. However, in some settings this requirement is problematic – e.g., suppose that the verifier runs in time $t = O(\sqrt{N})$. Here, a simulator that runs in time t^2 (and in particular can read the entire input) would be far less meaningful than one running in, say, $t^{3/2}$ time.

Thus, as pointed out in the introduction, it will be important for us to quantify more precisely what is the overhead incurred by the simulator. We refer to this as the *simulation overhead*, which we think of as a function of the verifier's running time (see precise statement below). Thus, rather than merely saying that a protocol is a ZKPP, we will say that it is a ZKPP with simulator overhead s.

We proceed to the formal definitions. A property is an ensemble $\Pi = (\Pi_n, \mathcal{D}_n, \mathcal{R}_n)_{n\in\mathbb{N}}$, where Π_n is a set of functions from $\mathcal{D}_n$ to $\mathcal{R}_n$, for every $n \in \mathbb{N}$. In certain contexts, it will be more convenient for us to view Π_n as a set of strings of length $|\mathcal{D}_n|$ over the alphabet $\mathcal{R}_n$ (in the natural way). In such cases we will also sometimes refer to properties as languages. We denote by N the bit-length of the input, i.e., $N = |\mathcal{D}_n| \cdot \log_2(|\mathcal{R}_n|)$. In the technical sections we will often measure efficiency in terms of the parameter N but in our actual definition below we will allow a direct dependence on n, $|\mathcal{D}_n|$ and $|\mathcal{R}_n|$. This makes the definitions slightly more cumbersome but allows us to capture certain auxiliary parameters that arise in specific models, e.g., the dependence on the degree of the graph in the bounded degree graph model (for details, see Section 2.2.1).

Lastly, similarly to [53], we use a security parameter k to control the quality of our soundness and zero-knowledge guarantees rather than letting these depend on the input length (although our reasons for doing so are slightly different from those in [53], see Section 2.2.1) for additional details).

Section Organization. We begin by recalling the definition of IPPs in Section 2.1, then proceed to define statistical ZKPP in Section 2.2, and finally we discuss computational ZKPP in Section 2.3.

2.1 Interactive Proofs of Proximity (IPPs)

Our definition of IPP follows [51] with minor adaptations.

▶ **Definition 9** (interactive proofs of proximity (IPP))**.** An r-message interactive proof of proximity (IPP), with respect to proximity parameter $\varepsilon > 0$, (in short, ε-IPP) for the property $\Pi = (\Pi_n, \mathcal{D}_n, \mathcal{R}_n)_{n\in\mathbb{N}}$ is an interactive protocol $(\mathbf{P}, \mathbf{V})$ between a prover $\mathbf{P}$, which gets *free* access to an input $f\colon \mathcal{D}_n \to \mathcal{R}_n$ as well as to ε, n, $|\mathcal{D}_n|$, $|\mathcal{R}_n|$ and k, and a verifier $\mathbf{V}$, which gets *oracle* access to f as well as free access to ε, n, $|\mathcal{D}_n|$, $|\mathcal{R}_n|$ and k. The following conditions are satisfied at the end of the protocol for every $k \in \mathbb{N}$ and large enough $n \in \mathbb{N}$:

- ***Completeness***: If $f \in \Pi_n$, then, when $\mathbf{V}$ interacts with $\mathbf{P}$, with probability $1 - \mathsf{negl}(k)$ it accepts.
- ***Soundness***: If f is ε-far from Π_n, then for every prover strategy $\widehat{\mathbf{P}}$, when $\mathbf{V}$ interacts with $\widehat{\mathbf{P}}$, with probability $1 - \mathsf{negl}(k)$ it rejects.

For $t = t(n, |\mathcal{D}_n|, |\mathcal{R}_n|, k, \varepsilon)$, we denote by $\mathsf{IPP}[t]$ the class of properties possessing ε-IPP in which the verifier's running time is at most $O(t)$. Finally, for a class of functions $\mathcal{C}$, we denote by $\mathsf{IPP}[\mathcal{C}(n, |\mathcal{D}_n|, |\mathcal{R}_n|, k, \varepsilon)]$ the class of properties Π for which there exists $t \in \mathcal{C}$ such that $\Pi \in \mathsf{IPP}[t]$.

The probabilities that the verifier rejects in the completeness condition, and accepts in the soundness condition, are called the completeness error and soundness error, respectively. If the completeness error is zero, then we say that the IPP has perfect completeness. A public-coin IPP is an IPP in which every message from the verifier to the prover consists only of fresh random coin tosses and the verifier does toss coins beyond those sent in its messages.

An IPP is said to have query complexity $q = q(n, |\mathcal{D}_n|, |\mathcal{R}_n|, k, \varepsilon) \in \mathbb{N}$ if for every $n, k \in \mathbb{N}$, $\varepsilon > 0$, $f : \mathcal{D}_n \to \mathcal{R}_n$, and any prover strategy $\widehat{\mathbf{P}}$, the verifier $\mathbf{V}$ makes at most $q(n, |\mathcal{D}_n|, |\mathcal{R}_n|, k, \varepsilon)$ queries to f when interacting with $\widehat{\mathbf{P}}$. The IPP is said to have communication complexity $c = c(n, |\mathcal{D}_n|, |\mathcal{R}_n|, k, \varepsilon) \in \mathbb{N}$ if for every $n, k \in \mathbb{N}$, $\varepsilon > 0$, and $f : \mathcal{D}_n \to \mathcal{R}_n$ the communication between $\mathbf{V}$ and $\mathbf{P}$ consists of at most $c(n, |\mathcal{D}_n|, |\mathcal{R}_n|, k, \varepsilon)$ bits.

Our main (but not exclusive) focus in this work is on properties that have IPPs in which the verifier's running time (and thus also the communication and query complexities) is poly-logarithmic in the input size and polynomial in the security parameter k and in the reciprocal of the proximity parameter ε. That is, the class $\mathsf{IPP}[\mathsf{poly}(\log(N), k, 1/\varepsilon)]$.

An IPP that consists of a single message sent from the prover (Merlin) to the verifier (Arthur) is called Merlin-Arthur proof of proximity (MAP) [34]. We extend all the above notations to MAPs in the natural way.

2.2 Statistical ZKPPs

Before defining general ZKPPs, we first consider zero-knowledge with respect to *honest verifiers*. Following [53], we require the simulator to run in strict polynomial-time but allow it to indicate a failure with probability 1/2 (which can then be reduced by repetition). The requirement is that conditioned on not failing, the simulated view is statistically close to the actual execution.

Recall that we say that an algorithm A is useful if $\Pr[\mathsf{A}(x) = \perp] \leq 1/2$ for every input x, and use $\widetilde{\mathsf{A}}(x)$ to denote the output distribution of $\mathsf{A}(x)$, conditioning on $\mathsf{A}(x) \neq \perp$. We define the view of the verifier $\mathbf{V}$ on a common input x (given as standard input or by oracle access to either of the parties) by $\text{view}_{\mathbf{P},\mathbf{V}}(x) \stackrel{def}{=} (m_1, m_2, \ldots, m_r; \rho)$, where $m_1, m_2, \ldots, m_r$ are the messages sent by the parties in a random execution of the protocol, and ρ contains of all the random coins $\mathbf{V}$ used during this execution.

▶ **Definition 10** (honest-verifier zero-knowledge proof of proximity (HVSZKPP, HVPZKPP))**.** Let $(\mathbf{P}, \mathbf{V})$ be an IPP for a property $\Pi = (\Pi_n, \mathcal{D}_n, \mathcal{R}_n)_{n\in\mathbb{N}}$. The protocol $(\mathbf{P}, \mathbf{V})$ is said to be honest-verifier statistical zero-knowledge with simulation overhead s, for some function $s\colon \mathbb{N}^5 \times (0,1] \to \mathbb{N}$ if there exists a useful probabilistic algorithm $\mathbf{S}$, which (like $\mathbf{V}$) gets oracle access to $f\colon \mathcal{D}_n \to \mathcal{R}_n$ as well as free access to ε, n, $|\mathcal{D}_n|$, $|\mathcal{R}_n|$ and k, and whose running time is at most $O(s(t_{\mathbf{V}}, n, |\mathcal{D}_n|, |\mathcal{R}_n|, k, \varepsilon))$, where $t_{\mathbf{V}}(n, |\mathcal{D}_n|, |\mathcal{R}_n|, k, \varepsilon)$ is $\mathbf{V}$'s running time, such that for every $k \in \mathbb{N}$, every large enough $n \in \mathbb{N}$ and $f\colon \mathcal{D}_n \to \mathcal{R}_n$, if $f \in \Pi_n$, it holds that:

$$\text{SD}\left(\widetilde{\mathbf{S}}^f(\varepsilon, n, |\mathcal{D}_n|, |\mathcal{R}_n|, k), \text{view}_{\mathbf{P},\mathbf{V}}(\varepsilon, n, |\mathcal{D}_n|, |\mathcal{R}_n|, k, f)\right) \leq \mathsf{negl}(k).$$

If the $\mathsf{negl}(k)$ can be replaced with 0 in the above equation, $(\mathbf{P}, \mathbf{V})$ is said to be honest-verifier perfect zero-knowledge with simulation overhead s.

For $t = t(n, |\mathcal{D}_n|, |\mathcal{R}_n|, k, \varepsilon)$, $\mathsf{HVSZKPP}[t, s]$ (resp., $\mathsf{HVPZKPP}[t, s]$) denotes the class of properties possessing honest-verifier statistical (resp., perfect) zero-knowledge proof of proximity with simulation overhead s in which the verifier's running time is at most $O(t)$.

We say that the query complexity of a simulator $\mathbf{S}$ is $q' = q'(n, |\mathcal{D}_n|, |\mathcal{R}_n|, k, \varepsilon) \in \mathbb{N}$ if for every $n, k \in \mathbb{N}$, $\varepsilon > 0$, $f : \mathcal{D}_n \to \mathcal{R}_n$, $\mathbf{S}_n^f$ makes at most $q'(n, |\mathcal{D}_n|, |\mathcal{R}_n|, k, \varepsilon)$ queries to f.

A typical setting (that we will focus on) is when the verifier's running time is $\mathsf{poly}(\log(N), k, 1/\varepsilon)$, namely poly-logarithmic in the input length N and polynomial in the security parameter k and in the proximity parameter $1/\varepsilon$. In this setting we often allow for *polynomial* simulation overhead, that is the simulator's running time is also $\mathsf{poly}(\log(N), k, 1/\varepsilon)$. Specifically, we denote by $\mathsf{HVSZKPP}\left[\mathsf{poly}(\log(N), k, 1/\varepsilon)\right]$ the class of properties $\Pi \in \mathsf{HVSZKPP}[t, s]$ for $t = \mathsf{poly}(\log(N), k, 1/\varepsilon)$ and $s = \mathsf{poly}(t, \log(N), k, 1/\varepsilon)$. The class $\mathsf{HVPZKPP}\left[\mathsf{poly}(\log(N), k, 1/\varepsilon)\right]$ is similarly defined.

Another setting of interest is when the verifier's running time is $N^\delta \cdot \mathsf{poly}(k, 1/\varepsilon)$, for some constant $\delta \in (0, 1)$. In this setting, unlike the previous one, allowing the simulation overhead to be polynomial will give the simulator much greater computational power than the verifier (e.g., if $\delta = 1/2$ and s is quadratic in the verifier's running time, then the simulator can run in time $O(N)$ and in particular may read the entire input). In this setting we aim for the simulation overhead to be *linear* in the verifier's running time (but it can be polynomial in k and $1/\varepsilon$).[16]

[16] This requirement is in the spirit of *constant* knowledge tightness, see [19, Section 4.4.4.2].

When the simulation overhead is clear from context we allow ourselves to say that the protocol is a ZKPP (rather than a ZKPP with specific simulation overhead as per Definition 10).

Cheating Verifier ZKPP. We will allow cheating verifiers to be non-uniform by giving them an auxiliary input. For an algorithm A and a string $z \in \{0,1\}^*$ (all auxiliary inputs will be binary strings, regardless of the properties' alphabet), let $\mathsf{A}_{[z]}$ be A when z was given as auxiliary input. Since we care about algorithms whose running time is insufficient to read the entire input, we would not want to allow the running time to depend on the auxiliary input (otherwise, we could artificially inflate z so that A would be able to read the entire input). Thus, following [53], we adopt the convention that the running time of A is independent of z, so if z is too long, A will not be able to access it in its entirety.

▶ **Definition 11** (cheating-verifier zero-knowledge proof of proximity (SZKPP, PZKPP))**.** Let $(\mathbf{P}, \mathbf{V})$ be an interactive proof of proximity for a property $\Pi = (\Pi_n, \mathcal{D}_n, \mathcal{R}_n)_{n\in\mathbb{N}}$. $(\mathbf{P}, \mathbf{V})$ is said to be cheating-verifier statistical zero-knowledge with simulation overhead s, for some function $s\colon \mathbb{N}^5 \times (0,1] \to \mathbb{N}$, if for every algorithm $\widehat{\mathbf{V}}$ whose running time is $O(t_{\widehat{\mathbf{V}}}(n, |\mathcal{D}_n|, |\mathcal{R}_n|, k, \varepsilon))$, there exists a useful probabilistic algorithm $\mathbf{S}$, which (like $\widehat{\mathbf{V}}$) gets oracle access to $f\colon \mathcal{D}_n \to \mathcal{R}_n$ as well as free access to ε, n, $|\mathcal{D}_n|$, $|\mathcal{R}_n|$ and k, and whose running time is at most $O(s(t_{\widehat{\mathbf{V}}}, n, |\mathcal{D}_n|, |\mathcal{R}_n|, k, \varepsilon))$, such that for every $k \in \mathbb{N}$, large enough $n \in \mathbb{N}$, $z \in \{0,1\}^*$ and $f\colon \mathcal{D}_n \to \mathcal{R}_n$, if $f \in \Pi_n$, then

$$\mathrm{SD}\left(\widetilde{\mathbf{S}}^f_{[z]}(\varepsilon, n, |\mathcal{D}_n|, |\mathcal{R}_n|, k), \mathrm{view}_{\mathbf{P},\widehat{\mathbf{V}}_{[z]}}(\varepsilon, n, |\mathcal{D}_n|, |\mathcal{R}_n|, k, f)\right) \leq \mathsf{negl}(k).$$

If the $\mathsf{negl}(k)$ can be replaced with 0 in the above equation, $(\mathbf{P}, \mathbf{V})$ is said to be a cheating-verifier perfect zero-knowledge with simulation overhead s.

For $t = t(n, |\mathcal{D}_n|, |\mathcal{R}_n|, k, \varepsilon)$, $\mathsf{SZKPP}[t, s]$ (resp., $\mathsf{PZKPP}[t, s]$) denotes the class of properties possessing cheating-verifier statistical (resp., perfect) zero-knowledge proof of proximity with simulation overhead s in which the verifier's running time is at most $O(t)$.

Expected Simulation Overhead. Definition 11 requires that the running time of the simulator always be bounded. Similarly to many results in the ZK literature, in some cases we can only bound the simulator's *expected* running time. The following definition captures this (weaker) notion:

▶ **Definition 12** (cheating-verifier ZKPP with expected simulation (ESZKPP, EPZKPP))**.** Let $(\mathbf{P}, \mathbf{V})$ be an interactive proof of proximity for a property $\Pi = (\Pi_n, \mathcal{D}_n, \mathcal{R}_n)_{n\in\mathbb{N}}$. The protocol $(\mathbf{P}, \mathbf{V})$ is said to be cheating-verifier statistical zero-knowledge with expected simulation overhead s if it satisfies the same requirement as in Definition 11 except that we only bound the *expected* running time of the simulator (where the expectation is over the coins of the simulator).

The classes $\mathsf{ESZKPP}[t, s]$ and $\mathsf{EPZKPP}[t, s]$ are defined analogous to $\mathsf{SZKPP}[t, s]$ and $\mathsf{PZKPP}[t, s]$ from Definition 11.

Unless explicitly saying otherwise, all zero-knowledge protocols we discuss are cheating-verifier ones.

As in the honest-verifier case, a typical setting is that in which the verifier's running time is poly-logarithmic in the input size N and polynomial in the security parameter k and in $1/\varepsilon$, and the simulator's (possibly only expected and not strict) running time is polynomial in the running time of the cheating-verifier that it simulates, poly-logarithmic in N and

polynomial in k and $1/\varepsilon$. Specifically, if we allow the cheating-verifier the same computational powers as the honest-verifier, then both the honest-verifier and every simulator run in time $\mathsf{poly}(\log(N), k, 1/\varepsilon)$. We let $\mathsf{ESZKPP}\big[\mathsf{poly}(\log(N), k, 1/\varepsilon)\big]$ be the class of properties $\Pi \in \mathsf{ESZKPP}[t, s]$ for $t = \mathsf{poly}(\log(N), k, 1/\varepsilon)$ and $s = \mathsf{poly}(t_{\widehat{\mathbf{V}}}, \log(N), k, 1/\varepsilon)$. The class $\mathsf{EPZKPP}\big[\mathsf{poly}(\log(N), k, 1/\varepsilon), \mathsf{poly}\big]$ is similarly defined.

2.2.1 Additional Discussions

We conclude Section 2.2 with a few remarks on statistical ZKPP.

▶ Remark (Proximity Promise Problems). Some of the protocols that we construct do not refer to a property but rather to a "proximity promise problem". Recall that a promise problem considers a pair of disjoint sets Π_{YES} and Π_{NO} and the goal is to distinguish input that are in Π_{YES} from those that are in Π_{NO} (and no requirement is given for inputs outside of $\Pi_{\mathsf{YES}} \cup \Pi_{\mathsf{NO}}$).

For some of our results we will consider *proximity* promise problems, which are also characterized by sets Π_{YES} and a family of sets $\left(\Pi_{\mathsf{NO}}^{(\varepsilon)}\right)_{\varepsilon\in(0,1)}$, and we require that for every $\varepsilon \in (0,1)$, it holds that $\Pi_{\mathsf{NO}}^{(\varepsilon)}$ is ε-*far* from Π_{YES} (rather than merely being disjoint). We extend the definitions above to handle proximity promise problems in the natural way (specifically, completeness and zero-knowledge should only hold for input in Π_{YES} whereas the soundness requirement is that if the verifier is given proximity parameter $\varepsilon > 0$ and an input in $\Pi_{\mathsf{NO}}^{(\varepsilon)}$, then it should reject with high probability).

▶ Remark (The Security Parameter). One of the original motivations for the introduction of a security parameter in the classical definitions of statistical zero-knowledge proofs was to control the error parameters (completeness, soundness and simulation deviation) independently from the input's length. Specifically, one may want to provide a high-quality proof (i.e., very small errors) for short inputs (see [53, Section 2.4]).

In our setting, the situation is somewhat reversed. We think of very large inputs that the verifier and simulator cannot even entirely read. Hence, it seems unreasonable to require errors that are negligible in the input length. Instead, we control the quality of the proof with the security parameter, independent of the input length.

▶ Remark (A Definitional Convention). Traditionally [21], a property tester gets an oracle access to a Boolean function $f\colon \{0,1\}^n \to \{0,1\}$ and needs to determined if the function has the property or is ε-far from having this property (i.e., ε-far from any function that has the property). The tester gets n (or alternatively 2^n — the input length of the truth table of f) as a standard input and its complexity (e.g., running time, number of oracle queries) is measured as a function of n. As models and properties evolved (e.g., the bounded degree model [26]) Boolean functions no longer sufficed to (conveniently) describe properties. For example, in the bounded degree model graphs with n vertices and degree d are specified as a functions $G\colon [n] \times [d] \to [n] \cup \{\perp\}$ such that $G(u,i) = v$ if v is the i'th neighbor of a vertex u and $G(u,i) = \perp$ if u has less than i neighbors. Consequently, the parameter n alone no longer suffices to measure the complexity of the tester.

The situation becomes even more delicate when interaction is added. The model of *interactive proofs of proximity* (IPP), introduced by [51], considers an interaction between a prover and a verifier in which the prover is trying to convince the verifier that a function has a property. In the definition of [51], in addition to the function f, to which the verifier has only oracle access and is referred to as the *implicit* input, the verifier also has full access to an additional (shorter) input w, called the *explicit* input. For example, in the bounded degree

model w might be simply d, and in "algebraic" properties w can contain a description of some underlying field. Roughly speaking, [51] chose to measure the complexity of the proof-system with respect to the length of the implicit input alone. This creates a slight inconvenience when trying to describe complexity measures. For example, in the bounded degree model, we would like the running time of the verifier to explicitly depend on the number of vertices and the degree d. However, as it is defined in [51], the function that bounds the verifier's running time gets only the input length (which has bit length $n \cdot d \cdot \log(n)$).

To avoid this minor issue, in this paper, we take a slightly different approach than that of [51] when defining IPPs. Our goal is to define a general model in which it is easy to compare properties from different domains (e.g., properties of bounded degree graphs and those considering algebra). To do so we no longer split the input to an implicit and explicit parts. We consider functions $f\colon \mathcal{D} \to \mathcal{R}$ from an arbitrary domain $\mathcal{D}$ to an arbitrary range $\mathcal{R}$. The verifier receives oracle access to f, and full access to $|\mathcal{D}|$ and $|\mathcal{R}|$. The prover receives full access the function f. Different complexity measures are now functions of the verifier's standard inputs — $|\mathcal{D}|$ and $|\mathcal{R}|$. Going back to the bounded degree graph example, we can see in this framework the function describing the verifier's running time gets $|\mathcal{D}_n| = n \cdot d$ and $|\mathcal{R}_n| = n$ as inputs, and can be easily "converted" into a function that simply gets n and d as inputs. Moreover, we can now define N to be the input length of the property (i.e., $N = |\mathcal{D}| \cdot \log(|\mathcal{R}|)$) and define complexity classes with respect to this input length. For example, we can define $\mathsf{IPP}[\mathsf{poly}(\log(N))]$ to be the class containing all properties with interactive proof of proximity in which the verifier's running time (as a function of $|\mathcal{D}|$ and $|\mathcal{R}|$) is bounded by $\mathsf{poly}(\log(N))$. Note that the above class does not depend on the domain and range of the propety, and properties of different "types" can still belong to $\mathsf{IPP}[\mathsf{poly}(\log(N))]$.

2.3 Computational ZKPP

Since our focus is on the statistical case, we do not provide explicit definitions of computational zero-knowledge proofs of proximity. Rather, these definitions can be easily extrapolated from the statistical ones in a standard way (see for example Vadhan's [53, Section 2] definition of computational zero-knowledge). Specifically, in the computational definitions one simply requires that the simulator's output and the protocol's view are computationally indistinguishable (rather than statistically close), with respect to the security parameter.

Acknowledgements. We thank Oded Goldreich for useful discussions and for his comments on an earlier version of this paper. We thank Omer Paneth for useful discussions and Prashant Vasudevan for referring us to [1].

References

1 Scott Aaronson. Impossibility of succinct quantum proofs for collision-freeness. *Quantum Information & Computation*, 12(1-2):21–28, 2012. URL: `http://www.rintonpress.com/xxqic12/qic-12-12/0021-0028.pdf`.

2 Noga Alon, Michael Krivelevich, Ilan Newman, and Mario Szegedy. Regular languages are testable with a constant number of queries. *SIAM J. Comput.*, 30(6):1842–1862, 2000. `doi:10.1137/S0097539700366528`.

3 Benny Applebaum and Pavel Raykov. From private simultaneous messages to zero-information arthur-merlin protocols and back. In Eyal Kushilevitz and Tal Malkin, editors, *Theory of Cryptography - 13th International Conference, TCC 2016-A, Tel Aviv, Israel,*

January 10-13, 2016, Proceedings, Part II, volume 9563 of *Lecture Notes in Computer Science*, pages 65–82. Springer, 2016. doi:10.1007/978-3-662-49099-0_3.

4 Mihir Bellare, Silvio Micali, and Rafail Ostrovsky. Perfect zero-knowledge in constant rounds. In *Proceedings of the 22nd Annual ACM Symposium on Theory of Computing (STOC)*, pages 482–493. ACM Press, 1990.

5 Mihir Bellare and Moti Yung. Certifying permutations: Noninteractive zero-knowledge based on any trapdoor permutation. *J. Cryptology*, 9(3):149–166, 1996.

6 Michael Ben-Or, Oded Goldreich, Shafi Goldwasser, Johan Håstad, Joe Kilian, Silvio Micali, and Phillip Rogaway. Everything provable is provable in zero-knowledge. In Shafi Goldwasser, editor, *Advances in Cryptology - CRYPTO '88, 8th Annual International Cryptology Conference, Santa Barbara, California, USA, August 21-25, 1988, Proceedings*, volume 403 of *Lecture Notes in Computer Science*, pages 37–56. Springer, 1988. doi:10.1007/0-387-34799-2_4.

7 Eli Ben-Sasson, Alessandro Chiesa, Michael A. Forbes, Ariel Gabizon, Michael Riabzev, and Nicholas Spooner. On probabilistic checking in perfect zero knowledge. *IACR Cryptology ePrint Archive*, 2016:988, 2016. URL: http://eprint.iacr.org/2016/988.

8 Eli Ben-Sasson, Alessandro Chiesa, Ariel Gabizon, and Madars Virza. Quasi-linear size zero knowledge from linear-algebraic pcps. In Eyal Kushilevitz and Tal Malkin, editors, *Theory of Cryptography - 13th International Conference, TCC 2016-A, Tel Aviv, Israel, January 10-13, 2016, Proceedings, Part II*, volume 9563 of *Lecture Notes in Computer Science*, pages 33–64. Springer, 2016. doi:10.1007/978-3-662-49099-0_2.

9 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.

10 Itay Berman, Ron D. Rothblum, and Vinod Vaikuntanathan. Zero-knowledge proofs of proximity. *IACR Cryptology ePrint Archive*, 2017:114, 2017. URL: http://eprint.iacr.org/2017/114.

11 Eric Blais, Joshua Brody, and Kevin Matulef. Property testing lower bounds via communication complexity. *Computational Complexity*, 21(2):311–358, 2012. doi:10.1007/s00037-012-0040-x.

12 Manuel Blum. Coin flipping by telephone. In *Advances in Cryptology – CRYPTO'81*, pages 11–15, 1981.

13 Ran Canetti and Amit Lichtenberg, 2017. Unpublished manuscript.

14 Alessandro Chiesa, Michael A. Forbes, and Nicholas Spooner. A zero knowledge sumcheck and its applications. *Electronic Colloquium on Computational Complexity (ECCC)*, 24:57, 2017. URL: https://eccc.weizmann.ac.il/report/2017/057.

15 Artur Czumaj and Christian Sohler. Testing expansion in bounded-degree graphs. *Combinatorics, Probability & Computing*, 19(5-6):693–709, 2010. doi:10.1017/S096354831000012X.

16 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.

17 Uriel Feige, Dror Lapidot, and Adi Shamir. Multiple non-interactive zero knowledge proofs under general assumptions. *sicomp*, 1999. Preliminary version in *FOCS'90*.

18 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.

19 Oded Goldreich. *Foundations of Cryptography: Basic Tools*. Cambridge University Press, 2001.

20 Oded Goldreich. *Introduction to Property Testing*. forthcoming (`http://www.wisdom.weizmann.ac.il/~oded/pt-intro.html`), 2016.

21 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`.

22 Oded Goldreich and Tom Gur. Universal locally testable codes. *Electronic Colloquium on Computational Complexity (ECCC)*, 23:42, 2016. URL: `http://eccc.hpi-web.de/report/2016/042`.

23 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`.

24 Oded Goldreich, Silvio Micali, and Avi Wigderson. Proofs that yield nothing but their validity or all languages in NP have zero-knowledge proof systems. *Journal of the ACM*, pages 691–729, 1991. Preliminary version in *FOCS'86*.

25 Oded Goldreich and Erez Petrank. Quantifying knowledge complexity. *Computational Complexity*, 8(1):50–98, 1999. `doi:10.1007/s000370050019`.

26 Oded Goldreich and Dana Ron. Property testing in bounded degree graphs. *Algorithmica*, 32(2):302–343, 2002. `doi:10.1007/s00453-001-0078-7`.

27 Oded Goldreich and Dana Ron. On testing expansion in bounded-degree graphs. In Oded Goldreich, editor, *Studies in Complexity and Cryptography. Miscellanea on the Interplay between Randomness and Computation - In Collaboration with Lidor Avigad, Mihir Bellare, Zvika Brakerski, Shafi Goldwasser, Shai Halevi, Tali Kaufman, Leonid Levin, Noam Nisan, Dana Ron, Madhu Sudan, Luca Trevisan, Salil Vadhan, Avi Wigderson, David Zuckerman*, volume 6650 of *Lecture Notes in Computer Science*, pages 68–75. Springer, 2011. `doi:10.1007/978-3-642-22670-0_9`.

28 Oded Goldreich and Ron D. Rothblum. Enhancements of trapdoor permutations. *J. Cryptology*, 26(3):484–512, 2013. `doi:10.1007/s00145-012-9131-8`.

29 Oded Goldreich, Amit Sahai, and Salil P. Vadhan. Honest-verifier statistical zero-knowledge equals general statistical zero-knowledge. 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 399–408. ACM, 1998. `doi:10.1145/276698.276852`.

30 Shafi Goldwasser, Silvio Micali, and Charles Rackoff. The knowledge complexity of interactive proof systems. *sicomp*, pages 186–208, 1989. Preliminary version in *STOC'85*.

31 Mika Göös, Toniann Pitassi, and Thomas Watson. Zero-information protocols and unambiguity in arthur-merlin communication. *Algorithmica*, 76(3):684–719, 2016. `doi:10.1007/s00453-015-0104-9`.

32 Vipul Goyal, Yuval Ishai, Mohammad Mahmoody, and Amit Sahai. Interactive locking, zero-knowledge pcps, and unconditional cryptography. In Tal Rabin, editor, *Advances in Cryptology - CRYPTO 2010, 30th Annual Cryptology Conference, Santa Barbara, CA, USA, August 15-19, 2010. Proceedings*, volume 6223 of *Lecture Notes in Computer Science*, pages 173–190. Springer, 2010. `doi:10.1007/978-3-642-14623-7_10`.

33 Tom Gur and Ron D. Rothblum, 2015. Unpublished observation.

34 Tom Gur and Ron D. Rothblum. Non-interactive proofs of proximity. *Computational Complexity*, pages 1–109, 2016. `doi:10.1007/s00037-016-0136-9`.

35 Tom Gur and Ron D. Rothblum. A hierarchy theorem for interactive proofs of proximity. In *Proceedings of the 2017 ACM Conference on Innovations in Theoretical Computer Science, Berkeley, CA, USA, January 9-11, 2016*, 2017.

36 Yuval Ishai, Eyal Kushilevitz, Rafail Ostrovsky, and Amit Sahai. Zero-knowledge proofs from secure multiparty computation. *SIAM J. Comput.*, 39(3):1121–1152, 2009. doi:10.1137/080725398.

37 Yuval Ishai and Mor Weiss. Probabilistically checkable proofs of proximity with zero-knowledge. In Yehuda Lindell, editor, *Theory of Cryptography - 11th Theory of Cryptography Conference, TCC 2014, San Diego, CA, USA, February 24-26, 2014. Proceedings*, volume 8349 of *Lecture Notes in Computer Science*, pages 121–145. Springer, 2014. doi:10.1007/978-3-642-54242-8_6.

38 Yuval Ishai, Mor Weiss, and Guang Yang. Making the best of a leaky situation: Zero-knowledge pcps from leakage-resilient circuits. In Eyal Kushilevitz and Tal Malkin, editors, *Theory of Cryptography - 13th International Conference, TCC 2016-A, Tel Aviv, Israel, January 10-13, 2016, Proceedings, Part II*, volume 9563 of *Lecture Notes in Computer Science*, pages 3–32. Springer, 2016. doi:10.1007/978-3-662-49099-0_1.

39 Toshiya Itoh, Yuji Ohta, and Hiroki Shizuya. A language-dependent cryptographic primitive. *Journal of Cryptology*, pages 37–49, 1997.

40 Yael Tauman Kalai and Ran Raz. Interactive PCP. In Luca Aceto, Ivan Damgård, Leslie Ann Goldberg, Magnús M. Halldórsson, Anna Ingólfsdóttir, and Igor Walukiewicz, editors, *Automata, Languages and Programming, 35th International Colloquium, ICALP 2008, Reykjavik, Iceland, July 7-11, 2008, Proceedings, Part II - Track B: Logic, Semantics, and Theory of Programming & Track C: Security and Cryptography Foundations*, volume 5126 of *Lecture Notes in Computer Science*, pages 536–547. Springer, 2008. doi:10.1007/978-3-540-70583-3_44.

41 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.

42 Satyen Kale and C. Seshadhri. An expansion tester for bounded degree graphs. *SIAM J. Comput.*, 40(3):709–720, 2011. doi:10.1137/100802980.

43 Joe Kilian. A note on efficient zero-knowledge proofs and arguments (extended abstract). In *Proceedings of the 24th Annual ACM Symposium on Theory of Computing (STOC)*, pages 723–732, 1992.

44 Joe Kilian, Erez Petrank, and Gábor Tardos. Probabilistically checkable proofs with zero knowledge. 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 496–505. ACM, 1997. doi:10.1145/258533.258643.

45 Silvio Micali and Rafael Pass. Precise zero knowledge. http://www.cs.cornell.edu/~rafael/papers/preciseZK.pdf, 2007.

46 Daniele Micciancio and Salil Vadhan. Statistical zero-knowledge proofs with efficient provers: lattice problems and more. In *crypto03*, pages 282–298, 2003.

47 Asaf Nachmias and Asaf Shapira. Testing the expansion of a graph. *Inf. Comput.*, 208(4):309–314, 2010. doi:10.1016/j.ic.2009.09.002.

48 Minh-Huyen Nguyen and Salil Vadhan. Zero knowledge with efficient provers. In *stoc38*, pages 287–295, 2006.

49 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 Con-*

ference, 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.

50 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.

51 Guy N. Rothblum, Salil P. Vadhan, and Avi Wigderson. Interactive proofs of proximity: delegating computation in sublinear time. In *Symposium on Theory of Computing Conference, STOC'13, Palo Alto, CA, USA, June 1-4, 2013*, pages 793–802, 2013.

52 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.

53 Salil P. Vadhan. *A Study of Statistical Zero-Knowledge Proofs*. PhD thesis, Massachusetts Institute of Technology, Cambridge, MA, USA, 1999.

Minimum Circuit Size, Graph Isomorphism, and Related Problems*

Eric Allender[†1], Joshua A. Grochow[‡2], Dieter van Melkebeek[§3], Cristopher Moore[4], and Andrew Morgan[¶5]

1 **Rutgers University, Piscataway, NJ, USA**
allender@cs.rutgers.edu

2 **University of Colorado at Boulder, Boulder, CO, USA**
joshua.grochow@colorado.edu

3 **University of Wisconsin–Madison, Madison, WI, USA**
dieter@cs.wisc.edu

4 **Santa Fe Institute, Santa Fe, NM, USA**
moore@santafe.edu

5 **University of Wisconsin–Madison, Madison, WI, USA**
amorgan@cs.wisc.edu

Abstract

We study the computational power of deciding whether a given truth-table can be described by a circuit of a given size (the Minimum Circuit Size Problem, or MCSP for short), and of the variant denoted MKTP where circuit size is replaced by a polynomially-related Kolmogorov measure. All prior reductions from supposedly-intractable problems to MCSP / MKTP hinged on the power of MCSP / MKTP to distinguish random distributions from distributions produced by hardness-based pseudorandom generator constructions. We develop a fundamentally different approach inspired by the well-known interactive proof system for the complement of Graph Isomorphism (GI). It yields a randomized reduction with zero-sided error from GI to MKTP. We generalize the result and show that GI can be replaced by any isomorphism problem for which the underlying group satisfies some elementary properties. Instantiations include Linear Code Equivalence, Permutation Group Conjugacy, and Matrix Subspace Conjugacy. Along the way we develop encodings of isomorphism classes that are efficiently decodable and achieve compression that is at or near the information-theoretic optimum; those encodings may be of independent interest.

1998 ACM Subject Classification F.1.1 Models of Computation, F.1.2 Modes of Computation, F.1.3 Complexity Measures and Classes

Keywords and phrases Reductions between NP-intermediate problems, Graph Isomorphism, Minimum Circuit Size Problem, time-bounded Kolmogorov complexity

Digital Object Identifier 10.4230/LIPIcs.ITCS.2018.20

* A full version of this paper is available as [3], https://arxiv.org/abs/1710.09806.
† EA acknowledges the support of National Science Foundation grant CCF-1555409.
‡ JAG was supported by an Omidyar Fellowship from the Santa Fe Institute and National Science Foundation grant DMS-1620484.
§ DvM acknowledges the support of National Science Foundation grant CCF-1319822.
¶ AM acknowledges the support of National Science Foundation grant CCF-1319822.

1 Introduction

Finding a circuit of minimum size that computes a given Boolean function constitutes the overarching goal in nonuniform complexity theory. It defines an interesting computational problem in its own right, the complexity of which depends on the way the Boolean function is specified. A generic and natural, albeit verbose, way to specify a Boolean function is via its truth-table. The corresponding decision problem is known as the Minimum Circuit Size Problem (MCSP): Given a truth-table and a threshold θ, does there exist a Boolean circuit of size at most θ that computes the Boolean function specified by the truth-table? The interest in MCSP dates back to the dawn of theoretical computer science [30]. It continues today partly due to the fundamental nature of the problem, and partly because of the work on natural proofs and the connections between pseudorandomness and computational hardness.

A closely related problem from Kolmogorov complexity theory is the Minimum KT Problem (MKTP), which deals with compression in the form of efficient programs instead of circuits. Rather than asking if the input has a small circuit when interpreted as the truth-table of a Boolean function, MKTP asks if the input has a small program that produces each individual bit of the input quickly. To be more specific, let us fix a universal Turing machine U. We consider descriptions of the input string x in the form of a program d such that, for every bit position i, U on input d and i outputs the i-th bit of x in T steps. The KT cost of such a description is defined as $|d| + T$, i.e., the bit-length of the program plus the running time. The KT complexity of x, denoted $\mathrm{KT}(x)$, is the minimum KT cost of a description of x. $\mathrm{KT}(x)$ is polynomially related to the circuit complexity of x when viewed as a truth-table (see Section 4.1 for a more formal treatment). On input a string x and an integer θ, MKTP asks whether $\mathrm{KT}(x) \leq \theta$.

Both MCSP and MKTP are in NP but are not known to be in P or NP-complete. As such, they are two prominent candidates for NP-intermediate status. Others include factoring integers, discrete log over prime fields, graph isomorphism (GI), and a number of similar isomorphism problems.

Whereas NP-complete problems all reduce one to another, even under fairly simple reductions, less is known about the relative difficulty of presumed NP-intermediate problems. Regarding MCSP and MKTP, factoring integers and discrete log over prime fields are known to reduce to both under randomized reductions with zero-sided error [1, 27]. Recently, Allender and Das [2] showed that GI and all of SZK (Statistical Zero Knowledge) reduce to both under randomized reductions with bounded error.

Those reductions and, in fact, *all* prior reductions of supposedly-intractable problems to MCSP / MKTP proceed along the same well-trodden path. Namely, MCSP / MKTP is used as an efficient statistical test to distinguish random distributions from pseudorandom distributions, where the pseudorandom distribution arises from a hardness-based pseudorandom generator construction. In particular, [20] employs the construction based on the hardness of factoring Blum integers, [1, 2, 5, 27] use the construction from [16] based on the existence of one-way functions, and [1, 9] make use of the Nisan-Wigderson construction [24]. The property that MCSP / MKTP breaks the construction implies that the underlying hardness assumption fails relative to MCSP / MKTP, and thus that the supposedly hard problem reduces to MCSP / MKTP.

1.1 Contributions

The main conceptual contribution of our paper is a fundamentally different way of constructing reductions to MKTP based on a novel use of known interactive proof systems. Our approach

applies to GI and a broad class of isomorphism problems. A common framework for those isomorphism problems is another conceptual contribution. In terms of results, our new approach allows us to eliminate the errors in the recent reductions from GI to MKTP, and more generally to establish *zero-sided error* randomized reductions to MKTP from many isomorphism problems within our framework. These include Linear Code Equivalence, Matrix Subspace Conjugacy, and Permutation Group Conjugacy (see Section 3.1 for the definitions). The technical contributions mainly consist of encodings of isomorphism classes that are efficiently decodable and achieve compression that is at or near the information-theoretic optimum.

We note that our techniques remain of interest even in light of the recent quasi-polynomial-time algorithm for GI [6]. For one, GI is still not known to be in P, and Group Isomorphism stands as a significant obstacle to this (as stated at the end of [6]). More importantly, our techniques also apply to the other isomorphism problems mentioned above, for which the current best algorithms are still exponential.

Let us also provide some evidence that our approach for constructing reductions to MKTP differs in an important way from the existing ones. We claim that the existing approach can only yield zero-sided error reductions to MKTP from problems that are in $\mathsf{NP} \cap \mathsf{coNP}$, a class that neither GI nor any of the other isomorphism problems mentioned above are known to reside in. The reason for the claim is that the underlying hardness assumptions are fundamentally average-case, which implies that the reduction can have both false positives and false negatives. For example, in the papers employing the construction from [16], MKTP is used in a subroutine to invert a polynomial-time-computable function, and the subroutine may fail to find an inverse. Given a reliable but imperfect subroutine, the traditional way to eliminate false positives is to use the subroutine for constructing an efficiently verifiable membership witness, and only accept after verifying its validity. As such, the existence of a traditional reduction without false positives from a language L to MKTP implies that $L \in \mathsf{NP}$. Similarly, a traditional reduction from L to MKTP without false negatives is only possible if $L \in \mathsf{coNP}$, and zero-sided error is only possible if $L \in \mathsf{NP} \cap \mathsf{coNP}$.

Instead of using the oracle for MKTP in the *construction* of a candidate witness and then verifying the validity of the candidate without the oracle, we use the power of the oracle in the *verification* process. This obviates the need for the language L to be in $\mathsf{NP} \cap \mathsf{coNP}$ in the case of reductions with zero-sided error.

1.2 Organization

In Section 2 we develop our technique for $L = \mathrm{GI}$ in an informal way, and in Section 3 we extend it to a broad class of isomorphism problems. In Section 4 we rigorously develop our result for GI. A formal treatment for other isomorphism problems is deferred to the full version [3]. Section 5 presents the information-theoretically optimal encodings that we use for our result for GI. In Section 6 we suggest directions for further research.

2 Main Idea for Graph Isomorphism

Recall that an instance of GI consists of a pair (G_0, G_1) of graphs on the vertex set $[n]$, and the question is whether $G_0 \equiv G_1$, i.e., whether there exists a permutation $\pi \in S_n$ such that $G_1 = \pi(G_0)$, where $\pi(G_0)$ denotes the result of applying the permutation π to the vertices of G_0. In order to develop a zero-sided error algorithm for GI, it suffices to develop one without false negatives. This is because the false positives can subsequently be eliminated using the known search-to-decision reduction for GI [22].

The crux for obtaining a reduction without false negatives from GI to MKTP is a witness system for the complement $\overline{\text{GI}}$ inspired by the well-known two-round interactive proof system for $\overline{\text{GI}}$ [11]. Consider the distribution $R_G(\pi) \doteq \pi(G)$ where $\pi \in S_n$ is chosen uniformly at random. By the Orbit–Stabilizer Theorem, for any fixed G, R_G is uniform over a set of size $N \doteq n!/|\operatorname{Aut}(G)|$ and thus has entropy $s = \log(N)$, where $\operatorname{Aut}(G) \doteq \{\pi \in S_n : \pi(G) = G\}$ denotes the set of automorphisms of G. For ease of exposition, let us assume that $|\operatorname{Aut}(G_0)| = |\operatorname{Aut}(G_1)|$ (which is actually the hardest case for GI), so both R_{G_0} and R_{G_1} have the same entropy s. Consider picking $r \in \{0,1\}$ uniformly at random, and setting $G = G_r$. If $(G_0, G_1) \in \text{GI}$, the distributions R_{G_0}, R_{G_1}, and R_G are all identical, and therefore R_G also has entropy s. On the other hand, if $(G_0, G_1) \notin \text{GI}$, the entropy of R_G equals $s + 1$. The extra bit of information corresponds to the fact that in the nonisomorphic case each sample of R_G reveals the value of r that was used, whereas that bit gets lost in the reduction in the isomorphic case.

The difference in entropy suggests that a typical sample of R_G can be compressed more in the isomorphic case than in the nonisomorphic case. If we can compute some threshold such that $\text{KT}(R_G)$ *never* exceeds the threshold in the isomorphic case, and exceeds it with nonnegligible probability in the nonisomorphic case, we have the witness system for $\overline{\text{GI}}$ that we aimed for: Take a sample from R_G, and use the oracle for MKTP to check that it cannot be compressed at or below the threshold. The entropy difference of 1 may be too small to discern, but we can amplify the difference by taking multiple samples and concatenating them. Thus, we end up with a randomized mapping reduction of the following form, where t denotes the number of samples and θ the threshold:

$$\begin{array}{l}\text{Pick } r \doteq r_1 \dots r_t \in \{0,1\}^t \text{ and } \pi_i \in S_n \text{ for } i \in [t], \text{ each uniformly at random.}\\ \text{Output } (y, \theta) \text{ where } y \doteq y_1 \dots y_t \text{ and } y_i \doteq \pi_i(G_{r_i}).\end{array} \tag{1}$$

2.1 Rigid Case

We need to analyze how to set the threshold θ and argue correctness for a value of t that is polynomially bounded. In order to do so, let us first consider the case where the graphs G_0 and G_1 are *rigid*, i.e., they have no nontrivial automorphisms, or equivalently, $s = \log(n!)$.

- If $G_0 \not\equiv G_1$, the string y contains all of the information about the random string r and the t random permutations $\pi_1, \dots, \pi_t$, which amounts to $ts + t = t(s+1)$ bits of information. This implies that y has KT-complexity close to $t(s+1)$ with high probability.
- If $G_0 \equiv G_1$, then we can efficiently produce each bit of y from the adjacency matrix representation of G_0 (n^2 bits) and the function table of permutations $\tau_i \in S_n$ (for $i \in [t]$) such that $y_i \doteq \pi_i(G_{r_i}) = \tau_i(G_0)$. Moreover, the set of all permutations S_n allows an efficiently decodable indexing, i.e., there exists an efficient algorithm that takes an index $k \in [n!]$ and outputs the function table of the k-th permutation in S_n according to some ordering. An example of such an indexing is the Lehmer code (see, e.g., [21, pp. 12-13] for specifics). This shows that

$$\text{KT}(y) \leq t\lceil s \rceil + (n + \log(t))^c \tag{2}$$

  for some constant c, where the first term represents the cost of the t indices of $\lceil s \rceil$ bits each, and the second term represents the cost of the n^2 bits for the adjacency matrix of G_0 and the polynomial running time of the decoding process.

If we ignore the difference between s and $\lceil s \rceil$, the right-hand side of (2) becomes $ts + n^c$, which is closer to ts than to $t(s+1)$ for t any sufficiently large polynomial in n, say $t = n^{c+1}$. Thus, setting θ halfway between ts and $t(s+1)$, i.e., $\theta \doteq t(s + \frac{1}{2})$, ensures that $\text{KT}(y) > \theta$

holds with high probability if $G_0 \not\equiv G_1$, and never holds if $G_0 \equiv G_1$. This yields the desired randomized mapping reduction without false negatives, modulo the rounding issue of s to $\lceil s \rceil$. The latter can be handled by aggregating the permutations τ_i into blocks so as to make the amortized cost of rounding negligible. The details are captured in the Blocking Lemma of Section 4.2.

2.2 General Case

What changes in the case of non-rigid graphs? For ease of exposition, let us again assume that $|\operatorname{Aut}(G_0)| = |\operatorname{Aut}(G_1)|$. There are two complications:

(i) We no longer know how to efficiently compute the threshold $\theta \doteq t(s + \frac{1}{2})$ because $s \doteq \log(N)$ and $N \doteq \log(n!/|\operatorname{Aut}(G_0)|) = \log(n!/|\operatorname{Aut}(G_1)|)$ involves the size of the automorphism group.

(ii) The Lehmer code no longer provides sufficient compression in the isomorphic case as it requires $\log(n!)$ bits per permutation whereas we only have s to spend, which could be considerably less than $\log(n!)$.

In order to resolve (ii) we develop an efficiently decodable indexing of cosets for any subgroup of S_n given by a list of generators (see Lemma 10 in Section 4.3). In fact, our scheme even works for cosets of a subgroup within another subgroup of S_n, a generalization that may be of independent interest (see Lemma 13 in Section 5). Applying our scheme to $\operatorname{Aut}(G)$ and including a minimal list of generators for $\operatorname{Aut}(G)$ in the description of the program p allows us to maintain (2).

Regarding (i), we can deduce a good approximation to the threshold with high probability by taking, for both choices of $r \in \{0, 1\}$, a polynomial number of samples of R_{G_r} and using the oracle for MKTP to compute the exact KT-complexity of their concatenation. This leads to a randomized reduction from GI to MKTP with bounded error (from which one without false positives follows as mentioned before), reproving the earlier result of [2] using our new approach (see Remark 11 in Section 4.3 for more details).

In order to avoid false negatives, we need to improve the above approximation algorithm such that it never produces a value that is too small, while maintaining efficiency and the property that it outputs a good approximation with high probability. In order to do so, it suffices to develop a *probably-correct overestimator* for the quantity $n!/|\operatorname{Aut}(G)|$, i.e., a randomized algorithm that takes as input an n-vertex graph G, produces the correct quantity with high probability, and never produces a value that is too small; the algorithm should run in polynomial time with access to an oracle for MKTP. Equivalently, it suffices to develop a probably-correct *under*estimator of similar complexity for $|\operatorname{Aut}(G)|$.

The latter can be obtained from the known search-to-decision procedures for GI, and answering the oracle calls to GI using the above two-sided error reduction from GI to MKTP. There are a number of ways to implement this strategy; here is one that generalizes to a number of other isomorphism problems including Linear Code Equivalence.

1. Find a list of permutations that generates a subgroup of $\operatorname{Aut}(G)$ such that the subgroup equals $\operatorname{Aut}(G)$ with high probability.

   Finding a list of generators for $\operatorname{Aut}(G)$ deterministically reduces to GI. Substituting the oracle for GI by a two-sided error algorithm yields a list of permutations that generates $\operatorname{Aut}(G)$ with high probability, and always produces a subgroup of $\operatorname{Aut}(G)$. The latter property follows from the inner workings of the reduction, or can be imposed explicitly by checking every permutation produced and dropping it if it does not map G to itself. We use the above randomized reduction from GI to MKTP as the two-sided error algorithm for GI.

2. Compute the order of the subgroup generated by those permutations.
   There is a known deterministic polynomial-time algorithm to do this [29].

The resulting probably-correct underestimator for $|\operatorname{Aut}(G)|$ runs in polynomial time with access to an oracle for MKTP. Plugging it into our approach, we obtain a randomized reduction from GI to MKTP without false negatives. A reduction with zero-sided error follows as discussed earlier. Thus, we have established the following result.

▶ **Theorem 1.** $\mathrm{GI} \in \mathsf{ZPP}^{\mathrm{MKTP}}$.

Before applying our approach to other isomorphism problems, let us point out the important role that the Orbit–Stabilizer Theorem plays. A randomized algorithm for finding generators for a graph's automorphism group yields a probably-correct underestimator for the size of the automorphism group, as well as a randomized algorithm for GI without false positives. The Orbit–Stabilizer Theorem allows us to turn a probably-correct underestimator for $|\operatorname{Aut}(G)|$ into a probably-correct overestimator for the size of the support of R_G, thereby switching the error from one side to the other, and allowing us to avoid false negatives instead of false positives.

3 Generalization

Our approach extends to several other isomorphism problems. We first present a definition of a generic isomorphism problem, and then informally develop the generalization of Theorem 1. We refer to the full version [3] for the formal proofs.

3.1 Framework

We consider the following framework, parameterized by an underlying family of group actions (Ω, H) where H is a group that acts on the universe Ω. We typically think of the elements of Ω as abstract objects, which need to be described in string format in order to be input to a computer; we let $\omega(z)$ denote the abstract object represented by the string z.

▶ **Definition 2** (Isomorphism Problem)**.** An instance of an Isomorphism Problem consists of a pair $x = (x_0, x_1)$ that determines a universe Ω_x and a group H_x that acts on Ω_x such that $\omega_0(x) \doteq \omega(x_0)$ and $\omega_1(x) \doteq \omega(x_1)$ belong to Ω_x. Each $h \in H_x$ is identified with the permutation $h : \Omega_x \to \Omega_x$ induced by the action. The goal is to determine whether there exists $h \in H_x$ such that $h(\omega_0(x)) = \omega_1(x)$.

When it causes no confusion, we drop the argument x and simply write H, Ω, ω_0, and ω_1. We often blur the—sometimes pedantic—distinction between z and $\omega(z)$. For example, in GI, each z is an $n \times n$ binary matrix (a string of length n^2), and represents the abstract object $\omega(z)$ of a graph with n labeled vertices; thus, in this case the correspondence between z and $\omega(z)$ is a bijection. The group H is the symmetric group S_n, and the action is by permuting the labels.

For completeness, we include below the definitions of the instantiations we mentioned in Section 1.1. Table 1 summarizes how they can be cast in the framework.

Linear code equivalence. A *linear code* over the finite field $\mathbb{F}_q$ is a d-dimensional linear subspace of $\mathbb{F}_q^n$ for some n. Two such codes are (permutationally) *equivalent* if there is a permutation of the n coordinates that makes them equal as subspaces.

Linear Code Equivalence is the problem of deciding whether two linear codes are equivalent, where the codes are specified as the row-span of a $d \times n$ matrix (of rank d), called a *generator*

Table 1 Instantiations of the Isomorphism Problem

Problem	H	Ω
Graph Isomorphism	S_n	graphs with n labeled vertices
Linear Code Equivalence	S_n	subspaces of dimension d in $\mathbb{F}_q^n$
Permutation Group Conjugacy	S_n	subgroups of S_n
Matrix Subspace Conjugacy	$\mathrm{GL}_n(\mathbb{F}_q)$	subspaces of dimension d in $\mathbb{F}_q^{n\times n}$

matrix. There exists a mapping reduction from GI to Linear Code Equivalence over any field [26, 15]; Linear Code Equivalence is generally thought to be harder than GI.

In order to cast Code Equivalence in our framework, we consider the family of actions $(S_n, \Omega_{n,d,q})$ where $\Omega_{n,d,q}$ denotes the linear codes of length n and dimension d over $\mathbb{F}_q$, and S_n acts by permuting the coordinates.

Permutation Group Conjugacy. Two permutation groups $\Gamma_0, \Gamma_1 \leq S_n$ are *conjugate* (or permutationally isomorphic) if there exists a permutation $\pi \in S_n$ such that $\Gamma_1 = \pi\Gamma_0\pi^{-1}$; such a π is called a conjugacy.

The *Permutation Group Conjugacy* problem is to decide whether two subgroups of S_n are conjugate, where the subgroups are specified by a list of generators. The problem is known to be in $\mathsf{NP} \cap \mathsf{coAM}$, and is at least as hard as Linear Code Equivalence. Currently the best known algorithm runs in time $2^{O(n)}\,\mathrm{poly}(|\Gamma_1|)$ [7]—that is, the runtime depends not only on the input size (which is polynomially related to n), but also on the size of the groups generated by the input permutations, which can be exponentially larger.

Matrix Subspace Conjugacy. A *linear matrix space* over $\mathbb{F}_q$ is a d-dimensional linear subspace of $n \times n$ matrices. Two such spaces V_0 and V_1 are *conjugate* if there is an invertible $n \times n$ matrix X such that $V_1 = XV_0X^{-1} \doteq \{X \cdot M \cdot X^{-1} : M \in V_0\}$, where "$\cdot$" represents matrix multiplication.

Matrix Subspace Conjugacy is the problem of deciding whether two linear matrix spaces are conjugate, where the spaces are specified as the linear span of d linearly independent $n \times n$ matrices. There exist mapping reductions from GI and Linear Code Equivalence to Matrix Subspace Conjugacy [15]; Matrix Subspace Conjugacy is generally thought to be harder than Linear Code Equivalence.

3.2 Generic Result

We generalize our construction for GI to any Isomorphism Problem by replacing $R_G(\pi) \doteq \pi(G)$ where $\pi \in S_n$ is chosen uniformly at random, by $R_\omega(h) \doteq h(\omega)$ where $h \in H$ is chosen uniformly at random. The analysis that the construction yields a randomized reduction without false negatives from the Isomorphism Problem to MKTP carries over, provided that the Isomorphism Problem satisfies the following properties.

1. The underlying group H is *efficiently samplable*, and the action $(\omega, h) \mapsto h(\omega)$ is efficiently computable. We need this property in order to make sure the reduction is efficient.
2. There is an efficiently computable *normal form* for representing elements of Ω as strings. This property trivially holds in the setting of GI as there is a unique adjacency matrix that represents any given graph on the vertex set $[n]$. However, uniqueness of representation need not hold in general. Consider, for example, Permutation Group Conjugacy. An instance of this problem abstractly consists of two permutation groups (Γ_0, Γ_1), represented

(as usual) by a sequence of elements of S_n generating each group. In that case there are many strings representing the same abstract object, i.e., a subgroup has many different sets of generators.

For the correctness analysis in the isomorphic case it is important that H acts on the abstract objects, and *not* on the binary strings that represent them. In particular, the output of the reduction should only depend on the abstract object $h(\omega)$, and not on the way ω was provided as input. This is because the latter may leak information about the value of the bit r that was picked. The desired independence can be guaranteed by applying a normal form to the representation before outputting the result. In the case of Permutation Group Conjugacy, this means transforming a set of permutations that generate a subgroup Γ into a canonical set of generators for Γ.

In fact, it suffices to have an efficiently computable *complete invariant* for Ω, i.e., a mapping from representations of objects from Ω to strings such that the image only depends on the abstract object, and is different for different abstract objects.

3. There exists a probably-correct overestimator for $N \doteq |H|/|\operatorname{Aut}(\omega)|$ that is computable efficiently with access to an oracle for MKTP. We need this property to set the threshold $\theta \doteq t(s + \frac{1}{2})$ with $s \doteq \log(N)$ correctly.
4. There exists an encoding for cosets of $\operatorname{Aut}(\omega)$ in H that achieves KT-complexity close to the information-theoretic optimum. This property ensures that in the isomorphic case the KT-complexity is never much larger than the entropy.

Properties 1 and 2 are fairly basic. Property 4 may seem to require an instantiation-dependent approach. However, we develop a *generic* hashing-based encoding scheme that meets the requirements. In fact, we give a nearly-optimal encoding scheme for any samplable distribution that is almost flat, without reference to isomorphism. Unlike the indexings from Lemma 10 for the special case where H is the symmetric group, the generic construction does not achieve the information-theoretic optimum, but it comes sufficiently close for our purposes.

The notion of a probably-correct overestimator in Property 3 can be further relaxed to that of a *probably-approximately-correct overestimator*, or *pac overestimator* for short. This is a randomized algorithm that with high probability outputs a value within an absolute deviation bound of Δ from the correct value, and never produces a value that is more than Δ below the correct value. More precisely, it suffices to efficiently compute with access to an oracle for MKTP a pac overestimator for $s \doteq \log(|H|/|\operatorname{Aut}(\omega)|)$ with deviation $\Delta = 1/4$. The relaxation suffices because of the difference of about $1/2$ between the threshold θ and the actual KT-values in both the isomorphic and the non-isomorphic case.

Moreover, Properties 1 and 2 are sufficient to generalize the construction of Allender and Das [2], which yields randomized reductions of the isomorphism problem to MKTP without false positives (irrespective of whether a search-to-decision reduction is known). This leads to the following generalization of Theorem 1.

▶ **Theorem 3.** *Let* Iso *denote an Isomorphism Problem as in Definition 2. Consider the following conditions:*

1. [action sampler] *The uniform distribution on H_x is uniformly samplable in polynomial time, and the mapping $(\omega, h) \mapsto h(\omega)$ underlying the action (Ω_x, H_x) is computable in* ZPP*.*
2. [complete universe invariant] *There exists a complete invariant ν for the representation ω that is computable in* ZPP*.*
3. [entropy estimator] *There exists a probably-approximately-correct overestimator for $(x, \omega) \mapsto \log(|H_x|/|\operatorname{Aut}(\omega)|)$ with deviation $\Delta = 1/4$ that is computable in randomized time* $\operatorname{poly}(|x|)$ *with access to an oracle for* MKTP*.*

With these definitions:
(a) *If conditions 1 and 2 hold, then* Iso $\in \mathsf{RP}^{\mathrm{MKTP}}$.
(b) *If conditions 1, 2, and 3 hold, then* Iso $\in \mathsf{coRP}^{\mathrm{MKTP}}$.

As $s = \log|H| - \log|\operatorname{Aut}(\omega)|$ in Property 3, it suffices to have a pac overestimator for $\log|H|$ and a pac *under*estimator for $\log|\operatorname{Aut}(\omega)|$, both to within deviation $\Delta/2 = 1/8$ and of the required efficiency. Generalizing our approach for GI, one way to obtain the desired underestimator for $\log|\operatorname{Aut}(\omega)|$ is by showing how to efficiently compute with access to an oracle for MKTP:
(a) a list L of elements of H that generates a subgroup $\langle L\rangle$ of $\operatorname{Aut}(\omega)$ such that $\langle L\rangle = \operatorname{Aut}(\omega)$ with high probability, and
(b) a pac underestimator for $\log|\langle L\rangle|$, the logarithm of the order of the subgroup generated by a given list L of elements of H.
Further mimicking our approach for GI, we know how to achieve (a) when the Isomorphism Problem allows a search-to-decision reduction. Such a reduction is known for Linear Code Equivalence, but remains open for problems like Matrix Subspace Conjugacy and Permutation Group Conjugacy. However, we show that (a) holds for a *generic* isomorphism problem provided that products and inverses in H can be computed efficiently. The proof hinges on the ability of MKTP to break the pseudo-random generator construction of [16] based on a purported one-way function (see Theorem 45 from [1]).

As for (b), we know how to efficiently compute the order of the subgroup *exactly* in the case of permutation groups ($H = S_n$), even without an oracle for MKTP, and in the case of many matrix groups over finite fields ($H = \mathrm{GL}_n(\mathbb{F}_q)$) with oracle access to MKTP, but some cases remain open. Instead, we show how to *generically* construct a *pac underestimator* with small deviation given access to MKTP as long as products and inverses in H can be computed efficiently, and H allows an efficient complete invariant. The first two conditions are sufficient to efficiently generate a distribution $\widetilde{p}$ on $\langle L\rangle$ that is uniform to within a small relative deviation [8]. The entropy $\widetilde{s}$ of that distribution equals $\log|\langle L\rangle|$ to within a small additive deviation. As $\widetilde{p}$ is (essentially) flat, our generic encoding scheme shows that $\widetilde{p}$ has an encoding whose length does not exceed $\widetilde{s}$ by much, and that can be decoded by small circuits. Given an efficient complete invariant for H, we can use an approach similar to the one we used to approximate the threshold θ to construct a pac underestimator for $\widetilde{s}$ with small additive deviation, namely the amortized KT-complexity of the concatenation of a polynomial number of samples from $\widetilde{p}$. With access to an oracle for MKTP we can efficiently evaluate KT. As a result, we obtain a pac underestimator for $\log|\langle L\rangle|$ with a small additive deviation that is efficiently computable with oracle access to MKTP.

This gives the following specialization of Theorem 3:

▶ **Theorem 4.** *Let* Iso *denote an Isomorphism Problem as in Definition 2. Suppose that the ensemble* $\{H_x\}$ *has a representation* η *such that conditions 1 and 2 of Theorem 3 hold as well as the following additional conditions:*
4. [group operations] *Products and inverses in* H_x *are computable in* ZPP.
5. [sample space estimator] *The map* $x \mapsto |H_x|$ *has a pac overestimator with deviation* $\Delta = 1/8$ *computable in* $\mathsf{ZPP}^{\mathrm{MKTP}}$.
6. [complete group invariant] *There exists a complete invariant* ζ *for the representation* η *that is computable in* ZPP.
Then Iso $\in \mathsf{ZPP}^{\mathrm{MKTP}}$.

Theorem 4 allows us to show that all of the isomorphism problems in Table 1 reduce to MKTP under randomized reductions with zero-sided error. We refer to the full version [3] for a complete treatment.

▶ **Corollary 5.** *Linear Code Equivalence, Permutation Group Conjugacy, and Matrix Subspace Conjugacy are in* $\mathsf{ZPP}^{\mathrm{MKTP}}$.

4 Technical Development for Graph Isomorphism

This section is dedicated to a rigorous proof of Theorem 1. We start with the formal definition of MKTP, and then follow the outline of Section 2, taking the same four steps, and filling in the missing details.

4.1 KT Complexity

Theorem 1 states a reduction from GI to the Minimum KT Problem. The measure KT that we informally described in Section 1, was introduced and formally defined as follows in [1]. We refer to that paper for more background and motivation for the particular definition.

▶ **Definition 6** (KT)**.** Let U be a universal Turing machine. For each string x, define $\mathrm{KT}_U(x)$ to be

$$\min\{\, |d| + T : \ (\forall \sigma \in \{0,1,*\})\ (\forall i \leq |x|+1)\ U^d(i,\sigma) \text{ accepts in } T \text{ steps iff } x_i = \sigma \,\}.$$

We define $x_i = *$ if $i > |x|$; thus, for $i = |x|+1$ the machine accepts iff $\sigma = *$. The notation U^d indicates that the machine U has random access to the description d.

$\mathrm{KT}(x)$ is defined to be equal to $\mathrm{KT}_U(x)$ for a fixed choice of universal machine U with logarithmic simulation time overhead [1, Proposition 5]. In particular, if d consists of the description of a Turing machine M that runs in time $t_M(n)$ and some auxiliary information a such that $M^a(i) = x_i$ for $i \in [n]$, then $\mathrm{KT}(x) \leq |a| + c_M T_M(\log n) \log(T_M(\log n))$, where $n \doteq |x|$ and c_M is a constant depending on M. It follows that $(\mu/\log n)^{\Omega(1)} \leq \mathrm{KT}(x) \leq (\mu \cdot \log n)^{O(1)}$ where μ represents the circuit complexity of the mapping $i \mapsto x_i$ [1, Theorem 11]. The Minimum KT Problem is defined as $\mathrm{MKTP} \doteq \{(x,\theta) \mid \mathrm{KT}(x) \leq \theta\}$.

4.2 Rigid Graphs

The crux of Theorem 1 is the randomized mapping reduction from deciding whether a given pair of n-vertex graphs (G_0, G_1) is in GI to deciding whether $(y,\theta) \in \mathrm{MKTP}$, as prescribed by (1). Recall that (1) involves picking a string $r \doteq r_1 \dots r_t \in \{0,1\}^t$ and permutations π_i at random, and constructing the string $y = y_1 \dots y_t$, where $y_i = \pi_i(G_{r_i})$. We show how to determine θ such that a sufficiently large polynomial t guarantees that the reduction has no false negatives.

We first consider the simplest setting, in which both G_0 and G_1 are rigid. We argue that $\theta \doteq t(s + \frac{1}{2})$ works, where $s = \log(n!)$.

Nonisomorphic Case If $G_0 \not\equiv G_1$, then (by rigidity), each choice of r and each distinct sequence of t permutations results in a different string y, and thus the distribution on the strings y has entropy $t(s+1)$ where $s \doteq \log(n!)$. By a straightforward counting argument, a typical string sampled from such a distribution will have high KT-complexity. Formally, we have the following:

▶ **Proposition 7.** *Let y be sampled from a distribution with min-entropy s. For all k, we have $\mathrm{KT}(y) \geq s - k$ except with probability at most 2^{-k}.*

Our y is sampled from a uniform distribution, hence the entropy and min-entropy coincide. Thus $\mathrm{KT}(y) > \theta = t(s+1) - \frac{t}{2}$ with all but exponentially small probability in t, and so with high probability the algorithm declares G_0 and G_1 nonisomorphic.

Isomorphic Case. If $G_0 \equiv G_1$, we need to show that $\mathrm{KT}(y) \leq \theta$ always holds. The key insight is that the information in y is precisely captured by the t permutations $\tau_1, \tau_2, \ldots, \tau_t$ such that $\tau_i(G_0) = y_i$. These permutations exist because $G_0 \equiv G_1$; they are unique by the rigidity assumption. Thus, y contains at most ts bits of information. We show that its KT-complexity is not much larger than this.

We do this using an *efficiently decodable indexing* of the symmetric groups S_n. This is, for each n, a bijective map $[n!] \to S_n$ so that, on input i, the image of i can be computed in time $\mathrm{poly}(n)$. We rely on the following indexing, due to Lehmer (see, e.g., [21, pp. 12–33]):

▶ **Proposition 8** (Lehmer code). *The symmetric groups S_n have indexings that are uniformly decodable in time* $\mathrm{poly}(n)$.

To bound $\mathrm{KT}(y)$, we consider a program d that has the following information hard-wired into it: n, the adjacency matrix of G_0, and the t integers $k_1, \ldots, k_t \in [n!]$ encoding $\tau_1, \ldots, \tau_t$. We use the decoder from Proposition 8 to compute the i-th bit of y on input i. This can be done in time $\mathrm{poly}(n, \log(t))$ given the hard-wired information.

As mentioned in Section 1, a naïve method for encoding the indices $k_1, \ldots, k_t$ only gives the bound $t\lceil s \rceil + \mathrm{poly}(n, \log(t))$ on $\mathrm{KT}(y)$, which may exceed $t(s+1)$ and—*a fortiori*—the threshold θ, no matter how large a polynomial t is. We remedy this by aggregating multiple indices into blocks, and amortizing the encoding overhead across multiple samples. The following technical lemma captures the technique.

▶ **Lemma 9** (Blocking Lemma for Indexings). *Let $\{T_x\}$ be an ensemble of sets of strings such that all strings in T_x have the same length* $\mathrm{poly}(|x|)$. *Suppose that each T_x has an indexing decodable by circuits of size* $\mathrm{poly}(|x|)$. *Then there are constants $\alpha_0 > 0$ and c so that, for all $t \in \mathbb{N}$ and all sufficiently large $x \in \{0,1\}^*$, and every y that is the concatenation of t elements of T_x*

$$\mathrm{KT}(y) \leq t \log |T_x| + t^{1-\alpha_0}|x|^c$$

We first show how to apply the Blocking Lemma and then prove it. For a given rigid graph G, we let T_G be the set of adjacency matrices of permutations of G. To index T_G, we associate to each permutation $\tau(G)$ the index k of τ from the Lehmer code. Then there is a circuit of size $\mathrm{poly}(|G|)$ which takes as input k, computes τ, and then outputs $\tau(G)$. By the Blocking Lemma, we have that

$$\mathrm{KT}(y) \leq ts + t^{1-\alpha_0} n^c \tag{3}$$

for some constants $\alpha_0 > 0$ and c, and all sufficiently large n. Taking $t = n^{1+c/\alpha_0}$, we see that for all sufficiently large n, $\mathrm{KT}(y) \leq t(s + \frac{1}{2}) \doteq \theta$.

Proof of Lemma 9. Let T_x and D_x be the hypothesized ensemble of sets of strings and corresponding decoders. Fix x and t, let $m = \mathrm{poly}(|x|)$ denote the length of the strings in T_x, and let $b \in \mathbb{N}$ be a parameter to be set later.

To bound $\mathrm{KT}(y)$, we first write $y = y_1 \cdots y_t$ where each $y_j \in T_x$, and let $k_j \in [|T_x|]$ be the index of y_j via D_x. (i.e., $D_x(k_j) = y_j$.) We group the y_j's into $\lceil t/b \rceil$ size-b blocks $\widetilde{y}_1, \widetilde{y}_2, \ldots, \widetilde{y}_{\lceil t/b \rceil}$. For each block $\widetilde{y}_j$, let $\widetilde{k}_j$ the number whose base-$|T_x|$ representation is

written $k_{b(j-1)+1}k_{b(j-1)+2}\cdots k_{bj}$. This is a number between 0 and $|T_x|^b - 1$, and hence can be expressed in binary with $\lceil b \log |T_x| \rceil$ bits. Given a circuit computing D_x, $\widetilde{y}_j$ can be computed from $\widetilde{k}_j$ in time polynomial in $|D_x|$, $\log |T_x|$, and b.

Consider a program d that has x, t, m, b, the circuit for computing D_x, and the indices $\widetilde{k}_1, \widetilde{k}_2, \ldots, \widetilde{k}_{\lceil t/b \rceil}$ hardwired, takes an input $i \in \mathbb{N}$, and determines the i-th bit of y as follows. It first computes $j_0, j_1 \in \mathbb{N}$ so that i points to the j_1-th bit position in $\widetilde{y}_{j_0}$. Then, using D_x, $\widetilde{k}_{j_0}$, and j_1, it computes $\widetilde{y}_{j_0}$ and outputs its j_1-th bit, which is the i-th bit of y.

The bit-length of d is at most $\lceil t/b \rceil \cdot \lceil b \log |T_x| \rceil$ for the indices, plus $\mathrm{poly}(|x|, b, \log t)$ for the rest. The time needed by d is bounded by $\mathrm{poly}(|x|, b, \log t)$. Thus $\mathrm{KT}(y) \leq \lceil t/b \rceil \lceil b \log |T_x| \rceil + \mathrm{poly}(|x|, b, \log t) \leq t \log |T_x| + t/b + \mathrm{poly}(|x|, b, \log t)$. The lemma follows by choosing $b = \lceil t^{\alpha_0} \rceil$ for a sufficiently small constant α_0. ◀

4.3 Known Number of Automorphisms

We generalize the case of rigid graphs to graphs for which we know the size of their automorphism groups. Specifically, in addition to the two input graphs G_0 and G_1, we are also given numbers N_0, N_1 where $N_i \doteq n!/|\operatorname{Aut}(G_i)|$. Note that if $N_0 \neq N_1$, we can right away conclude that $G_0 \not\equiv G_1$. Nevertheless, we do not assume that $N_0 = N_1$ as the analysis of the case $N_0 \neq N_1$ will be useful in Section 4.4.

The reduction is the same as in Section 4.2 with the correct interpretation of s. The main difference lies in the analysis, where we need to accommodate for the loss in entropy that comes from having multiple automorphisms.

Let $s_i \doteq \log(N_i)$ be the entropy in a random permutation of G_i. Set $s \doteq \min(s_0, s_1)$, and $\theta \doteq t(s + \frac{1}{2})$. In the nonisomorphic case the min-entropy of y is at least $t(s+1)$, so $\mathrm{KT}(y) > \theta$ with high probability. In the isomorphic case we upper bound $\mathrm{KT}(y)$ by about ts. Unlike the rigid case, we can no longer afford to encode an entire permutation for each permuted copy of G_0.

Instead, we need an indexing for the cosets of $\operatorname{Aut}(G_0)$ within S_n. Formally, this means a map $[n!/|\operatorname{Aut}(G_0)|] \to S_n$ so that for every coset of $\operatorname{Aut}(G_0)$ in S_n, there is an index whose image through the map is in the coset. Given such an indexing, we can get an indexing of the set T_{G_0} of strings consisting of the adjacency matrices of permutations of G_0. The following indexing, applied to $\Gamma = \operatorname{Aut}(G_0)$, suffices for this purpose.

▶ **Lemma 10.** *For every subgroup Γ of S_n there exists an indexing of the cosets of Γ that is decodable by circuits of size* $\mathrm{poly}(n)$.

We prove Lemma 10 in Section 5 as a corollary to a more general lemma that gives, for each $\Gamma \leq H \leq S_n$, an efficiently computable indexing for the cosets of Γ in H.

▶ **Remark 11.** Before we continue towards Theorem 1, we point out that the above ideas yield an alternate proof that $\mathrm{GI} \in \mathsf{BPP}^{\mathrm{MKTP}}$ (and hence that $\mathrm{GI} \in \mathsf{RP}^{\mathrm{MKTP}}$). This weaker result was already obtained in [2] along the well-trodden path discussed in Section 1; this remark shows how to obtain it using our new approach.

The key observation is that in both the isomorphic and the nonisomorphic case, with high probability $\mathrm{KT}(y)$ stays away from the threshold θ by a growing margin. Moreover, the above analysis allows us to efficiently obtain high-confidence approximations of θ to within any constant using sampling and queries to the MKTP oracle.

More specifically, for $i \in \{0, 1\}$, let $\widetilde{y}_i$ denote the concatenation of $\widetilde{t}$ independent samples from R_{G_i}. Our analysis shows that $\mathrm{KT}(\widetilde{y}_i) \leq \widetilde{t}s_i + \widetilde{t}^{1-\alpha_0} n^c$ always holds, and that $\mathrm{KT}(\widetilde{y}_i) \geq \widetilde{t}s_i - \widetilde{t}^{1-\alpha_0} n^c$ holds with high probability. Thus, $\widetilde{s}_i \doteq \mathrm{KT}(\widetilde{y}_i)/\widetilde{t}$ approximates s_i with high

confidence to within an additive deviation of $n^c/\widetilde{t}^{\alpha_0}$. Similarly, $\widetilde{s} \doteq \min(\widetilde{s}_0, \widetilde{s}_1)$ approximates s to within the same deviation margin, and $\widetilde{\theta} \doteq t(\widetilde{s} + \frac{1}{2})$ approximates θ to within an additive deviation of $tn^c/\widetilde{t}^{\alpha_0}$. The latter bound can be made less than 1 by setting $\widetilde{t}$ to a sufficiently large polynomial in n and t. Moreover, all these estimates can be computed in time $\text{poly}(\widetilde{t}, n)$ with access to MKTP as MKTP enables us to evaluate KT efficiently.

4.4 Probably-Correct Underestimators for the Number of Automorphisms

The reason the $\mathsf{BPP}^{\text{MKTP}}$-algorithm in Remark 11 can have false negatives is that the approximation $\widetilde{\theta}$ to θ may be too small. Knowing the quantities $N_i \doteq n!/|\operatorname{Aut}(G_i)|$ exactly allows us to compute θ exactly and thereby obviates the possibility of false negatives. In fact, it suffices to compute overestimates for the quantities N_i which are correct with non-negligible probability. We capture this notion formally as follows:

▶ **Definition 12** (probably-correct overestimator)**.** Let $g : \Omega \to \mathbb{R}$ be a function, and M a randomized algorithm that, on input $\omega \in \Omega$, outputs a value $M(\omega) \in \mathbb{R}$. We say that M is a *probably-correct overestimator* for g if, for every $\omega \in \Omega$, $M(\omega) = g(\omega)$ holds with probability at least $1/\operatorname{poly}(|\omega|)$, and $M(\omega) > g(\omega)$ otherwise. A *probably-correct underestimator* for g is defined similarly by reversing the inequality.

We point out that, for any probably-correct over-/underestimator, taking the min/max among $\operatorname{poly}(|\omega|)$ independent runs yields the correct value with probability $1 - 2^{-\operatorname{poly}(|\omega|)}$.

We are interested in the case where $g(G) = n!/|\operatorname{Aut}(G)|$. Assuming this g on a given class of graphs Ω has a probably-correct overestimator M computable in randomized polynomial time with an MKTP oracle, we argue that GI on Ω reduces to MKTP in randomized polynomial time without false negatives.

To see this, consider the algorithm that, on input a pair (G_0, G_1) of n-vertex graphs, computes $\widetilde{N}_i = M(G_i)$ as estimates of the true values $N_i = \log(n!/|\operatorname{Aut}(G_i)|)$, and then runs the algorithm from Section 4.3 using the estimates $\widetilde{N}_i$.

- In the case where G_0 and G_1 are not isomorphic, if both estimates $\widetilde{N}_i$ are correct, then the algorithm detects $G_0 \not\equiv G_1$ with high probability.
- In the case where $G_0 \equiv G_1$, if $\widetilde{N}_i = N_i$ we showed in Section 4.3 that the algorithm always declares G_0 and G_1 to be isomorphic. Moreover, increasing θ can only decrease the probability of a false negative. As the computed threshold θ increases as a function of $\widetilde{N}_i$, and the estimate $\widetilde{N}_i$ is always at least as large as N_i, it follows that G_0 and G_1 are always declared isomorphic.

4.5 Arbitrary Graphs

A probably-correct overestimator for the function $G \mapsto n!/|\operatorname{Aut}(G)|$ on *any* graph G can be computed in randomized polynomial time with access to MKTP. The process is described in full detail in Section 1, based on a $\mathsf{BPP}^{\text{MKTP}}$ algorithm for GI (taken from Remark 11 or from [2]). This means that the setting of Section 4.4 is actually the general one. The only difference is that we no longer obtain a mapping reduction from GI to MKTP, but an oracle reduction: We still make use of (1), but we need more queries to MKTP in order to set the threshold θ.

This shows that $\text{GI} \in \mathsf{coRP}^{\text{MKTP}}$. As $\text{GI} \in \mathsf{RP}^{\text{MKTP}}$ follows from the known search-to-decision reduction for GI, this concludes the proof of Theorem 1 that $\text{GI} \in \mathsf{ZPP}^{\text{MKTP}}$.

5 Coset Indexings for Permutation Groups

In this section we develop the efficiently decodable indexings for cosets of permutation subgroups claimed in Lemma 10. In fact, we present a further generalization that may be of independent interest, namely an efficiently decodable indexing for cosets of permutation subgroups within another permutation subgroup.

▶ **Lemma 13.** *For all $\Gamma \leq H \leq S_n$, there exists an indexing of the cosets*[1] *of Γ within H that is uniformly decodable in polynomial time when Γ and H are given by a list of generators.*

Lemma 10 is just the instantiation of Lemma 13 with $H = S_n$ followed by hardwiring generators for Γ and S_n into a circuit simulating the decoder from Lemma 13. The proof of Lemma 13 requires some elements of the theory of permutation groups. Given a list of permutations $\pi_1, \ldots, \pi_k \in S_n$, we write $\Gamma = \langle \pi_1, \ldots, \pi_k \rangle \leq S_n$ for the subgroup they generate. Given a permutation group $\Gamma \leq S_n$ and a point $i \in [n]$, the Γ-orbit of i is the set $\{g(i) : g \in \Gamma\}$, and the Γ-stabilizer of i is the subgroup $\{g \in \Gamma : g(i) = i\} \leq \Gamma$.

We make use of the fact that (a) the number of cosets of a subgroup Γ of a group H equals $|H|/|\Gamma|$, and (b) the orbits of a subgroup Γ of H form a refinement of the orbits of H. We also need the following basic routines from computational group theory (see, for example, [19, 29]).

▶ **Proposition 14.** *Given a set of permutations that generate a subgroup $\Gamma \leq S_n$, the following can be computed in time polynomial in n:*

(1) *the cardinality $|\Gamma|$,*

(2) *a permutation in Γ that maps u to v for given $u, v \in [n]$, or report that no such permutation exists in Γ, and*

(3) *a list of generators for the subgroup Γ_v of Γ that stabilizes a given element $v \in [n]$.*

The proof of Lemma 13 makes implicit use of an efficient process for finding a *canonical representative* of $\pi\Gamma$ for a given permutation $\pi \in H$, where "canonical" means that the representative depends on the coset $\pi\Gamma$ only. The particular canonical representative the process produces can be specified as follows.

▶ **Definition 15.** For a permutation $\pi \in S_n$ and a subgroup $\Gamma \leq S_n$, the *canonical representative* of π modulo Γ, denoted $\pi \bmod \Gamma$, is the lexicographically least $\pi' \in \pi\Gamma$, where the lexicographic ordering is taken by viewing a permutation π' as the sequence $(\pi'(1), \pi'(2), \ldots, \pi'(n))$.

We describe the process as it provides intuition for the proof of Lemma 13.

▶ **Lemma 16.** *There exists a polynomial-time algorithm that takes as input a generating set for a subgroup $\Gamma \leq S_n$ and a permutation $\pi \in S_n$, and outputs the canonical representative $\pi \bmod \Gamma$.*

Proof of Lemma 16. Consider the element 1 of $[n]$. Permutations in $\pi\Gamma$ map 1 to an element v in the same Γ-orbit as $\pi(1)$, and for every element v in the Γ-orbit of $\pi(1)$ there exists a permutation in $\pi\Gamma$ that maps 1 to v. We can canonize the behavior of π on the element 1 by replacing π with a permutation $\pi_1 \in \pi\Gamma$ that maps 1 to the minimum element m in the

[1] The choice of left ($\pi\Gamma$) vs right ($\Gamma\pi$) cosets is irrelevant for us; all our results hold for both, and one can usually switch from one statement to the other by taking inverses. Related to this, there is an ambiguity regarding the order of application in the composition gh of two permutations: first apply g and then h, or vice versa. Both interpretations are fine. For concreteness, we assume the former.

Γ-orbit of $\pi(1)$. This can be achieved by multiplying π to the right with a permutation in Γ that maps $\pi(1)$ to m.

Next we apply the same process to π_1 but consider the behavior on the element 2 of $[n]$. Since we are no longer allowed to change the value of $\pi_1(1)$, which equals m, the canonization of the behavior on 2 can only use multiplication on the right with permutations in Γ_m, i.e., permutations in Γ that stabilize the element m. Doing so results in a permutation $\pi_2 \in \pi_1\Gamma$.

We repeat this process for all elements $k \in [n]$ in order. In the k-th step, we canonize the behavior on the element k by multiplying on the right with permutations in $\Gamma_{\pi_{k-1}([k-1])}$, i.e., permutations in Γ that pointwise stabilize all of the elements $\pi_{k-1}(\ell)$ for $\ell \in [k-1]$. ◀

Proof of Lemma 13. The number of canonical representatives modulo Γ in H equals the number of distinct (left) cosets of Γ in H, which is $|H|/|\Gamma|$. We construct an algorithm that takes as input a list of generators for Γ and H, and an index $i \in [|H|/|\Gamma|]$, and outputs the permutation σ that is the lexicographically i-th canonical representative modulo Γ in H.

The algorithm uses a prefix search to construct σ. In the k-th step, it knows the prefix $(\sigma(1), \sigma(2), \ldots, \sigma(k-1))$ of length $k-1$, and needs to figure out the correct value $v \in [n]$ to extend the prefix with. In order to do so, the algorithm needs to compute for each $v \in [n]$ the count c_v of canonical representatives modulo Γ in H that agree with σ on $[k-1]$ and take the value v at k. The following claims allow us to do that efficiently when given a permutation $\sigma_{k-1} \in H$ that agrees with σ on $[k-1]$. The claims use the notation $T_{k-1} \doteq \sigma_{k-1}([k-1])$, which also equals $\sigma([k-1])$.

▶ **Claim 17.** *The canonical representatives modulo Γ in H that agree with $\sigma \in H$ on $[k-1]$ are exactly the canonical representatives modulo $\Gamma_{T_{k-1}}$ in $\sigma_{k-1}H_{T_{k-1}}$.*

Proof. The following two observations imply Claim 17.

(i) A permutation $\pi \in H$ agrees with $\sigma \in H$ on $[k-1]$
$\Leftrightarrow$ π agrees with σ_{k-1} on $[k-1]$
$\Leftrightarrow$ $\sigma_{k-1}^{-1}\pi \in H_{T_{k-1}}$
$\Leftrightarrow$ $\pi \in \sigma_{k-1}H_{T_{k-1}}$.

(ii) Two permutations in $\sigma_{k-1}H_{T_{k-1}}$, say $\pi \doteq \sigma_{k-1}g$ and $\pi' \doteq \sigma_{k-1}g'$ for $g, g' \in H_{T_{k-1}}$, belong to the same left coset of Γ iff they belong to the same left coset of $\Gamma_{T_{k-1}}$. This follows because if $\sigma_{k-1}g' = \sigma_{k-1}gh$ for some $h \in \Gamma$, then h equals $g^{-1}g' \in H_{T_{k-1}}$, so $h \in \Gamma \cap H_{T_{k-1}} = \Gamma_{T_{k-1}}$. ◀

▶ **Claim 18.** *The count c_v for $v \in [n]$ is nonzero iff v is the minimum of some $\Gamma_{T_{k-1}}$-orbit contained in the $H_{T_{k-1}}$-orbit of $\sigma_{k-1}(k)$.*

Proof. The set of values of $\pi(k)$ when π ranges over $\sigma_{k-1}H_{T_{k-1}}$ is the $H_{T_{k-1}}$-orbit of $\sigma_{k-1}(k)$. Since $\Gamma_{T_{k-1}}$ is a subgroup of $H_{T_{k-1}}$, this orbit is the union of some $\Gamma_{T_{k-1}}$-orbits. Combined with Claim 17 and the construction of the canonical representatives modulo $\Gamma_{T_{k-1}}$, this implies Claim 18. ◀

▶ **Claim 19.** *If a count c_v is nonzero then it equals $|H_{T_{k-1}\cup\{v\}}|/|\Gamma_{T_{k-1}\cup\{v\}}|$.*

Proof. Since the count is nonzero, there exists a permutation $\sigma' \in H$ that is a canonical representative modulo Γ that agrees with σ_{k-1} on $[k-1]$ and satisfies $\sigma'(k) = v$. Applying Claim 17 with σ replaced by σ', k by $k' \doteq k+1$, T_{k-1} by $T'_k \doteq T_{k-1} \cup \{v\}$, and σ_{k-1} by any permutation $\sigma'_k \in H$ that agrees with σ' on $[k]$, yields Claim 19. This is because the number of canonical representatives modulo $\Gamma_{T'_k}$ in $\sigma'_k H_{T'_k}$ equals the number of (left) cosets of $\Gamma_{T'_k}$ in $H_{T'_k}$, which is the quantity stated in Claim 19. ◀

Algorithm 1

Input: positive integer n, $\Gamma \leq H \leq S_n$, $i \in [|H|/|\Gamma|]$
Output: lexicographically i-th canonical representative modulo Γ in H

$\sigma_0 \leftarrow id$
for $k = 1$ to n **do**
  $O_1, O_2, \ldots \leftarrow$ Γ-orbits contained in the H-orbit of $\sigma_{k-1}(k)$, in increasing order of $\min(O_i)$
  find integer ℓ such that $\sum_{j=1}^{\ell-1} c_{\min(O_j)} < i \leq \sum_{j=1}^{\ell} c_{\min(O_j)}$, where $c_v \doteq |H_v|/|\Gamma_v|$
  $i \leftarrow i - \sum_{i=1}^{\ell-1} c_{\min(O_j)}$
  $m \leftarrow \min(O_\ell)$
  find $\tau \in H$ such that $\tau(\sigma_{k-1}(k)) = m$
  $\sigma_k \leftarrow \sigma_{k-1}\tau$
  $H \leftarrow H_m$; $\Gamma \leftarrow \Gamma_m$
return σ_n

The algorithm builds a sequence of permutations $\sigma_0, \sigma_1, \ldots, \sigma_n \in H$ such that σ_k agrees with σ on $[k]$. It starts with the identity permutation $\sigma_0 = id$, builds σ_k out of σ_{k-1} for increasing values of $k \in [n]$, and outputs the permutation $\sigma_n = \sigma$.

Pseudocode for the algorithm is presented in Algorithm 1. Note that the pseudocode modifies the arguments Γ, H, and i along the way. Whenever a group is referenced in the pseudocode, the actual reference is to a list of generators for that group.

The correctness of the algorithm follows from Claims 18 and 19. The fact that the algorithm runs in polynomial time follows from Proposition 14. ◀

6 Future Directions

We end with a few directions for further research.

6.1 What about Minimum Circuit Size?

We suspect that our techniques also apply to MCSP in place of MKTP, but we have been unsuccessful in extending them to MCSP so far. To show our result for the complexity measure $\mu = \mathrm{KT}$, we showed the following property for polynomial-time samplable flat distributions R: There exists an efficiently computable bound $\theta(s,t)$ and a polynomial t such that if y is the concatenation of t independent samples from R, then

$$\mu(y) > \theta(s,t) \text{ holds with high probability if } R \text{ has entropy } s+1, \text{ and} \tag{4}$$

$$\mu(y) \leq \theta(s,t) \text{ always holds if } R \text{ has entropy } s. \tag{5}$$

We set $\theta(s,t)$ slightly below $\kappa(s+1,t)$ where $\kappa(s,t) \doteq st$. (4) followed from a counting argument, and (5) by showing that

$$\mu(y) \leq \kappa(s,t) \cdot \left(1 + \frac{n^c}{t^\alpha}\right) \tag{6}$$

always holds for some positive constants c and α. We concluded by observing that for a sufficiently large polynomial t the right-hand side of (6) is significantly below $\kappa(s+1,t)$.

Mimicking the approach with μ denoting circuit complexity, we set

$$\kappa(s,t) = \frac{st}{\log(st)} \cdot \left(1 + (2 - o(1)) \cdot \frac{\log\log(st)}{\log(st)}\right).$$

Then (4) follows from [31]. As for (5), the best counterpart to (6) we know of (see, e.g., [10]) is

$$\mu(y) \leq \frac{st}{\log(st)} \cdot \left(1 + (3 + o(1)) \cdot \frac{\log\log(st)}{\log(st)}\right).$$

However, in order to make the right-hand side of (6) smaller than $\kappa(s+1,t)$, t needs to be exponential in s.

One possible way around the issue is to boost the entropy gap between the two cases. This would not only show that all our results for MKTP apply to MCSP as well, but could also form the basis for reductions between different versions of MCSP (defined in terms of different circuit models, or in terms of different size parameters), and to clarify the relationship between MKTP and MCSP. Until now, all of these problems have been viewed as morally equivalent to each other, although no efficient reduction is known between *any* two of these, in either direction. Given the central role that MCSP occupies, it would be desirable to have a theorem that indicates that MCSP is fairly robust to minor changes to its definition. Currently, this is lacking.

On a related point, it would be good to know how the complexity of MKTP compares with the complexity of the KT-random strings: $R_{\mathrm{KT}} = \{x : \mathrm{KT}(x) \geq |x|\}$. Until now, all prior reductions from natural problems to MCSP or MKTP carried over to R_{KT}—but this would seem to require even stronger gap amplification theorems. The relationship between MKTP and R_{KT} is analogous to the relationship between MCSP and the special case of MCSP that is denoted MCSP′ in [23]: MCSP′ consists of truth tables f of m-ary Boolean functions that have circuits of size at most $2^{m/2}$.

6.2 Statistical Zero Knowledge

Allender and Das [2] generalized their result that $\mathsf{GI} \in \mathsf{RP}^{\mathrm{MKTP}}$ to $\mathsf{SZK} \subseteq \mathsf{BPP}^{\mathrm{MKTP}}$ by applying their approach to a known SZK-complete problem. Our proof that $\mathsf{GI} \in \mathsf{coRP}^{\mathrm{MKTP}}$ similarly generalizes to $\mathsf{SZK} \subseteq \mathsf{BPP}^{\mathrm{MKTP}}$. We use the problem Entropy Approximation, which is complete for SZK under oracle reductions [12, Lemma 5.1]:[2] Given a circuit C and a threshold θ with the promise that the distribution induced by C has entropy either at most $\theta - 1$ or else at least $\theta + 1$, decide whether the former is the case. By combining the Flattening Lemma [13] with our hashing-based generic encoding mentioned in the introduction, one can show that for any distribution of entropy s sampled by a circuit C, the concatenation of t random samples from C has, with high probability, KT complexity between $ts - t^{1-\alpha_0} \cdot \mathrm{poly}(|C|)$ and $ts + t^{1-\alpha_0} \cdot \mathrm{poly}(|C|)$ for some positive constant α_0. Along the lines of Remark 11, this allows us to show that Entropy Approximation, and hence all of SZK, is in $\mathsf{BPP}^{\mathrm{MKTP}}$.

We do not know how to eliminate the errors from those reductions: Is $\mathsf{SZK} \subseteq \mathsf{ZPP}^{\mathrm{MKTP}}$, or equivalently, is Entropy Approximation in $\mathsf{ZPP}^{\mathrm{MKTP}}$? Our approach yields that Entropy Approximation is in $\mathsf{coRP}^{\mathrm{MKTP}}$ (no false negatives) *when the input distributions are almost flat*, i.e., when the difference between the max- and min-entropy is small. However, it is not known whether that restriction of Entropy Approximation is complete for SZK[3] (Goldreich

[2] Entropy Approximation is complete for NISZK (Non-Interactive SZK) under *mapping* reductions [12]. Problems that are complete for SZK under mapping reductions include Statistical Difference [28] and Entropy Difference [13]. The mapping reduction from Statistical Difference to Entropy Difference in [14, Theorem 3] is similar to our reduction from Isomorphism Problems to MKTP.

[3] The Flattening Lemma [13] allows us to restrict to distributions that are almost flat in an *average-case*

and Vadhan, personal communication). Moreover, we do not see how to eliminate the false positives.

Trying to go beyond SZK, we mention that except for the possible use of the MKTP oracle in the construction of the probably-correct overestimator from condition 3 in Theorem 3, the reduction in Theorem 3 makes only one query to the oracle. It was observed in [18] that the reduction also works for any relativized KT problem MKTP^A (where the universal machine for KT complexity has access to oracle A). More significantly, [18] shows that any problem that is accepted with negligible error probability by a probabilistic reduction that makes only one query, relative to *every* set MKTP^A, must lie in $\mathsf{AM} \cap \mathsf{coAM}$. Thus, without significant modification, our techniques cannot be used in order to reduce any class larger than $\mathsf{AM} \cap \mathsf{coAM}$ to MKTP.

The property that only one query is made to the oracle was subsequently used in order to show that MKTP is hard for the complexity class DET under mapping reductions computable in nonuniform NC^0 [4]. Similar hardness results (but for a more powerful class of reducibilities) hold also for MCSP [25]. This has led to unconditional lower bounds on the circuit complexity of MKTP [4, 17], showing that MKTP does not lie in the complexity class $\mathsf{AC}^0[p]$ for any prime p; it is still open whether similar circuit lower bounds hold for MCSP.

Acknowledgments. We thank V. Arvind for helpful comments about the graph automorphism problem and rigid graphs, Alex Russell and Yoav Kallus for helpful ideas on encoding and decoding graphs, Laci Babai and Peter Brooksbank for answering questions about computational group theory, and Oded Goldreich and Salil Vadhan for answering questions about SZK. We also thank the anonymous reviewers for their helpful suggestions.

References

1 Eric Allender, Harry Buhrman, Michal Koucký, Dieter van Melkebeek, and Detlef Ronneburger. Power from random strings. *SIAM Journal on Computing*, 35:1467–1493, 2006.

2 Eric Allender and Bireswar Das. Zero knowledge and circuit minimization. In *Mathematical Foundations of Computer Science 2014*, pages 25–32, 2014.

3 Eric Allender, Joshua Grochow, Dieter van Melkebeek, Cristopher Moore, and Andrew Morgan. Minimum circuit size, graph isomorphism, and related problems. Technical Report TR17-158, Electronic Colloquium on Computational Complexity, 2017.

4 Eric Allender and Shuichi Hirahara. New insights on the (non)-hardness of circuit minimization and related problems. In *Proceedings of the 42nd International Symposium on Mathematical Foundations of Computer Science*, 2017. To appear.

5 Eric Allender, Michal Koucký, Detlef Ronneburger, and Sambuddha Roy. The pervasive reach of resource-bounded Kolmogorov complexity in computational complexity theory. *Journal of Computer and System Sciences*, 77:14–40, 2010.

6 László Babai. Graph isomorphism in quasipolynomial time. In *Proceedings of the 48th annual ACM Symposium on Theory of Computing*, pages 684–697, 2016.

7 László Babai, Paolo Codenotti, and Youming Qiao. Polynomial-time isomorphism test for groups with no abelian normal subgroups. In *Automata, Languages, and Programming*, pages 51–62, 2012.

sense, but we need almost flatness in the above *worst-case* sense. For example, consider a circuit C that induces a distribution of entropy less than $\theta - 1$ whose support contains all strings of length n where $n \gg \theta$. In that case, there is no nontrivial worst-case bound on the KT complexity of samples from C; with positive probability, t samples from C may have KT-complexity close to $t \cdot n \gg t \cdot (\theta + 1)$.

8 László Babai. Local Expansion of Vertex-transitive Graphs and Random Generation in Finite Groups. In *Proceedings of the 23rd Annual ACM Symposium on Theory of Computing*, pages 164–174, 1991.

9 Marco Carmosino, Russell Impagliazzo, Valentine Kabanets, and Antonina Kolokolova. Algorithms from natural lower bounds. In *Proceedings of the 31st Computational Complexity Conference*, pages 10:1–10:24, 2016.

10 Gudmund Skovbjerg Frandsen and Peter Bro Miltersen. Reviewing bounds on the circuit size of the hardest functions. *Information Processing Letters*, 95(2):354–357, 2005.

11 Oded Goldreich, Silvio Micali, and Avi Wigderson. Proofs that yield nothing but their validity for all languages in NP have zero-knowledge proof systems. *Journal of the ACM*, 38(3):691–729, 1991.

12 Oded Goldreich, Amit Sahai, and Salil Vadhan. Can statistical zero knowledge be made non-interactive? or on the relationship of SZK and NISZK. In *Advances in Cryptology — CRYPTO '99*, pages 467–484, 1999.

13 Oded Goldreich and Salil Vadhan. Comparing entropies in statistical zero knowledge with applications to the structure of SZK. In *Proceedings of the 14th Annual IEEE Conference on Computational Complexity*, pages 54–73, 1999.

14 Oded Goldreich and Salil Vadhan. On the complexity of computational problems regarding distributions. In Oded Goldreich, editor, *Studies in Complexity and Cryptography – Miscellanea on the Interplay between Randomness and Computation*, pages 13–29. Springer, 2011.

15 Joshua A. Grochow. Matrix Lie algebra isomorphism. In *Proceedings of the 2012 IEEE Conference on Computational Complexity*, pages 203–213, 2012.

16 Johan Håstad, Russell Impagliazzo, Leonid Levin, and Michael Luby. A pseudorandom generator from any one-way function. *SIAM Journal on Computing*, 28:1364–1396, 1999.

17 Shuichi Hirahara and Rahul Santhanam. On the average-case complexity of MCSP and its variants. In *Proceedings of the 32nd Computational Complexity Conference*, pages 7:1–7:20, 2017.

18 Shuichi Hirahara and Osamu Watanabe. Limits of minimum circuit size problem as oracle. In *Proceedings of the 31st Conference on Computational Complexity*, pages 18:1–18:20, 2016.

19 Derek F. Holt, Bettina Eick, and Eamonn A. O'Brien. *Handbook of Computational Group Theory*. Discrete Mathematics and its Applications. Chapman & Hall/CRC, 2005.

20 Valentine Kabanets and Jin-Yi Cai. Circuit minimization problem. In *Proceedings of the 32nd ACM Symposium on Theory of Computing*, pages 73–79, 2000.

21 Donald E. Knuth. *The Art of Computer Programming*, volume 3: Sorting and Searching. Addison-Wesley, 1973.

22 Johannes Köbler, Uwe Schöning, and Jacobo Torán. *The Graph Isomorphism Problem: Its Structural Complexity*. Birkhauser Verlag, Basel, Switzerland, Switzerland, 1993.

23 Cody D. Murray and R. Ryan Williams. On the (Non) NP-Hardness of Computing Circuit Complexity. *Theory of Computing*, 13:1–22, 2017.

24 Noam Nisan and Avi Wigderson. Hardness vs randomness. *Journal of Computer and System Sciences*, 49(2):149–167, 1994.

25 Igor C. Carboni Oliveira and Rahul Santhanam. Conspiracies Between Learning Algorithms, Circuit Lower Bounds, and Pseudorandomness. In *Proceedings of the 32nd Computational Complexity Conference*, pages 18:1–18:49, 2017.

26 Erez Petrank and Ron M. Roth. Is code equivalence easy to decide? *IEEE Transactions on Information Theory*, 43:1602–1604, 1997.

27 Michael Rudow. Discrete logarithm and minimum circuit size. *Information Processing Letters*, 128:1–4, 2017.

28 Amit Sahai and Salil Vadhan. A complete problem for statistical zero knowledge. *Journal of the ACM*, 50(2):196–249, 2003.

29 Ákos Seress. *Permutation group algorithms*, volume 152 of *Cambridge Tracts in Mathematics*. Cambridge University Press, Cambridge, 2003.

30 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.

31 Masaki Yamamoto. A tighter lower bound on the circuit size of the hardest Boolean functions. Technical Report TR11-086, Electronic Colloquium on Computational Complexity, 2011.

Foundations of Homomorphic Secret Sharing*†

Elette Boyle[‡1], **Niv Gilboa**[§2], **Yuval Ishai**[¶3], **Huijia Lin**[‖4], and **Stefano Tessaro**[**5]

1 **IDC Herzliya, Herzliya, Israel**
eboyle@alum.mit.edu
2 **Ben Gurion University, Be'er Sheva, Israel**
gilboan@bgu.ac.il
3 **Technion, Haifa, Israel**
yuvali@cs.technion.ac.il
4 **University of California Santa Barbara, Santa Barbara, CA, USA**
rachel.lin@cs.ucsb.edu
5 **University of California Santa Barbara, Santa Barbara, CA, USA**
tessaro@cs.ucsb.edu

Abstract

Homomorphic secret sharing (HSS) is the secret sharing analogue of homomorphic encryption. An HSS scheme supports a local evaluation of functions on shares of one or more secret inputs, such that the resulting shares of the output are short. Some applications require the stronger notion of *additive* HSS, where the shares of the output add up to the output over some finite Abelian group. While some strong positive results for HSS are known under specific cryptographic assumptions, many natural questions remain open.

We initiate a systematic study of HSS, making the following contributions.

- **A definitional framework.** We present a general framework for defining HSS schemes that unifies and extends several previous notions from the literature, and cast known results within this framework.
- **Limitations.** We establish limitations on *information-theoretic* multi-input HSS with short output shares via a relation with communication complexity. We also show that *additive* HSS for non-trivial functions, even the AND of two input bits, implies non-interactive key exchange, and is therefore unlikely to be implied by public-key encryption or even oblivious transfer.
- **Applications.** We present two types of applications of HSS. First, we construct 2-round protocols for secure multiparty computation from a simple constant-size instance of HSS. As a corollary, we obtain 2-round protocols with attractive asymptotic efficiency features under the Decision Diffie Hellman (DDH) assumption. Second, we use HSS to obtain nearly

* The views expressed are those of the authors and do not reflect the official policy or position of the Department of Defense, the National Science Foundation, or the U.S. Government.

† A full version of the paper is available at https://eprint.iacr.org/2017/1248.

‡ E. Boyle was supported by ISF grant 1861/16, AFOSR Award FA9550-17-1-0069, and ERC grants 307952, 742754.

§ N. Gilboa was supported by ISF grant 1638/15, a grant by the BGU Cyber Center by the European Union's Horizon 2020 ICT program (Mikelangelo project), and ERC grant 742754.

¶ Y. Ishai was supported by ERC grant 742754, NSF-BSF grant 2015782, BSF grant 2012366, ISF grant 1709/14, DARPA/ARL SAFEWARE award, NSF Frontier Award 1413955, NSF grants 1619348, 1228984, 1136174, and 1065276, a Xerox Faculty Research Award, a Google Faculty Research Award, an equipment grant from Intel, and an Okawa Foundation Research Grant. This material is based upon work supported by the DARPA through the ARL under Contract W911NF-15-C-0205.

‖ H. Lin was supported by NSF grants CNS-1528178, CNS-1514526, CNS-1652849 (CAREER), a Hellman Fellowship, the Defense Advanced Research Projects Agency (DARPA) and Army Research Office (ARO) under Contract No. W911NF-15-C-0236, and a subcontract No. 2017-002 through Galois.

** S. Tessaro was supported by NSF grants CNS-1553758 (CAREER), CNS-1423566, CNS-1719146, CNS-1528178, and IIS-1528041, and by an Alfred P. Sloan Research Fellowship.

optimal worst-case to average-case reductions in P. This in turn has applications to fine-grained average-case hardness and verifiable computation.

1998 ACM Subject Classification E.3.3 Public key cryptosystems

Keywords and phrases Cryptography, homomorphic secret sharing, secure computation, communication complexity, worst-case to average case reductions

Digital Object Identifier 10.4230/LIPIcs.ITCS.2018.21

1 Introduction

Fully homomorphic encryption (FHE) [53, 35] is a powerful cryptographic primitive that supports general computations on encrypted inputs. Despite intensive study, FHE schemes can only be based on a narrow class of cryptographic assumptions [56, 17, 36], which are all related to lattices, and their concrete efficiency leaves much to be desired.

In this paper we consider the following secret sharing analogue of FHE, referred to as *homomorphic secret sharing* (HSS) [14]. A standard (threshold) secret sharing scheme randomly splits an input x into m shares, $(x^1, \ldots, x^m)$, such that any set of t shares reveals nothing about the input. An HSS scheme supports computations on shared inputs by means of local computations on their shares. More concretely, there is a local evaluation algorithm Eval and decoder algorithm Dec satisfying the following homomorphism requirement. Given a description of a function F, the algorithm $\mathsf{Eval}(F; x^j)$ maps an input share x^j to a corresponding output share y^j, such that $\mathsf{Dec}(y^1, \ldots, y^m) = F(x)$.

An HSS scheme as above can be trivially obtained by letting Eval output (F, x^j) and Dec first reconstruct x from the shares and then compute F. Analogously to the output compactness requirement of FHE, we require that the HSS output shares be *compact* in the sense that their length depends only on the output length of F and the security parameter. In fact, it is often useful to make the more stringent requirement that Dec compute $F(x)$ as the sum $y^1 + \ldots + y^m$ in some finite Abelian group. We refer to such an HSS scheme as an *additive HSS*. We also consider a relaxed notion of *weak compactness* that allows the length of the output shares to grow sublinearly with the input size.

Finally, one can naturally consider a *multi-input* variant of HSS, where inputs $x_1, \ldots, x_n$ are independently shared, Eval locally maps the j-th shares of the n inputs to the j-th output share, and Dec outputs $F(x_1, \ldots, x_n)$. In fact, multi-input HSS is meaningful even when F is a fixed function rather than an input of Eval. For instance, one may consider additive 2-input HSS where F computes the AND of two input bits, or compact 2-input HSS where F takes an inner product of two input vectors.

HSS vs. FHE. HSS can generally be viewed as a relaxation of FHE that offers protection against bounded collusions. However, as observed in [14], in some applications of FHE it is possible to use HSS as an alternative that offers the same level of security. For instance, in the context of secure two-party computation [57, 40], using HSS to share the inputs of the two parties does not compromise security in any way, since the two parties together can anyway learn both inputs.

More importantly for this work, HSS can potentially offer several useful features that are inherently impossible for FHE. One such feature is *information-theoretic security*. Information-theoretic HSS schemes for multiplying two secrets with security threshold $t < m/2$ serve as the basis for information-theoretic protocols for secure multiparty computation [9, 20, 25].

Information-theoretic HSS schemes for certain classes of depth-2 circuits implicitly serve as the basis for the best known constructions of information-theoretic private information retrieval schemes and locally decodable codes [58, 29, 8]. Another potential feature of HSS is *optimal compactness*: if F has a single output bit, then the output shares y^j can be as short as a single bit. Indeed, special types of FHE schemes can be used to obtain additive HSS schemes with $t = m - 1$ that support general homomorphic computations with optimal compactness [27]. This feature is useful for several applications of HSS, including ones we discuss in this work.

Finally, recent works obtain HSS schemes that support rich classes of computations under the Decision Diffie Hellman (DDH) assumption [14, 16] or the security of the Paillier encryption scheme [30], which are not known to imply FHE. These constructions use very different techniques from those underlying known FHE constructions. This suggests a potential for further diversifying the assumptions and structures on which HSS can be based, which may potentially lead to more efficient substitutes for known FHE schemes.

1.1 Our Contribution

The current state of the art in HSS mostly consists of isolated positive results and leaves open some of the most basic questions. In this work we initiate a more systematic study of HSS, making the following contributions. We refer the reader to the relevant sections for a high level overview of the main ideas behind each contribution.

A definitional framework. We start, in Section 2, by presenting a general framework for HSS that unifies and extends several previous notions from the literature. In Section 3 we cast some known primitives and previous results within this framework. This includes a simple extension of a previous Learning With Errors (LWE)-based construction from [27] to the setting of multi-input HSS, whose details appear in full version.

Limitations. In Section 4 we establish two types of limitations on multi-input HSS. First, in Section 4.1, we show that weakly compact *information-theoretic* multi-input HSS schemes for security threshold $t \geq m/2$ shares do not exist for functions that have high (randomized, one-way) two-party communication complexity. This includes simple functions such as inner product or set disjointness. The high level idea is to obtain a low-communication two-party protocol from the HSS scheme by having the two parties use a common source of randomness to locally simulate the HSS input shares of both inputs, without any interaction, and then have one party send its HSS output share to the other. Second, in Section 4.2, we show that *additive* HSS for non-trivial functions, or even for computing the AND of two input bits, implies non-interactive key exchange (NIKE), a cryptographic notion which is not known to be implied by standard public-key primitives such as oblivious transfer. Loosely, two parties can simultaneously exchange HSS shares of input bits whose AND is zero, and output their HSS-evaluated output share as a shared key. This result provides some explanation for the difficulty of constructing strong types of HSS schemes from general assumptions.

Applications. In Section 5 we present two types of applications of HSS. First, in Section 5.1, we construct 2-round protocols for secure Multi-Party Computation (MPC) from a simple constant-size instance of additive HSS with $n = 3$ inputs and $m = 2$ shares, for computing $\mathsf{3Mult\text{-}Plus}((x_1, z_1), (x_2, z_2), (x_3, z_3)) = x_1x_2x_3 + z_1 + z_2 + z_3$. At a very high level, this reduction crucially relies on a randomized encoding of functions by degree-3 polynomials [2], to decompose the computation of an arbitrary function F into the computation of many

degree-3 monomials. The computation of each monomial is further decomposed into many invocation of HSS for 3Mult-Plus among only a constant number of parties. As a corollary, we can transform a previous DDH-based 2-round MPC protocol in [16] (which requires a public-key infrastructure) for only a constant number of parties, into a 2-round protocol for an arbitrary polynomial number of parties.

In the literature, 2-round MPC protocols exist *in the CRS model*, based on LWE (e.g., [3, 52]) and *in the plain model,* from indistinguishability obfuscation or witness encryption with NIZK (e.g., [32, 42]) or bilinear groups [33], or even 2-round semi-honest Oblivious Transfer (OT) protocols [34, 11]. Our protocol can be instantiated in the public-key infrastructure model under DDH, which is weaker than or incomparable to the feasibility results of other recent constructions. However, our protocols using HSS still have several advantages, in particular, they enjoy better asymptotic efficiency, and they are in the more general client-server model, where the input clients' computation can be done offline, and the output clients' computation is relatively cheap.

A second type of applications, presented in Section 5.2, is to obtaining worst-case to average-case reductions in P. Roughly speaking, the HSS evaluation function Eval for computing F defines a function $\hat{F}$ such that computing F on any given input x can be reduced to computing $\hat{F}$ on two or more inputs that are individually pseudorandom. A similar application of FHE was already pointed out in [24]. However, an advantage of the HSS-based reductions is that they allow $\hat{F}$ to have a single bit of output. Another advantage is the potential of diversifying assumptions. We discuss applications of the reductions implied by HSS to fine-grained average-case hardness and verifiable computation. In particular, the HSS-based approach yields checking procedures for polynomial-time computations that achieve better soundness vs. succinctness tradeoffs than any other approach we are aware of.

2 General Definitional Framework for HSS

In this section we give a general definition of homomorphic secret sharing (HSS) that can be instantiated to capture different notions from the literature. We consider multi-input HSS schemes that support a compact evaluation of a function F on shares of inputs $x_1, \ldots, x_n$ that originate from different clients. More concretely, each client i randomly splits its input x_i between m servers using the algorithm Share, so that x_i is hidden from any t colluding servers. We assume $t = m - 1$ by default. Each server j applies a local evaluation algorithm Eval to its share of the n inputs, and obtains an output share y^j. The output $F(x_1, \ldots, x_n)$ is reconstructed by applying a decoding algorithm Dec to the output shares $(y^1, \ldots, y^m)$.

To make HSS useful, we require that Dec be in some sense "simpler" than computing F. The most natural simplicity requirement, referred to as *compactness*, is that the output length of Eval, and hence the complexity of Dec, depend only on the output length of F and not on the input length of F. A more useful notion of simplicity is the stronger requirement of *additive* decoding, where the decoder computes the exclusive-or of the output shares or, more generally, adds them up in some Abelian group $\mathbb{G}$. We also consider weaker notions of simplicity that are are needed to capture HSS constructions from the literature.

Finally, for some of the main applications of HSS it is useful to let F and Eval take an additional input x_0 that is known to all servers. This is necessary for a meaningful notion of single-input HSS (with $n = 1$). Typically, the input x_0 will be a description of a function f applied to the input of a single client, e.g., a description of a circuit, branching program, or low-degree polynomial. The case of single-input HSS is considerably different from the case

of multi-input HSS with no server input. In particular, the negative results presented in this work do not apply to single-input HSS.

We now give our formal definition of general HSS. We refer the reader to Example 5 for an example of using this definition to describe an HSS scheme for multiplying two field elements using Shamir's secret sharing scheme. Here and in the following, we use the notation $\Pr[A_1; \ldots; A_m : E]$ to denote the probability that event E occurs following an experiment defined by executing the sequence $A_1, \ldots, A_m$ in order.

▶ **Definition 1** (HSS). An n-client, m-server, t-secure homomorphic secret sharing scheme for a function $F : (\{0,1\}^*)^{n+1} \to \{0,1\}^*$, or (n, m, t)-*HSS* for short, is a triple of PPT algorithms $(\mathsf{Share}, \mathsf{Eval}, \mathsf{Dec})$ with the following syntax:

- $\mathsf{Share}(1^\lambda, i, x)$: On input 1^λ (security parameter), $i \in [n]$ (client index), and $x \in \{0,1\}^*$ (client input), the sharing algorithm Share outputs m input shares, $(x^1, \ldots, x^m)$. By default, we require Share to run in (probabilistic) polynomial time in its input length; however, we also consider a relaxed notion of efficiency where Share is given the total length ℓ of *all* $n+1$ inputs (including x_0) and may run in time $\mathsf{poly}(\lambda, \ell)$.
- $\mathsf{Eval}\big(j, x_0, (x_1^j, \ldots, x_n^j)\big)$: On input $j \in [m]$ (server index), $x_0 \in \{0,1\}^*$ (common server input), and $x_1^j, \ldots, x_n^j$ (jth share of each client input), the evaluation algorithm Eval outputs $y^j \in \{0,1\}^*$, corresponding to server j's share of $F(x_0; x_1, \ldots, x_n)$.
- $\mathsf{Dec}(y^1, \ldots, y^m)$: On input $(y^1, \ldots, y^m)$ (list of output shares), the decoding algorithm Dec computes a final output $y \in \{0,1\}^*$.

The algorithms $(\mathsf{Share}, \mathsf{Eval}, \mathsf{Dec})$ should satisfy the following requirements:

- **Correctness:** For any $n+1$ inputs $x_0, \ldots, x_n \in \{0,1\}^*$,

$$\Pr\left[\begin{array}{l} \forall i \in [n]\ (x_i^1, \ldots, x_i^m) \leftarrow \mathsf{Share}(1^\lambda, i, x_i) \\ \forall j \in [m]\ y^j \leftarrow \mathsf{Eval}\big(j, x_0, (x_1^j, \ldots, x_n^j)\big) \end{array} : \mathsf{Dec}(y^1, \ldots, y^m) = F(x_0; x_1, \ldots, x_n)\right] = 1.$$

  Alternatively, in a *statistically correct* HSS the above probability is at least $1 - \mu(\lambda)$ for some negligible μ and in a δ-correct HSS (or δ-HSS for short) it is at least $1 - \delta - \mu(\lambda)$. In the case of δ-HSS the error parameter δ may be given as an additional input to Eval, and the running time of Eval is allowed to grow polynomially with $1/\delta$.
- **Security:** Consider the following semantic security challenge experiment for corrupted set of servers $T \subset [m]$:
  1: The adversary gives challenge index and inputs $(i, x, x') \leftarrow \mathcal{A}(1^\lambda)$, with $|x| = |x'|$.
  2: The challenger samples $b \leftarrow \{0,1\}$ and $(x^1, \ldots, x^m) \leftarrow \mathsf{Share}(1^\lambda, i, \tilde{x})$, where $\tilde{x} = x$ if $b = 0$ and $\tilde{x} = x'$ if $b = 1$.
  3: The adversary outputs a guess $b' \leftarrow \mathcal{A}((x^j)_{j \in T})$, given the shares for corrupted T.

  Denote by $\mathsf{Adv}(1^\lambda, \mathcal{A}, T) := \Pr[b = b'] - 1/2$ the advantage of $\mathcal{A}$ in guessing b in the above experiment. For circuit size bound $S = S(\lambda)$ and advantage bound $\alpha = \alpha(\lambda)$, we say that an (n, m, t)-HSS scheme $\Pi = (\mathsf{Share}, \mathsf{Eval}, \mathsf{Dec})$ is (S, α)-secure if for all $T \subset [m]$ of size $|T| \le t$, and all non-uniform adversaries $\mathcal{A}$ of size $S(\lambda)$, we have $\mathsf{Adv}(1^\lambda, \mathcal{A}, T) \le \alpha(\lambda)$. We say that Π is:
  - *computationally secure* if it is $(S, 1/S)$-secure for all polynomials S;
  - *statistically α-secure* if it is (S, α)-secure for all S;
  - *statistically secure* if it statistically α-secure for some negligible $\alpha(\lambda)$;
  - *perfectly secure* if it is statistically 0-secure.

▶ **Remark 2** (Unbounded HSS). *Definition 1 treats the number of inputs n as being fixed. We can naturally consider an unbounded multi-input variant of HSS where F is defined over*

arbitrary sequences of inputs x_i, and the correctness requirement is extended accordingly. We denote this flavor of multi-input HSS by $(*, m, t)$-HSS.

▶ **Remark 3** (Robust decoding)**.** *Definition 1 allows* Dec *to use all output shares for decoding the output. When $t < m-1$, one can consider a stronger variant of HSS where* Dec *can recover the output from any $t+1$ output shares. Such a robust notion of threshold homomorphic encryption was recently considered in [44]. In this work we do not consider robust decoding.*

2.1 Notions of Simple Decoding

As discussed above, to make HSS useful we impose two types of simplicity requirements on Dec. The most stringent requirement is that Dec adds its input over some Abelian group $\mathbb{G}$. We refer to such an HSS scheme as being *additive*. Note that any HSS scheme where Dec computes a fixed linear combination of the output shares (over some finite field) can be converted into an additive scheme by letting Eval multiply its outputs by the coefficients of the linear combination. See Example 5 for a relevant concrete example.

A more liberal requirement is *compactness*, which says that the length of the output shares depends only on the output length and the security parameter, independently of the input length. Finally, we also consider a further relaxation that we call *weak compactness*, requiring that the length of the output shares be sublinear in the input length when the input length is sufficiently bigger than the security parameter and the output length. This weaker notion is needed to capture some HSS constructions from the literature, and is used for making our negative results stronger. We formalize these notions below.

▶ **Definition 4** (Additive and compact HSS)**.** We say that an (n, m, t)-HSS scheme $\Pi = (\mathsf{Share}, \mathsf{Eval}, \mathsf{Dec})$ for F is:

- *Additive* if Dec outputs the exclusive-or of the m output shares, or $\mathbb{G}$-additive if Dec computes addition in an Abelian group $\mathbb{G}$;
- *Compact* if there is a polynomial p such that for every $\lambda, \ell_{\mathsf{out}}$, and inputs $x_0, x_1, \ldots, x_n \in \{0,1\}^*$ such that $|F(x_0; x_1, \ldots, x_n)| = \ell_{\mathsf{out}}$, the length of each output share obtained by applying Share with security parameter λ and then Eval is at most $p(\lambda) \cdot \ell_{\mathsf{out}}$ (or $O(\ell_{\mathsf{out}})$ for perfect or statistically α-secure HSS with a constant α);
- *Weakly compact* if there is a polynomial p and sublinear function $g(\ell) = o(\ell)$, such that for every $\lambda, \ell_{\mathsf{in}}, \ell_{\mathsf{out}}$, and inputs $x_0, x_1, \ldots, x_n \in \{0,1\}^*$ of total length ℓ_{in} with $|F(x_0; x_1, \ldots, x_n)| = \ell_{\mathsf{out}}$, the length of each output share obtained by applying Share with security parameter λ and then Eval is at most $g(\ell_{\mathsf{in}}) + p(\lambda) \cdot \ell_{\mathsf{out}}$ (or $g(\ell_{\mathsf{in}}) + O(\ell_{\mathsf{out}})$ for perfect or statistically α-secure HSS with a constant α). More generally, we can specify the precise level of compactness by referring to an HSS scheme as being $g(\lambda, \ell_{\mathsf{in}}, \ell_{\mathsf{out}})$-compact.

2.2 Default Conventions

It is convenient to make the following default choices of parameters and other conventions.

- We assume $t = m - 1$ by default and write (n, m)-HSS for $(n, m, m-1)$-HSS.
- We assume computational security by default, and refer to the statistical and perfect variants collectively as "information-theoretic HSS."
- In the case of *perfectly secure* or *statistically α-secure* HSS, λ is omitted.
- For $n \geq 2$ clients, we assume by default that the servers have no input and write $F(x_1, \ldots, x_n)$, omitting the server input x_0. Note that (n, m, t)-HSS with server input can be reduced to $(n+1, m, t)$-HSS with no server input by letting the server input be shared by one of the clients.

- We consider *additive HSS* by default. This stronger notion is useful for several application of HSS, and most HSS constructions realize it.
- We will sometimes be interested in additive HSS for a constant-size (finite) function F, such as the AND of two bits; this can be cast into Definition 1 by just considering an extension $\hat{F}$ of F that outputs 0 on all invalid inputs. Note that our two notions of compactness are not meaningful for a constant-size F. We can similarly handle functions F that impose restrictions on the relation between the lengths of different inputs. Since Eval can know all inputs lengths, we can ensure that Dec outputs 0 in case of mismatch.
- As noted above, the common server input x_0 is often interpreted as a "program" P from a class of programs $\mathcal{P}$ (e.g., circuits or branching programs), and F is the universal function defined by $F(P; x_1, \ldots, x_n) = P(x_1, \ldots, x_n)$. We refer to this as *HSS for the class* $\mathcal{P}$.

2.3 HSS with Setup

When considering multi-input HSS schemes, known constructions require different forms of *setup* to coordinate between clients. This setup is generated by a PPT algorithm Setup and is reusable, in the sense that the same setup can be used to share an arbitrary number of inputs. We consider the following types of setup:

- *No setup*: This is the default notion of HSS defined above.
- *Common random string (CRS) setup*: An algorithm $\mathsf{Setup}(1^\lambda)$ is used to generate a uniformly random string σ which is given as input to Share, Eval, and Dec.
- *Public-key setup:* We consider here a strong form of public-key setup in which $\mathsf{Setup}(1^\lambda)$ outputs a public key pk and m secret evaluation keys $(\mathsf{ek}_1, \ldots, \mathsf{ek}_m)$, where each key is given to a different server. The algorithm Share is given pk as an additional input, and $\mathsf{Eval}(j, \ldots)$ is given ek_j as an additional input. The security game is changed by giving both the adversary and the challenger pk and giving to the adversary $(\mathsf{ek}_j)_{j \in T}$ in addition to $(x^j)_{j \in T}$. Following the terminology from [16], we refer to HSS with this type of setup as *public-key* $(*, m, t)$*-HSS*.

3 Constructions

In this section we present positive results on HSS that are either implicit in the literature or can be easily obtained from known results. We cast these results in terms of the general HSS framework from Section 2.

We start with a detailed example for casting Shamir's secret sharing scheme [54] over a finite field $\mathbb{F}$ as a perfectly secure, $\mathbb{F}$-additive $(2, m, t)$-HSS scheme for the function F that multiplies two field elements. Such a scheme exists if and only if $m > 2t$.

▶ **Example 5** (Additive $(2, m, t)$-HSS for field multiplication). *Let m, t be parameters such that $m > 2t$, let $\mathbb{F}$ be a finite field with $|\mathbb{F}| > m$, let $\theta_1, \ldots, \theta_m$ be distinct nonzero field elements, and let $\lambda_1, \ldots, \lambda_n$ be field elements ("Lagrange coefficients") such that for any univariate polynomial p over $\mathbb{F}$ of degree at most $2t$ we have $p(0) = \sum_{j=1}^{m} \lambda_j p(\theta_j)$. Let $F : \mathbb{F} \times \mathbb{F} \to \mathbb{F}$ be the (constant-size) function defined by $F(x_1, x_2) = x_1 \cdot x_2$. A perfectly secure, additive $(2, m, t)$-HSS scheme for F is defined by the following algorithms. (Since F is a constant-size function we are not concerned with efficiency; we also omit x_0 since there is no server input and omit the security parameter λ since security is perfect.)*

1. $\mathsf{Share}(i, x)$*: pick $r_1, \ldots, r_t$ uniformly at random from $\mathbb{F}$ and let $p(Z) = x + r_1 Z + r_2 Z^2 + \ldots + r_t Z^t$ be a random polynomial of degree at most t with x as its free coefficient. Output $(p(\theta_1), \ldots, p(\theta_m))$. Note that* Share *does not depend on i (the inputs are shared the same).*

2. $\mathsf{Eval}\big(j, (x_1^j, x_2^j)\big)$: *Output* $\lambda_j \cdot x_1^j x_2^j$.
3. $\mathsf{Dec}(y^1, \ldots, y^m)$: *Output* $y^1 + \ldots + y^m$.

We now survey some other instances of HSS schemes from the literature.

- Additive m-out-of-m secret sharing over an Abelian group $\mathbb{G}$ is a $\mathbb{G}$-additive, perfectly secure $(*, m)$-HSS for the function $F(x_1, \ldots, x_n) = x_1 + \ldots + x_n$ where $x_i \in \mathbb{G}$. This is the first instance of HSS considered in the literature [10].
- Generalizing Example 5, multiplicative secret sharing [25] over a finite field $\mathbb{F}$ is an $\mathbb{F}$-additive, perfectly secure $(2, m, t)$-HSS for the function F that multiplies two field elements. Such schemes exist if and only if $m > 2t$. Multiplicative secret sharing schemes such as Shamir's scheme serve as the basis for secure multiparty computation protocols in the information-theoretic setting [9, 20]. More generally, information-theoretic $\mathbb{F}$-additive (d, m, t)-HSS for multiplying d elements of $\mathbb{F}$ exists if and only if $m > dt$ [7]. Multiplicative schemes with a smaller threshold t that work over a constant-size field (independent of m) can be based on algebraic geometric codes [21]. Efficient multiplicative schemes that support a pointwise multiplication of two vectors are considered in [31, 19].
- A 1-round k-server private information retrieval (PIR) scheme [22, 23] can be seen as a *weakly compact* $(1, k, 1)$-HSS for the selection function $F(D; \gamma) = D_\gamma$. For the 2-server case ($k = 2$), information theoretic PIR schemes provably cannot achieve our stronger notion of compactness unless the share size is linear in $|D|$ [39, 47]. Moreover, current schemes only realize our relaxed notion of efficiency for Share, since the share size is super-polynomial in $|\gamma|$ (see [28] for the best known construction in terms of total size of input shares and output shares). In the computational case, there are in fact *additive* 2-server schemes based on the existence of one-way functions, where Share satisfies the default strict notion of efficiency (see [15] for the best known construction).
- Non-trivial instances of compact, perfectly-secure (1,3,1)-HSS for certain classes of depth-2 boolean circuits [8] implicitly serve as the basis for the best known constructions of information-theoretic 3-server PIR schemes and 3-query locally decodable codes [58, 29].
- The main result of [14] is a construction of (single-input, computationally secure, additive) $(1, 2)$-δ-HSS for branching programs under the DDH assumption. The same paper also obtains a public-key $(*, 2)$-δ-HSS variant of this result. Similar results assuming the circular security of the Paillier encryption were recently obtained in [30].
- The notion of function secret sharing (FSS) from [13] is dual to the notion of HSS for a program class $\mathcal{P}$. It can be cast as an additive $(1, m)$-HSS for the universal function $F(x; P) = P(x)$, where $P \in \mathcal{P}$ is a program given as input to the client and x is the common server input. The special case of distributed point function (DPF) [37] is FSS for the class of point functions (namely, functions that have nonzero output for at most one input). DPF can be seen as additive $(1, m)$-HSS for the function $F(x; (\alpha, \beta))$ that outputs β if $x = \alpha$ and outputs 0 otherwise. It is known that one-way functions are necessary and sufficient for DPF [37].
- We observe that additive[1] $(*, m)$-HSS for *circuits* with statistical correctness can be obtained from the Learning With Errors (LWE) assumption, by a simple variation of the FSS construction from spooky encryption of [27] (more specifically, their techniques for obtaining 2-round MPC). The share size in this construction must grow with the circuit

[1] If one settles for the weaker notion of compactness, then single-input HSS can be trivially obtained from any FHE scheme by letting Share include an encryption of the input in one of the shares and split the decryption key into m shares.

depth, hence Share only satisfies the relaxed notion of efficiency; this dependence can be eliminated by relying on a stronger variant of LWE that involves circular security. We provide details of the underlying tools and construction in the full version.

We note that a key feature of HSS is that Dec does not require a secret key. This rules out nontrivial instances of single-server HSS. In particular, single-server PIR [49] and fully homomorphic encryption [35] cannot be cast as instances of our general definitional framework of HSS.

4 Limitations

In this section, we discuss some inherent limitations in HSS. First, in Section 4.1, we show lower bounds on the length of output shares in statistically-secure HSS using communication complexity lower bounds. In Section 4.2, we show that additive $(2, 2)$-HSS for the AND of two bits implies non-interactive key-exchange (NIKE). Given what is known about NIKE (in particular it only follows from non-generic assumptions, and it is not known to be implied directly by public-key encryption or OT), this gives a strong justification for the lack of instantiations from generic assumptions.

4.1 Lower Bounds for Statistically-Secure Multi-Input HSS

We show lower bounds on the length of output shares in statistically-secure multi-input HSS using lower bounds from communication complexity. The key step is to derive a public-coin two-party protocol to compute a function F from an HSS scheme for the function F, and such that the communication cost of the resulting protocol only depends on the length of the output shares.

Communication complexity refresher. We consider public-coin interactive protocols Π between two parties, Alice and Bob, who start the execution with respective inputs $x \in \mathcal{X}$ and $y \in \mathcal{Y}$, and common random tape R. (We can assume wlog that the protocol is otherwise deterministic, and all random coins come from R.) At any point in the execution, one of the parties can return an output value, denoted $\Pi(R, x, y)$. The cost of Π is the maximum number of bits exchanged by Alice and Bob, taken as the worst case over all possible inputs x, y, and random tapes R. We also say that such a protocol is *one-way* (or *one round*) if only one message is sent, and this goes from Alice to Bob.

We are interested in the inherent cost of a protocol Π that evaluates a function $F : \mathcal{X} \times \mathcal{Y} \to \mathcal{Z}$. In particular, the *(randomized) communication complexity of F with error ϵ*, denoted $\mathsf{R}_\epsilon(F)$, is the minimum cost of a public-coin protocol Π such that $\Pr[\Pi(R, x, y) \neq F(x, y)] \leq \epsilon$ for all x, y, where the probability is over the public random string R. If we restrict ourselves to one-way protocols, then we define analogously the *one-way communication complexity of F with error ϵ*, denoted $\mathsf{R}_\epsilon^{A\to B}(F)$. It is clear that $R_\epsilon^{A\to B}(F) \geq R_\epsilon(F)$.

The following are classical examples of lower bounds on the (one-way) randomized communication complexity.

▶ **Theorem 6** (e.g., [50]). *Let* $\mathsf{IP}_\ell : \{0,1\}^\ell \times \{0,1\}^\ell \to \{0,1\}$ *be such that* $\mathsf{IP}_\ell(x, y) = \sum_{i=1}^{\ell} x_i y_i \pmod 2$. *Then,* $\mathsf{R}_{1/3}(\mathsf{IP}_\ell) = \Omega(\ell)$.

▶ **Theorem 7** ([45]). *Let* $\mathsf{DISJ}_\ell : \{0,1\}^\ell \times \{0,1\}^\ell \to \{0,1\}$ *be such that* $\mathsf{DISJ}_\ell(x, y) = \neg \bigvee_{i=1}^{\ell}(x_i \wedge y_i)$. *Then,* $\mathsf{R}_{1/3}(\mathsf{DISJ}_\ell) = \Omega(\ell)$.

▶ **Theorem 8** ([48]). *Let* $\mathsf{INDEX}_\ell : \{0,1\}^\ell \times [\ell] \to \{0,1\}$ *be such that* $\mathsf{INDEX}_\ell(x_1 x_2 \ldots x_\ell, i) = x_i$. *Then,* $\mathsf{R}_{1/3}^{A\to B}(\mathsf{INDEX}_\ell) = \Omega(\ell)$.

Lower bounds on the length of output shares. We start with a lower bound on the length of the output shares in $(2,2)$-HSS. (Recall that (n,m)-HSS is a shorthand for $(n,m,t=m-1)$-HSS.) Below, we extend the technique to more general settings.

Recall that a $(2,2)$-HSS scheme is defined for a function $F:(\{0,1\}^*)^2 \to \{0,1\}^*$. (In this section, we consider the case where the servers have no input x_0, but the results extend straightforwardly to handle server inputs.) For any two integers $\ell_{1,\mathsf{in}}, \ell_{2,\mathsf{in}}$, it is convenient to define the restriction $F^{\ell_{1,\mathsf{in}},\ell_{2,\mathsf{in}}} : \{0,1\}^{\ell_{1,\mathsf{in}}} \times \{0,1\}^{\ell_{2,\mathsf{in}}} \to \{0,1\}^*$ such that $F^{\ell_{1,\mathsf{in}},\ell_{2,\mathsf{in}}}(x_1,x_2) = F(x_1,x_2)$. Also, for a suitable function g, we say that a $(2,2)$-HSS scheme is *g-compact* if, for security parameter λ, when the two inputs have lengths $\ell_{1,\mathsf{in}}$ and $\ell_{2,\mathsf{in}}$, respectively, the output shares have length each at most $g(\lambda, \ell_{1,\mathsf{in}}, \ell_{2,\mathsf{in}})$.

▶ **Proposition 9** (Compactness lower bound). *Let* $(\mathsf{Share}, \mathsf{Eval}, \mathsf{Dec})$ *be a* $(2,2)$*-HSS scheme for a function* $F:(\{0,1\}^*)^2 \to \{0,1\}^*$*, which is statistically* α*-secure,* g*-compact, and* δ*-correct. Then, for all* λ*, and* $\ell_{1,\mathsf{in}}, \ell_{2,\mathsf{in}} > 0$*,* $g(\lambda, \ell_{1,\mathsf{in}}, \ell_{2,\mathsf{in}}) \geq \mathsf{R}^{A\to B}_{\delta(\lambda)+4\alpha(\lambda)}(F^{\ell_{1,\mathsf{in}},\ell_{2,\mathsf{in}}})$.

We defer a proof to the full version, and only give a sketch here. It is easy to give a protocol for Alice and Bob to evaluate $F^{\ell_{1,\mathsf{in}},\ell_{2,\mathsf{in}}}$ on their respective inputs x_1, x_2: Bob runs $(x_2^1, x_2^2) \leftarrow \mathsf{Share}(1^\lambda, 2, x_2)$, and sends x_2^1 to Alice. Alice then runs $(x_1^1, x_1^2) \leftarrow \mathsf{Share}(1^\lambda, 1, x_1)$ and $y_1 \leftarrow \mathsf{Eval}(1^\lambda, 1, (x_1^1, x_2^1))$, and sends (x_1^2, y_1) to Bob. Finally, Bob computes $y_2 \leftarrow \mathsf{Eval}(1^\lambda, 2, (x_1^2, x_2^2))$, as well as the output $y \leftarrow \mathsf{Dec}(1^\lambda, y_1, y_2)$. However, we would like to make the protocol complexity independent of the input shares – this can be achieved by exploiting HSS security, as well as reverse sampling. Namely, we generate the shares x_1^2 and x_2^1 by running $\mathsf{Share}(1^\lambda, 1, 0^{\ell_{1,\mathsf{in}}})$ and $\mathsf{Share}(1^\lambda, 2, 0^{\ell_{2,\mathsf{in}}})$, respectively, and make these shares part of the common randomness. Then, when Alice and Bob need the shares x_1^1 and x_2^2, respectively, they each exploit knowlege of their respective inputs x_1 and x_2 to locally sample a share consistent with the input and the other share being equal the pre-sampled share in the common random tape. HSS security implies that the distribution of the resulting shares is close to the correct one, and the new protocol only has Alice send y_1 to Bob.

As an application, consider any statistically secure $(2,2)$-HSS scheme for inner products, i.e., for the function $\mathsf{IP} : (\{0,1\}^*)^2 \to \{0,1\}$ such that $\mathsf{IP}(x_1,x_2) = \mathsf{IP}_\ell(x_1,x_2)$ whenever $|x_1| = |x_2| = \ell$. Then, the following corollary implies that such scheme cannot be weakly compact. Similar lower bounds can be obtained for disjointness, and for the index function.

▶ **Corollary 10.** *There exists no weakly compact, statistically* $1/24$*-secure* $1/6$*-correct* $(2,2)$*-HSS scheme for* IP*.*

Proof. Apply Proposition 9 with $\ell_{1,\mathsf{in}} = \ell_{2,\mathsf{in}} = \ell$, $\delta = 1/6$, and $\alpha = \frac{1}{24}$. Regardless of the security parameter, the length of the output shares must be at least $\mathsf{R}^{A\to B}_{1/3}(\mathsf{IP}_\ell) = \Omega(\ell_{1,\mathsf{in}}+\ell_{2,\mathsf{in}})$ by Theorem 6, and this violates weak compactness. ◀

Extensions. Proposition 9 can be extended to obtain lower bounds for general (n,m,t)-HSS where $m, n \geq 2$ and $t \geq m/2$. We briefly summarize the main ideas here.

- $(n,2)$*-HSS.* For any n-ary function $F:(\{0,1\}^*)^n \to \{0,1\}^*$, we can define a two-party function as follows. Fix $k \in \{1,\ldots,n-1\}$, as well as Alice's indices $I_1 = (a_1,\ldots,a_k)$, and Bob's indices $I_2 = (b_1,\ldots,b_{n-k})$, where $\{a_1,\ldots,a_k,b_1,\ldots,b_{n-k}\} = [n]$. Then, $F'((x_{a_1},\ldots,x_{a_k}),(x_{b_1},\ldots,x_{b_{n-k}})) = F(x_1,\ldots,x_n)$. The proof of Proposition 9 can be adapted to lower bound the length of the output shares in an $(n,2)$-HSS scheme for F via $\mathsf{R}^{A\to B}(F')$, noting one would then choose the sets I_1, I_2 to maximize communication complexity of the resulting F'.

- (n, m, t)-*HSS for* $t \geq m/2$. A lower bound for $(n, 2)$-HSS extends straightforwardly to a lower bound for (n, m, t)-HSS where $t \geq m/2$, since the latter type of HSS implies the former type, by simply having one of the two servers in the $(n, 2)$-HSS simulate $m/2 \leq m_1 \leq t$ servers from the (n, m, t)-HSS scheme, and the other simulate the remaining $m_2 = m - m_1$ servers.
- *Simultaneous messages.* In the case of (n, m)-HSS, where $n \geq m$, we can alternatively obtain useful lower bounds via communication complexity in the *simultaneous message model* [48, 5], where m players send a message to a referee that decides the output. Roughly, a variant of the proof of Proposition 9 would build a protocol where the messages sent are exactly the m servers' output shares.

4.2 Additive Multi-Input HSS Implies Non-Interactive Key Exchange

It is known that roughly any non-trivial additive HSS (even for a single input) implies the existence of one-way functions [37, 13]; in turn, one-way functions have been shown to imply additive $(1, 2)$- and $(1, m)$-HSS for certain classes of simple functions [23, 37, 13, 15]. However, to date, all constructions of additive HSS supporting *multiple* inputs rely on a select list of heavily structured assumptions: DDH, LWE, Paillier, and obfuscation [14, 27, 30]. A clear challenge is whether one can instantiate such an object from weaker general assumptions, such as one-way functions, public-key encryption, or oblivious transfer.

We show that this is unlikely to occur. We demonstrate the power of additive multi-input HSS by proving that even the minimal version of $(2, 2)$-additive-HSS for the AND of two input bits already implies the existence of *non-interactive key exchange (NIKE)* [26], a well-studied cryptographic notion whose known constructions similarly are limited to select structured assumptions. NIKE is black-box separated from one-way functions and highly unlikely to be implied by generic public-key encryption or oblivious transfer.

On the other hand, we observe that $(2, 2)$-additive-HSS for AND is unlikely to be implied *by* NIKE, as the primitive additionally implies the existence of 2-message oblivious transfer (OT) [14], unknown to follow from NIKE alone.

We first recall the definition – for a two-party protocol Π between Alice and Bob, we denote by $\mathsf{out}_A(\Pi)$ and $\mathsf{out}_B(\Pi)$ their respective outputs, and $\mathsf{Transc}(\Pi)$ the resulting transcript.

▶ **Definition 11** (NIKE)**.** A 2-party protocol Π with single-bit output is a secure *non-interactive key-exchange (NIKE)* protocol if the following conditions hold:

- **Non-Interactive:** The protocol Π consists of exchanging a single (simultaneous) message.
- **Correctness:** The parties agree on a consistent output bit: $\Pr[\mathsf{out}_A(\Pi) = \mathsf{out}_B(\Pi)] = 1$, over randomness of Π.
- **Security:** There exists a negligible function ν such that for any non-uniform polynomial-time E, for every $\lambda \in \mathbb{N}$, it holds $\Pr[b \leftarrow E(1^\lambda, \mathsf{Transc}(\Pi)) : b = \mathsf{out}_A(\Pi)] \leq 1/2 + \nu(\lambda)$, where probability is taken over the randomness of Π and E.

▶ **Proposition 12.** *The existence of additive* $(2, 2)$*-HSS for the AND function* $F : \{0, 1\}^2 \to \{0, 1\}$ *defined by* $F(x_1, x_2) = x_1 x_2$ *implies the existence of non-interactive key exchange.*

Proof. Consider the candidate NIKE protocol given in Figure 1.

Non-interactive: By construction, the protocol consists of a single communication round.

Correctness: Follows by the additive decoding correctness of the $(2, 2)$-HSS for AND. Namely, with probability 1, it holds $z^A + z^B = 0 \in \{0, 1\}$; that is, $z^A = z^B$.

Security: Suppose there exists a polynomial-time eavesdropper E who, given the transcript of the protocol x^B, y^A succeeds in predicting Bob's output bit $z^B = \mathsf{Eval}(B(x^B, y^B))$ with

Communication Round:

- Alice samples shares of 0: i.e., $(x^A, x^B) \leftarrow \mathsf{Share}(1^\lambda, A, 0)$.
  Send x^B to Bob.
- Bob samples a random bit $b \leftarrow \{0,1\}$ and shares b: $(y^A, y^B) \leftarrow \mathsf{Share}(1^\lambda, B, b)$.
  Send y^A to Alice.

Output round:

- Alice outputs $z^A = \mathsf{Eval}(A, (x^A, y^A)) \in \{0,1\}$.
- Bob outputs $z^B = \mathsf{Eval}(B, (x^B, y^B)) \in \{0,1\}$.

Figure 1 NIKE protocol from any additive $(2,2)$-HSS for AND.

advantage α: i.e.,

$$Pr\left[\begin{array}{c} b \leftarrow 0,1; \\ (x^A, x^B) \leftarrow \mathsf{Share}(1^\lambda, A, 0); \\ (y^A, y^B) \leftarrow \mathsf{Share}(1^\lambda, B, b); \\ b' \leftarrow E(1^\lambda, (x^B, y^A)) \end{array} : b' = \mathsf{Eval}(B, (x^B, y^B))\right] \geq 1/2 + \alpha(\lambda).$$

We prove in such case α must be negligible, via the following two claims.

▶ **Claim 13.** *E must succeed with advantage α if Alice shares 1 instead of 0: Explicitly, there exists a negligible function ν_1 for which*

$$Pr\left[\begin{array}{c} b \leftarrow 0,1; \\ (x^A, x^B) \leftarrow \mathsf{Share}(1^\lambda, A, 1); \\ (y^A, y^B) \leftarrow \mathsf{Share}(1^\lambda, B, b); \\ b' \leftarrow E(1^\lambda, (x^B, y^A)) \end{array} : b' = \mathsf{Eval}(B, (x^B, y^B))\right] \geq 1/2 + \alpha(\lambda) - \nu_1(\lambda).$$

Proof of Claim 13. Follows by the security of Alice's HSS execution. Namely, consider a distinguishing adversary D for the $(2,2)$-AND-HSS, who performs the following:

1: Sample a random bit $b \leftarrow \{0,1\}$, and HSS share b as $(y^A, y^B) \leftarrow \mathsf{Share}(1^\lambda, B, b)$.
2: Receive a challenge secret share x^B, generated either as $(x^A, x^B) \leftarrow \mathsf{Share}(1^\lambda, A, 0)$ or $(x^A, x^B) \leftarrow \mathsf{Share}(1^\lambda, A, 1)$.
3: Execute E on "transcript" x^B and y^A: Let $b' \leftarrow E(1^\lambda, (x^B, y^A))$.
4: Output 0 if and only if $b' = b$.

By construction, the distinguishing advantage of D is exactly the difference in the prediction advantage of E from the real protocol and the protocol in which Alice shares 1 instead of 0. Thus, this difference must be bounded by some negligible function ν_1. ◀

▶ **Claim 14.** *The prediction advantage $\alpha(\lambda)$ of E must be negligible in λ.*

Proof of Claim 14. Follows by the security of Bob's HSS execution. Namely, consider a distinguishing adversary D for the $(2,2)$-AND-HSS, who performs the following:

1: Generate HSS shares of 1, as $(x^A, x^B) \leftarrow \mathsf{Share}(1^\lambda, A, 1)$.
2: Receive challenge secret share y^A, generated as $(y^A, y^B) \leftarrow \mathsf{Share}(1^\lambda, B, b)$ for random challenge bit $b \leftarrow \{0,1\}$.
3: Execute E on "transcript" x^B and y^A: Let $b' \leftarrow E(1^\lambda, (x^B, y^A))$.
4: Output b' as a guess for b.

By construction, the distinguishing advantage of D is precisely $\alpha(\lambda) - \nu_1(\lambda)$. Thus (since ν_1 is negligible), it must be that α is negligible, as desired. ◀

This concludes the proof of Proposition 12. ◀

As a direct corollary of this result, any form of HSS which implies additive $(2,2)$-HSS for AND automatically implies NIKE as well. This includes HSS for any functionality F with an embedded AND in its truth table.

As an example, consider a form of *split* distributed point function [37], where the nonzero input value $\alpha \in \{0,1\}^\ell$ of the secret point function f_α is held split as additive shares across two clients. This corresponds to additive $(2,2)$-HSS for the function $F(x;\alpha_1,\alpha_2) = [x == (\alpha_1 \oplus \alpha_2)]$ (i.e., evaluates to 1 if and only if $x = \alpha_1 \oplus \alpha_2$). Such a notion would have applications for secure computation protocols involving large public databases, where the index α of the desired data item is not known to either party, but rather determined as the result of an intermediate computation. Unfortunately, we show that such a tool (even for inputs of length 2 bits) implies NIKE, and thus is unlikely to exist from lightweight primitives.

▶ **Corollary 15.** *The existence of "split" DPF, i.e. additive $(2,2)$-HSS for the function $F(x;\alpha_1,\alpha_2) = [x == (\alpha_1 \oplus \alpha_2)]$, implies the existence of NIKE.*

Proof. Consider the special case of 2-bit values $\alpha_0, \alpha_1 \in \{0,1\}^2$. We show evaluation of F enables evaluation of AND of clients' input bits, and thus additive $(2,2)$-HSS for AND. Indeed, for any $b_1, b_2 \in \{0,1\}$, observe that $F((0,0);(1,b_1),(b_2,1)) = [(0,0) == ((1,b_1) \oplus (b_2,1))] = [(0,0) == (b_1 \oplus 1, b_2 \oplus 1)] = b_1 \wedge b_2$. ◀

5 Applications

In this section we present two types of applications of HSS. In Section 5.1 we present an application to 2-round secure multiparty computation, and in Section 5.2 we present an application to worst-case to average-case reductions.

5.1 From (3,2)-HSS to 2-Round MPC

Let us define the following function over $\mathbb{Z}_2$: $\mathsf{3Mult}(x_1,\ x_2,\ x_3) = x_1x_2x_3$. In this section, we show that $(3,2)$-HSS for $\mathsf{3Mult}$ implies 2-round MPC for arbitrary functions in the client-server model. Recall that (n,m)-HSS refers to HSS with n clients, m servers, tolerating up to $m-1$ corrupted servers. An n-client m-server MPC protocol for computing an n-ary functionality F, is a standard MPC protocol with $n+m+1$ parties, including n (input) clients each holding an input x_i, m servers, and a single output client who receives the output $F(x_1,\cdots,x_n)$. A 2-round n-client m-server MPC protocol has the special communication pattern that in the first round each client sends a message (a.k.a. input share) to each server, and in the second round each server sends a message (a.k.a. output share) to the output client, who then recovers the output. Such a protocol is t-secure if secure against any passive, semi-honest, adversary corrupting any set of parties including at most t servers, according to the standard definition of semi-honest security of MPC. Below we consider the default case of $t = m-1$, and denote such MPC as (n,m)-MPC. Due to the lack of space, we refer the reader to [18, 38] for standard definitions of MPC protocols, and to the full version for more details on client-server MPC.

▶ **Theorem 16.** *Assume the existence of PRGs in NC^1. For any n, m, and any polynomial-time computable function $F : (\{0,1\}^*)^n \to \{0,1\}$, there is a construction of an (n,m)-MPC protocol that securely computes F, from an additive $(3,2)$-HSS for $\mathsf{3Mult}$.*

Combining this with the additive δ-HSS construction of [14] from DDH would result in (n, m)-MPC from DDH with (at best) only $1/\mathsf{poly}(\lambda)$ correctness. Fortunately, we can do better. Indeed, as an intermediate step in the proof of Theorem 16 (Lemmas 21 and 22 below), we prove that $(3, 3)$-MPC for 3Mult also suffices to imply (n, m)-MPC for general functions. A construction of $(3, 3)$-MPC for general functions (in the PKI model) was shown to follow from DDH in [16] (in fact, they obtain (n, c)-MPC for any constant number of servers c). Combining this with Lemmas 21 and 22, and the fact that PRGs in NC^1 also follow from DDH, we obtain the following result. This improves directly over the 2-round MPC result of [16], by supporting an arbitrary polynomial number of servers instead of constant. (See Introduction for comparison with other recent 2-round MPC results.)

▶ **Corollary 17** (2-round MPC from DDH)**.** *For any n, m, and any polynomial-time computable function $F : (\{0,1\}^*)^n \to \{0,1\}$, there is a construction of an (n, m)-MPC protocol that securely computes F in the PKI model, assuming DDH.*

We prove Theorem 16 by combining the following steps; see full version for details.

Step 1: (3,2)-HSS for 3Mult-Plus. Starting from an additive $(3, 2)$-HSS scheme Π_{3Mult} for the function 3Mult, thanks to the property of additive reconstruction, we can directly modify it to obtain an additive $(3, 2)$-HSS for the function 3Mult-Plus (again over $\mathbb{Z}_2$) defined as

$$\mathsf{3Mult\text{-}Plus}((x_1, z_1),\ (x_2, z_2),\ (x_3, z_3)) \quad = \quad x_1x_2x_3 + z_1 + z_2 + z_3\ .$$

▶ **Lemma 18.** *There is a construction of additive $(3, 2)$-HSS for the function* 3Mult-Plus *from any additive $(3, 2)$-HSS for the function* 3Mult*.*

Step 2: (3,3)-MPC for 3Mult-Plus. From an additive $(3, 2)$-HSS scheme for 3Mult-Plus, we can use the *server-emulation technique* from [16] to construct a 3-client *3-server* MPC protocol for 3Mult-Plus. In fact, the technique in [16] is way more general, it shows that from any given n-client m-server HSS for 3Mult-Plus, one can construct a n-client m^2-server MPC protocol for any n-ary function F, assuming the existence of low-depth PRGs.

▶ **Lemma 19** (Server-Emulation in [16])**.** *Assume existence of PRGs in NC^1. For any n, m and polynomial-time function $F : (\{0,1\}^*)^n \to \{0,1\}$, there is a construction of an (n, m^2)-MPC protocol Π that securely computes F, from an additive (n, m)-HSS for* 3Mult-Plus*.*

Their general lemma implies the following corollary we need, using the fact that one can reduce the number of servers by having a single server simulating multiple ones.

▶ **Corollary 20.** *Assume the existence of PRGs in NC^1. There is a construction of a $(3, 3)$-MPC protocol that securely computes* 3Mult-Plus*, from an additive $(3, 2)$-HSS for* 3Mult-Plus*.*

Step 3: (3,m)-MPC for 3Mult-Plus – Increase the number of servers. Next, from a $(3, 3)$-MPC protocol for computing 3Mult-Plus, we show how to construct $(3, m)$-MPC protocol for computing the same function 3Mult-Plus, with an arbitrary number m of servers.

▶ **Lemma 21.** *For any m, there is a construction of $(3, m)$-MPC protocol that securely computes* 3Mult-Plus*, from a $(3, 3)$-MPC protocol that securely computes* 3Mult-Plus*.*

Proof Overview. Let Π^3 be a $(3, 3)$-MPC protocol for 3Mult-Plus; consider m servers, and three clients C_1, C_2, and C_3. Recall that each client C_d has input (x_d, z_d). If we naively let the three clients execute Π^3 with some subset of 3 servers, in the case all three servers are

corrupted, the security of Π^3 no longer holds, and the inputs of all clients are potentially revealed. Thus, the challenge is ensuring that when all but one server is corrupted, the inputs of honest clients remain hidden. To achieve this, each client secret-shares its input bit $x_d = \sum_j s_j^d$; as long as server S_j is uncorrupted, the j'th share s_j^d for each honest client's input x_d remains hidden, and hence so are the inputs x^d. (As we will see shortly, the additive part of the inputs z_d can be hidden easily.) Towards this, note that multiplying x_1, x_2, x_3 boils down to computing the sum of all possible degree 3 monomials over the shares $x_1x_2x_3 = \sum_{ijk} s_i^1 s_j^2 s_k^3$. Our idea is using the protocol Π^3 to compute the each monomial $s_i^1 s_j^2 s_k^3$ hidden with some random blinding bits, and in parallel, use a protocol Π_{Add} for addition to cancel out these random blinding bits, as well as add z_1, z_2, z_3. More specifically,

- for every i, j, k, C_1, C_2, C_3 together with three appropriate servers described below run Π^3 to enable the output client to obtain $M_{ijk} = s_i^1 s_j^2 s_k^3 + t_{ijk}^1 + t_{ijk}^2 + t_{ijk}^3$, where t_{ijk}^d is a random blinding bit sampled by client C_d;
- in parallel, C_1, C_2, C_3 together with all m servers run a $(3, m)$-MPC protocol Π_{Add} to enable the output client to obtain the sum $T = T^1 + T^2 + T^3$, where $T^d = z_d - \sum_{i,j,k} t_{ijk}^d$;
- finally, the output client adds all M_{ijk} with T, which gives the correct output, i.e., $x_1x_2x_3 + z_1 + z_2 + z_3$.

The only question left is what are the three servers involved for computing M_{ijk}; they naturally should be servers S_i, S_j, S_k, since for an honest client, say C_1, if server S_i is uncorrupted, the share s_i^1 remains hidden in all computations of M_{ijk} involving this share. This allows us to argue security. One technicality is that some monomials may have form $s_i^1 s_i^2 s_j^3$ or $s_i^1 s_i^2 s_i^3$ and only correspond to two servers S_i, S_j or one S_i. In the former case, we will use the $(3, 2)$-MPC protocol Π^2, and in the latter case, we directly implement a trivial protocol with one server. ◀

Step 4: (n,m)-MPC for F – Increase the number of clients and handle general function. Finally, we show how to construct MPC protocols for computing any n-ary function F, from MPC protocols for computing 3Mult-Plus, using the same number m of servers.

▶ **Lemma 22.** *Assume the existence of PRGs in* NC^1*. For any n, m, and any polynomial-time computable function $F : (\{0,1\}^*)^n \to \{0,1\}$, there is a construction of (n, m)-MPC protocol that securely computes F, from a $(3, m)$-MPC protocol that securely computes* 3Mult-Plus*.*

Proof Overview. Staring from a $(3, m)$-MPC protocol $\Pi_{\mathsf{3Mult\text{-}Plus}}$ for 3Mult-Plus, our goal is constructing a (n, m)-MPC protocol Π_F for an arbitrary F with an arbitrary number of clients. To do so, we reduce the task of computing F to the task of computing a *degree-3* randomized encoding $\mathsf{RE}_F(x_1, \cdots, x_n;\ r)$ of F. Here, having a degree of 3 means that RE_F can be represented as a degree 3 polynomial in its input *and* random bits. Such a randomized encoding scheme is constructed in [43, 1], assuming the existence of a low-depth PRG. The first question is where does the random tape r come from. Clearly, r can not be determined by any subset of clients. Therefore, the natural choice is having $r = r_1 + \cdots + r_n$ contributed by all clients. When the randomized encoding has degree 3, its computation can be expanded into a sum of degree three monomials, that is, $\mathsf{RE}_F(x_1, \cdots, x_n\ ;\ r = r_1 + \cdots + r_n) = \sum a_{ijk}^{\ell}\ v_i v_j v_k$, where each variable v_i is either a bit in some input x_l or a bit in some random tape r_l. This decomposes the computation of F into many 3-way multiplications, which can be done securely using 3Mult-Plus. More specifically, in the protocol Π_F,

- for every monomial $a_{ijk}^{\ell} v_i v_j v_k$, the three clients $C_{l_i}, C_{l_j}, C_{l_k}$ holding the variables v_i, v_j, v_k run $\Pi_{\mathsf{3Mult\text{-}Plus}}$ with all m servers to enable the output client to obtain $M_{ijk} = a_{ijk}^{\ell} v_i v_j v_k +$

$t^{\ell,1}_{ijk} + t^{\ell,2}_{ijk} + t^{\ell,3}_{ijk}$, where the three t variables are random blinding bits sampled by the three clients respectively;

- in parallel, all clients and servers run a (n, m)-MPC protocol Π_{Add} for addition to enable the output client to obtain the sum of all t blinding elements;
- the output client adds all M_{ijk} terms, subtracts the sum of blinding elements to obtain the randomized encoding of F, and decodes the randomized encoding. ◀

5.2 Worst-Case to Average-Case Reductions

In this section we describe a simple application of HSS to worst-case to average-case reductions. We then discuss applications of these reductions to fine-grained average-case hardness and verifiable computation. These applications of HSS can be seen as more efficient or more general conditional variants of previous applications of locally random reductions that rely on arithmetization or error-correcting codes [51, 12, 4, 55, 37, 6]. In contrast to the above reductions, the HSS-based reductions can reduce any polynomial-time computable function to another polynomial-time computable function with closely related complexity.

Worst-case to average-case reductions based on fully homomorphic encryption (FHE) were previously used by Chung et al. [24] in the context of delegating computations. Compared to the FHE-based reductions, the use of HSS has the advantages of diversifying assumptions, making only a constant number of queries to a *Boolean* function (as small as 2), and minimizing the complexity of recovering the output from the answers to the queries.

To make the discussion concrete, we focus here on the application of (computationally secure, additive[2] for the universal function $F(C; x) = C(x)$. Such HSS schemes can be based on variants of the LWE assumption (as described in the full version). Weaker versions of the following results that apply to branching programs can be based on the DDH assumption or the circular security of Paillier encryption using the HSS schemes from [14, 30].

A high level overview. The idea of using HSS for worst-case to average-case reductions is similar to previous applications of locally random reductions for this purpose, except that we apply a "hybrid HSS" technique [14] to improve the efficiency of the reduction. Concretely, the reduction proceeds as follows. Suppose for simplicity that the HSS sharing algorithm $\mathsf{Share}(1^\lambda, x)$ outputs a pair of shares (x^1, x^2) such that each share is individually pseudorandom. Moreover, suppose that the evaluation function $\mathsf{Eval}(j, C, x^j)$ does not depend on j. The evaluation of a circuit $C : \{0,1\}^n \to \{0,1\}$ on an arbitrary input $x \in \{0,1\}^n$ can then be reduced to the evaluation of an extended circuit $\hat{C}$, defined by $\hat{C}(\hat{x}) = \mathsf{Eval}(C, \hat{x})$, on the two inputs x^1, x^2. Indeed, $C(x) = \hat{C}(x^1) \oplus \hat{C}(x^2)$. Now, suppose that $\hat{C}^*$ is a polynomial-size circuit that agrees with $\hat{C}$ on all but an ϵ fraction of the inputs. Then, by the pseudo-randomness of x^1, x^2, the probability that $\hat{C}^*$ agrees with $\hat{C}$ on both inputs, and hence the reduction outputs the correct value $C(x)$, is at least $1 - 2\epsilon - \mathsf{negl}(n)$. Finally, to make the reduction run in near-linear time, we convert the given HSS into a hybrid HSS scheme in which the sharing Share' can be implemented in near-linear time. The algorithm Share' uses Share to share a short seed r for a pseudorandom generator G, and includes the masked input $G(r) \oplus x$ as part of both shares. Given a circuit C and $G(r) \oplus x$, one can efficiently compute a circuit C' such that $C'(r) = C(x)$. The algorithm Eval' of the hybrid scheme applies Eval to homomorphically evaluate C' on r.

[2] The requirement of being additive can be relaxed here to small decoding complexity.

The following theorem formalizes and generalizes the above. Here, by a "near-linear time" algorithm we refer to an algorithm whose running time is $O(n^{1+\epsilon})$ for an arbitrary $\epsilon > 0$. The proof of the theorem is deferred to the full version.

▶ **Theorem 23** (Worst-case to average-case reductions from HSS). *Suppose there is a* $(1,2)$*-HSS scheme* (Share, Eval, Dec) *for circuits. Then, there is a near-linear time probabilistic oracle algorithm* $Q^{[\cdot]} : \{0,1\}^* \to \{0,1\}^*$*, polynomial-time algorithm* $A : \{0,1\}^* \times \{0,1\}^* \to \{0,1\}$*, and a PPT sampling algorithm* $D(1^n)$ *with the following properties:*

- *Q makes two queries to a Boolean oracle (where the queries are computed in near-linear time and the answers are 1-bit long) and outputs the exclusive-or of the two answer bits.*
- *For any $x \in \{0,1\}^n$ and circuit $C : \{0,1\}^n \to \{0,1\}$, we have $\Pr[Q^{A(C,\cdot)}(x) = C(x)] = 1$.*
- *For any polynomial $p(\cdot)$ there is a negligible $\mu(\cdot)$ such that the following holds. For any $x \in \{0,1\}^n$ and circuits $C : \{0,1\}^n \to \{0,1\}$, $A^*_C : \{0,1\}^n \to \{0,1\}$ of size $\leq p(n)$ such that $\Pr_{\hat{x} \leftarrow D(1^n)}[A^*_C(\hat{x}) = A(C,\hat{x})] \geq 1-\epsilon$, we have $\Pr[Q^{A^*_C(\cdot)}(x) = C(x)] \geq 1 - 2\epsilon - \mu(n)$.*

Moreover, if Share *produces pseudorandom shares then the distribution $D(1^n)$ can be replaced by the uniform distribution.*

▶ **Remark 24** (Instantiating Theorem 23). *The strong flavor of HSS required by Theorem 23 can be instantiated under a variant of the LWE assumption that further assumes circular security [35, 27]. Due to the negligible decoding error of the HSS, we get a slightly weaker version of the conclusion where $\Pr[Q^{A(C,\cdot)}(x) = C(x)] \geq 1 - \mathsf{negl}(n)$. On the other hand, since the implementation of* Eval *has a small asymptotic overhead, we get the stronger guarantee that the oracle $A(C,\cdot)$ has roughly the same circuit size as C (rather than being polynomially bigger). One can relax the assumption to a more standard variant of LWE by using* depth-dependent *HSS for circuits, where the length of the input shares grows polynomially with the depth of the circuit C being evaluated by* Eval*. In this case, using an LWE-based NC^1 implementation of the PRG G, the conclusion of Theorem 23 still holds when restricted to NC-circuits C. More generally, the complexity of Q should in this case be allowed to grow with the depth of C.*

We now informally discuss two types of applications of Theorem 23, which follow previous applications of such worst-case to average-case reductions from the literature.

Fine-grained average-case hardness. Theorem 23 implies, assuming HSS for circuits, that the following holds for any constants $c' > c$. For every polynomial-time computable function f there is a polynomial-time computable "extension" $\hat{f}$, such that if $\hat{f}$ has a time-$O(n^c)$ algorithm that computes it correctly on, say, 90% on the inputs, then f has a time-$O(n^{c'})$ probabilistic algorithm that computes it correctly (with overwhelming probability) on every input. This implies that if f is hard in the worst case for time $O(n^{c'})$ then $\hat{f}$ is hard in the average case for time $O(n^c)$. The same connection holds also in a non-uniform setting.

A similar result under the incomparable assumption that FHE exists is given in [24]. These results are incomparable to recent results on fine-grained average case hardness [6, 41] that obtain tighter and unconditional connections of this kind, but only for specific functions f.

Verifiable computation. The goal of program checking [12] is to reliably compute a given function f using an untrusted program or piece of hardware that purportedly computes f. We consider a variant of the problem in which a program M for computing $f : \{0,1\}^n \to \{0,1\}$ can access a purported implementation of a related function $\hat{f}$. The program M can make oracle calls to $\hat{f}$ and perform additional computations, as long as the complexity of these

additional computations is significantly smaller than that of computing f from scratch. The requirements are that if $\hat{f}$ is implemented correctly, then $M^{\hat{f}}(x) = f(x)$ for all x. On the other hand, even if $\hat{f}$ is replaced by an incorrect implementation $\hat{f}^*$, the output of $M^{\hat{f}^*}(x)$ on every input x is either $f(x)$ or $\perp$ except with small failure probability ϵ. This is very similar to the traditional goal of verifiable computation, except that a malicious "prover" $\hat{f}^*$ is required to be stateless. In this setting, one can make a direct use of probabilistically checkable proofs (PCPs) for proving the correctness of $f(x)$ without any additional cryptographic machinery.

Using the HSS-based worst-case to average-case reduction from Theorem 23, we get checkers M with the following feature: after an input-independent polynomial-time preprocessing, any computation $f(x)$ can be verified with an arbitrarily small inverse polynomial error by receiving just a constant number of bits from $\hat{f}^*$. (See full version for details.) We do not know of any other approach for verifiable computation that yields such a result.

6 Conclusions and Open Problems

In this work we initiate a systematic study of homomorphic secret sharing (HSS) by providing a taxonomy of HSS variants and establishing some negative results and relations with other primitives. We also present applications of HSS in cryptography and complexity theory.

There is much left to understand about the feasibility and efficiency of HSS in different settings. In the information-theoretic setting, we have no strong negative results for *single-input,* (weakly) *compact* HSS. This should be contrasted with *multi-input* compact HSS, for which negative results are obtained in this work, and with single-input *additive* HSS, where information-theoretic impossibility results are also known [22]. The difficulty of making progress on this question can be partially explained by its relation with information-theoretic private information retrieval and locally decodable codes [46, 8], for which proving good lower bounds is still an outstanding challenge. However, this barrier only seems to apply to special instances of the general problem. In the computational setting, the main open problems are to obtain HSS schemes for circuits under new assumptions and, more broadly, extend the capabilities of HSS schemes that do not rely on FHE.

Acknowledgements. We thank the anonymous ITCS reviewers for helpful comments.

References

1 Benny Applebaum, Yuval Ishai, and Eyal Kushilevitz. Cryptography in NC^0. In *FOCS 2004*, pages 166–175, 2004.

2 Benny Applebaum, Yuval Ishai, and Eyal Kushilevitz. Computationally private randomizing polynomials and their applications. In *CCC*, pages 260–274, 2005.

3 Gilad Asharov, Abhishek Jain, Adriana López-Alt, Eran Tromer, Vinod Vaikuntanathan, and Daniel Wichs. Multiparty computation with low communication, computation and interaction via threshold FHE. In *EUROCRYPT*, pages 483–501, 2012.

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

5 László Babai, Anna Gál, Peter G. Kimmel, and Satyanarayana V. Lokam. Communication complexity of simultaneous messages. *SIAM J. Comput.*, 33(1):137–166, 2003.

6 Marshall Ball, Alon Rosen, Manuel Sabin, and Prashant Nalini Vasudevan. Average-case fine-grained hardness. In *STOC 2017*, pages 483–496, 2017.

7 Omer Barkol, Yuval Ishai, and Enav Weinreb. On d-multiplicative secret sharing. *J. Cryptology*, 23(4):580–593, 2010.

8 Amos Beimel, Yuval Ishai, Eyal Kushilevitz, and Ilan Orlov. Share conversion and private information retrieval. In *CCC 2012*, pages 258–268, 2012.

9 Michael Ben-Or, Shafi Goldwasser, and Avi Wigderson. Completeness theorems for non-cryptographic fault-tolerant distributed computation. In *STOC*, pages 1–10, 1988.

10 Josh Cohen Benaloh. Secret sharing homomorphisms: Keeping shares of A secret sharing. In *CRYPTO 1986*, pages 251–260, 1986.

11 Fabrice Benhamouda and Huijia Lin. k-round MPC from k-round OT via garbled interactive circuits. Manuscript, 2017.

12 Manuel Blum and Sampath Kannan. Designing programs that check their work. In *STOC 1989*, pages 86–97, 1989.

13 E. Boyle, N. Gilboa, and Y. Ishai. Function secret sharing. In *EUROCRYPT*, pages 337–367, 2015.

14 Elette Boyle, Niv Gilboa, and Yuval Ishai. Breaking the circuit size barrier for secure computation under DDH. In *CRYPTO*, pages 509–539, 2016. Full version: IACR Cryptology ePrint Archive 2016: 585 (2016).

15 Elette Boyle, Niv Gilboa, and Yuval Ishai. Function secret sharing: Improvements and extensions. In *ACM CCS 2016*, pages 1292–1303, 2016.

16 Elette Boyle, Niv Gilboa, and Yuval Ishai. Group-based secure computation: Optimizing rounds, communication, and computation. In *EUROCRYPT*, pages 163–193, 2017.

17 Zvika Brakerski and Vinod Vaikuntanathan. Efficient fully homomorphic encryption from (standard) LWE. *SIAM J. Comput.*, 43(2):831–871, 2014.

18 Ran Canetti. Security and composition of multiparty cryptographic protocols. *Journal of Cryptology*, pages 143–202, 2000.

19 Ignacio Cascudo Pueyo, Hao Chen, Ronald Cramer, and Chaoping Xing. Asymptotically good ideal linear secret sharing with strong multiplication over *Any* fixed finite field. In *CRYPTO*, pages 466–486, 2009.

20 David Chaum, Claude Crépeau, and Ivan Damgård. Multiparty unconditionally secure protocols. In *STOC*, pages 11–19, 1988.

21 Hao Chen and Ronald Cramer. Algebraic geometric secret sharing schemes and secure multi-party computations over small fields. In *CRYPTO*, pages 521–536, 2006.

22 B. Chor, O. Goldreich, E. Kushilevitz, and M. Sudan. Private information retrieval. *J. ACM*, 45(6):965–981, 1998.

23 Benny Chor and Niv Gilboa. Computationally private information retrieval (extended abstract). In *STOC 1997*, pages 304–313, 1997.

24 Kai-Min Chung, Yael Tauman Kalai, and Salil P. Vadhan. Improved delegation of computation using fully homomorphic encryption. In *CRYPTO 2010*, pages 483–501, 2010.

25 Ronald Cramer, Ivan Damgård, and Ueli M. Maurer. General secure multi-party computation from any linear secret-sharing scheme. In *EUROCRYPT*, pages 316–334, 2000.

26 Whitfield Diffie and Martin Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, 22(6):644–654, 1976.

27 Yevgeniy Dodis, Shai Halevi, Ron D. Rothblum, and Daniel Wichs. Spooky encryption and its applications. In *CRYPTO*, pages 93–122, 2016.

28 Z. Dvir and S. Gopi. 2-server PIR with sub-polynomial communication. In *STOC*, pages 577–584, 2015.

29 Klim Efremenko. 3-query locally decodable codes of subexponential length. In *STOC*, pages 39–44, 2009.

30 Nelly Fazio, Rosario Gennaro, Tahereh Jafarikhah, and William E. Skeith III. Homomorphic secret sharing from paillier encryption. In *ProvSec*, pages 381–399, 2017.

31 Matthew K. Franklin and Moti Yung. Communication complexity of secure computation (extended abstract). In *STOC*, pages 699–710, 1992.

32 Sanjam Garg, Craig Gentry, Shai Halevi, and Mariana Raykova. Two-round secure MPC from indistinguishability obfuscation. In *TCC*, pages 74–94, 2014.

33 Sanjam Garg and Akshayaram Srinivasan. Garbled protocols and two round MPC from bilinear maps. In *FOCS 2017*, 2017.

34 Sanjam Garg and Akshayaram Srinivasan. Two-round secure multiparty computation from minimal assumptions. Manuscript, 2017.

35 Craig Gentry. Fully homomorphic encryption using ideal lattices. 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 169–178. ACM, 2009. doi:10.1145/1536414.1536440.

36 Craig Gentry, Amit Sahai, and Brent Waters. Homomorphic encryption from learning with errors: Conceptually-simpler, asymptotically-faster, attribute-based. In *CRYPTO (1)*, pages 75–92, 2013.

37 N. Gilboa and Y. Ishai. Distributed point functions and their applications. In *Proc. EUROCRYPT '14*, pages 640–658, 2014.

38 Oded Goldreich. *Foundations of Cryptography — Basic Applications*. Cambridge University Press, 2004.

39 Oded Goldreich, Howard J. Karloff, Leonard J. Schulman, and Luca Trevisan. Lower bounds for linear locally decodable codes and private information retrieval. *Computational Complexity*, 15(3):263–296, 2006.

40 Oded Goldreich, Silvio Micali, and Avi Wigderson. How to play any mental game or a completeness theorem for protocols with honest majority. In *STOC*, pages 218–229, 1987.

41 Oded Goldreich and Guy N. Rothblum. Worst-case to average-case reductions for subclasses of p. *Electronic Colloquium on Computational Complexity (ECCC)*, 17-130, 2017.

42 S. Dov Gordon, Feng-Hao Liu, and Elaine Shi. Constant-round MPC with fairness and guarantee of output delivery. In *CRYPTO, Part II*, pages 63–82, 2015.

43 Yuval Ishai and Eyal Kushilevitz. Perfect constant-round secure computation via perfect randomizing polynomials. In *ICALP*, pages 244–256, 2002.

44 Aayush Jain, Peter M. R. Rasmussen, and Amit Sahai. Threshold fully homomorphic encryption. *IACR Cryptology ePrint Archive*, 2017:257, 2017.

45 Bala Kalyanasundaram and Georg Schintger. The probabilistic communication complexity of set intersection. *SIAM J. Discret. Math.*, 5(4), 1992.

46 Jonathan Katz and Luca Trevisan. On the efficiency of local decoding procedures for error-correcting codes. In *STOC*, pages 80–86, 2000.

47 Iordanis Kerenidis and Ronald de Wolf. Exponential lower bound for 2-query locally decodable codes via a quantum argument. *J. Comput. Syst. Sci.*, 69(3):395–420, 2004.

48 Ilan Kremer, Noam Nisan, and Dana Ron. On randomized one-round communication complexity. *Computational Complexity*, 8(1):21–49, 1999.

49 E. Kushilevitz and R. Ostrovsky. Replication is NOT needed: SINGLE database, computationally-private information retrieval. In *Proc. FOCS '97*, pages 364–373, 1997.

50 Eyal Kushilevitz and Noam Nisan. *Communication complexity*. Cambridge University Press, 1997.

51 Richard J. Lipton. New directions in testing. In *DIMACS Workshop on Distributed Computing And Cryptography*, pages 191–202, 1989.

52 Pratyay Mukherjee and Daniel Wichs. Two round multiparty computation via multi-key FHE. In *EUROCRYPT*, pages 735–763, 2016.

53 Ronald L. Rivest, Len Adleman, and Michael L. Dertouzos. On data banks and privacy homomorphisms. *Foundations of secure computation*, 4(11):169–180, 1978.

54 Adi Shamir. How to share a secret. *Commun. ACM*, 22(11):612–613, 1979.

55 Madhu Sudan, Luca Trevisan, and Salil P. Vadhan. Pseudorandom generators without the XOR lemma. *J. Comput. Syst. Sci.*, 62(2):236–266, 2001.

56 Marten van Dijk, Craig Gentry, Shai Halevi, and Vinod Vaikuntanathan. Fully homomorphic encryption over the integers. 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 24–43. Springer, 2010. doi:10.1007/978-3-642-13190-5_2.

57 Andrew Chi-Chih Yao. How to generate and exchange secrets (extended abstract). In *FOCS*, pages 162–167, 1986.

58 S. Yekhanin. Towards 3-query locally decodable codes of subexponential length. In *Proc. STOC*, pages 266–274, 2007.

Convergence Results for Neural Networks via Electrodynamics*

Rina Panigrahy[1], Ali Rahimi[2], Sushant Sachdeva[†3], and Qiuyi Zhang[4]

1 **Google Inc., Mountain View, CA, USA**
rinap@google.com
2 **Google Inc., Mountain View, CA, USA**
arahimi@google.com
3 **University of Toronto, Toronto, Canada**
sachdeva@cs.toronto.edu
4 **University of California Berkeley, Berkeley, CA, USA**
10zhangqiuyi@berkeley.edu

Abstract

We study whether a depth two neural network can learn another depth two network using gradient descent. Assuming a linear output node, we show that the question of whether gradient descent converges to the target function is equivalent to the following question in electrodynamics: Given k fixed protons in $\mathbb{R}^d$, and k electrons, each moving due to the attractive force from the protons and repulsive force from the remaining electrons, whether at equilibrium all the electrons will be matched up with the protons, up to a permutation. Under the standard electrical force, this follows from the classic Earnshaw's theorem. In our setting, the force is determined by the activation function and the input distribution. Building on this equivalence, we prove the existence of an activation function such that gradient descent learns at least one of the hidden nodes in the target network. Iterating, we show that gradient descent can be used to learn the entire network one node at a time.

1998 ACM Subject Classification I.2.6 Learning

Keywords and phrases Deep Learning, Learning Theory, Non-convex Optimization

Digital Object Identifier 10.4230/LIPIcs.ITCS.2018.22

1 Introduction

Deep learning has resulted in major strides in machine learning applications including speech recognition, image classification, and ad-matching. The simple idea of using multiple layers of nodes with a non-linear activation function at each node allows one to express any function. To learn a certain target function we just use (stochastic) gradient descent to minimize the loss; this approach has resulted in significantly lower error rates for several real world functions, such as those in the above applications. Naturally the question remains: how close are we to the optimal values of the network weight parameters? Are we stuck in some bad local minima? While there are several recent works [8, 11, 17] that have tried to study the presence of local minima, the picture is far from clear.

* The full version of this paper is available at [25], https://arxiv.org/abs/1702.00458
† Part of this work was done when this author was a research scientist at Google Inc., Mountain View, CA, USA

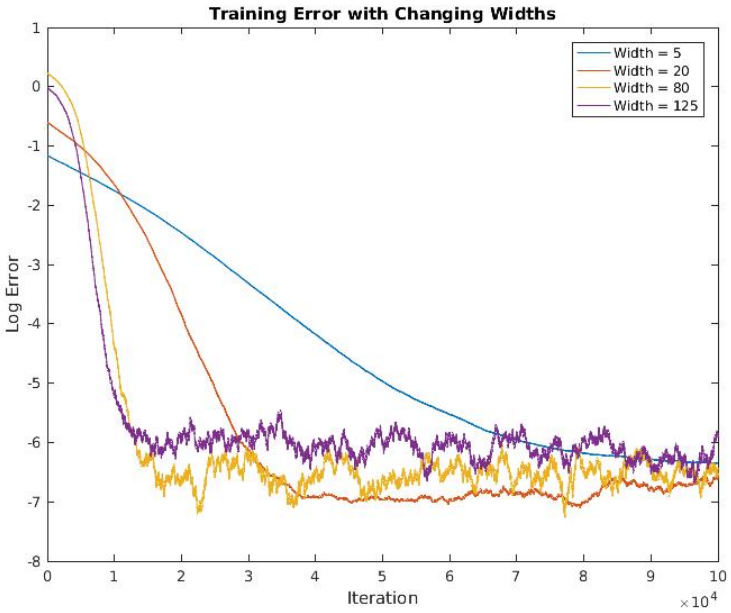


Figure 1 Test Error of Depth 2 Networks of Varying Width.

There has been some work on studying how well can neural networks learn some synthetic function classes (e.g. polynomials [1], decision trees). In this work we study how well can neural networks learn neural networks with gradient descent? Our focus here, via the framework of proper learning, is to understand if a neural network can learn a function from the same class (and hence achieve vanishing error).

Specifically, if the target function is a neural network with randomly initialized weights, and we attempt to learn it using a network with the same architecture, then, will gradient descent converge to the target function?

Experimental simulations (see Figure 1 and Section 5 for further details) show that for depth 2 networks of different widths, with random network weights, stochastic gradient descent of a hypothesis network with the same architecture converges to a squared ℓ_2 error that is a small percentage of a random network, indicating that SGD can learn these shallow networks with random weights. Because our activations are sigmoidal from -1 to 1, the training error starts from a value of about 1 (random guessing) and diminishes quickly to under 0.002. This seems to hold even when the width, the number of hidden nodes, is substantially increased (even up to 125 nodes), but depth is held constant at 2.

In this paper, we attempt to understand this phenomenon theoretically. We prove that, under some assumptions, depth-2 neural networks can learn functions from the same class with vanishingly small error using gradient descent.

1.1 Results and Contributions

We theoretically investigate the question of convergence for networks of depth two. Our main conceptual contribution is that for depth 2 networks where the top node is a sum node, the question of whether gradient descent converges to the desired target function is equivalent to the following question in electrodynamics: Given k fixed protons in $\mathbb{R}^d$, and k moving electrons, with all the electrons moving under the influence of the electrical force of

Table 1 Activation, Potentials, and Convergence Results Summary

Name of Activation	Potential ($\Phi(\theta, w)$)	Convergence?
Almost λ-harmonic	Complicated (see Lem 13)	Yes, Thm 17
Sign	$1 - \frac{2}{\pi}\cos^{-1}(\theta^T w)$	Yes for d = 2, Lem 29
Polynomial	$(\theta^T w)^m$	Yes, for orthonormal w_i. Lem 30

attraction from the protons and repulsion from the remaining electrons, at equilibrium, are all the electrons matched up with all the fixed protons, up to a permutation?

In the above, k is the number of hidden units, d is the number of inputs, the positions of each fixed charge is the input weight vector of a hidden unit in the target network, and the initial positions of the moving charges are the initial values of the weight vectors for the hidden units in the learning network. The motion of the charges essentially tracks the change in the network during gradient descent. The force between a pair of charges is not given by the standard electrical force of $1/r^2$ (where r is the distance between the charges), but by a function determined by the activation and the input distribution. Thus the question of convergence in these simplified depth two networks can be resolved by studying the equivalent electrodynamics question with the corresponding force function.

▶ **Theorem 1** (informal statement of Theorem 5). *Applying gradient descent for learning the output of a depth two network with k hidden units with activation σ, and a linear output node, under squared loss, using a network of the same architecture, is equivalent to the motion of k charges in the presence of k fixed charges where the force between each pair of charges is given by a potential function that depends on σ and the input distribution.*

Based on this correspondence we prove the existence of an activation function such that the corresponding gradient descent dynamics under standard Gaussian inputs result in learning at least one of the hidden nodes in the target network. We then show that this allows us to learn the complete target network one node at a time. For more realistic activation functions, we only obtain partial results. We assume the sample complexity is close to its infinite limit.

▶ **Theorem 2** (informal statement of Theorem 12). *There is an activation function such that running gradient descent for minimizing the squared loss along with ℓ_2 regularization for standard Gaussian inputs, at convergence, we learn at least one of the hidden weights of the target neural network.*

We prove that the above result can be iterated to learn the entire network node-by-node using gradient descent (Theorem 17). Our algorithm learns a network with the same architecture and number of hidden nodes as the target network, in contrast with several existing improper learning results.

In the appendix, we show some weak results for more practical activations. For the sign activation, we show that for the loss with respect to a single node, the only local minima are at the hidden target nodes with high probability if the target network has a randomly picked top layer. For the polynomial activation, we derive a similar result under the assumption that the hidden nodes are orthonormal.

1.2 Intuition and Techniques

Note that for the standard electric potential function given by $\Phi = 1/r$ where r is the distance between the charges, it is known from Earnshaw's theorem that an electrodynamic system with some fixed protons and some moving electrons is at equilibrium only when the moving electrons coincide with the fixed protons. Given our translation above between electrodynamic systems and depth 2 networks (Section 2), this would imply learnability of depth 2 networks under gradient descent under ℓ_2 loss, if the activation function corresponds to the electrostatic potential. However, there exists no activation function σ corresponding to this Φ.

The proof of Earnshaw's theorem is based on the fact that the electrostatic potential is harmonic, *i.e*, its Laplacian (trace of its Hessian) is identically zero. This ensures that at every critical point, there is direction of potential reduction (unless the hessian is identically zero). We generalize these ideas to potential functions that are eigenfunctions of the Laplacians, λ-harmonic potentials (Section 3). However, these potentials are unbounded. Subsequently, we construct a non-explicit activation function such that the corresponding potential is bounded and is almost λ-harmonic, *i.e.*, it is λ-harmonic outside a small sphere (Section 4). For this activation function, we show at a stable critical point, we must learn at least one of the hidden nodes. Gradient descent (possibly with some noise, as in the work of Ge *et al.* [12]) is believed to converge to stable critical points. However, for simplicity, we descend along directions of negative curvature to escape saddle points. Our activation lacks some regularity conditions required in [12]. We believe the results in [16] can be adapted to our setting to prove that perturbed gradient descent converges to stable critical points.

There is still a large gap between theory and practice. However, we believe our work can offer some theoretical explanations and guidelines for the design of better activation functions for gradient-based training algorithms. For example, better accuracy and training speed were reported when using the newly discovered exponential linear unit (ELU) activation function in [9, 21]. We hope for more theory-backed answers to these and many other questions in deep learning.

1.3 Related Work

If the activation functions are linear or if some independence assumptions are made, Kawaguchi shows that the only local minima are the global minima [17]. Under the spin-glass and other physical models, some have shown that the loss landscape admits well-behaving local minima that occur usually when the overall error is small [8, 11]. When only training error is considered, some have shown that a global minima can be achieved if the neural network contains sufficiently many hidden nodes [23]. Recently, Daniely has shown that SGD learns the conjugate kernel class [10]. Under simplifying assumptions, some results for learning ReLU's with gradient descent are given in [24, 7]. Our research is inspired by [1], where the authors show that for polynomial target functions, gradient descent on neural networks with one hidden layer converges to low error, given a large number of hidden nodes, and under complex perturbations, there are no robust local minima. Even more recently, similar results about the convergence of SGD for two-layer neural networks have been established for a polynomial activation function under a more complex loss function [13]. And in [19], they study the same problem as ours with the RELU activation and where lower layer of the network is close to identity and the upper layer has weights all one. This corresponds to the case where each electron is close to a distinct proton – under these assumptions they show that SGD learns the true network.

Under worst case assumptions, there has been hardness results for even simple networks. A neural network with one hidden unit and sigmoidal activation can admit exponentially many local minima [4]. Backprogration has been proven to fail in a simple network due to the abundance of bad local minima [6]. Training a 3-node neural network with one hidden layer is NP-complete [5]. But, these and many similar worst-case hardness results are based on worst case training data assumptions. However, by using a result in [18] that learning a neural network with threshold activation functions is equivalent to learning intersection of halfspaces, several authors showed that under certain cryptographic assumptions, depth-two neural networks are not efficiently learnable with smooth activation functions [20, 27, 26].

Due to the difficulty of analysis of the non convex gradient descent in deep learning, many have turned to improper learning and the study of non-gradient methods to train neural networks. Janzamin et. al use tensor decomposition methods to learn the shallow neural network weights, provided access to the score function of the training data distribution [15]. Eigenvector and tensor methods are also used to train shallow neural networks with quadratic activation functions in [20]. Combinatorial methods that exploit layerwise correlations in sparse networks have also been analyzed provably in [3]. Kernel methods, ridge regression, and even boosting were explored for regularized neural networks with smooth activation functions in [22, 27, 26]. Non-smooth activation functions, such as the ReLU, can be approximated by polynomials and are also amenable to kernel methods[14]. These methods however are very different from the simple popular SGD.

2 Deep Learning, Potentials, and Electron-Proton Dynamics

2.1 Preliminaries

We will work in the space $\mathcal{M} = \mathbb{R}^d$. We denote the gradient and Hessian as $\nabla_{\mathbb{R}^d} f$ and $\nabla^2_{\mathbb{R}^d} f$ respectively. The Laplacian is defined as $\Delta_{\mathbb{R}^d} f = \mathrm{Tr}(\nabla^2_{\mathbb{R}^d} f)$. If f is multivariate with variable x_i, then let f_{x_i} be a restriction of f onto the variable x_i with all other variables fixed. Let $\nabla_{x_i} f, \Delta_{x_i} f$ to be the gradient and Laplacian, respectively, of f_{x_i} with respect to x_i. Lastly, we say x is a critical point of f if ∇f does not exist or $\nabla f = 0$.

We focus on learning depth two networks with a linear activation on the output node. If the network takes inputs $x \in \mathbb{R}^d$ (say from some distribution $\mathcal{D}$), then the network output, denoted $f(x)$ is a sum over $k = \mathrm{poly}(d)$ hidden units with weight vectors $w_i \in \mathbb{R}^d$, activation $\sigma(x, w) : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$, and output weights $b_i \in \mathbb{R}$. Thus, we can write $f(x) = \sum_{i=1}^k b_i \sigma(x, w_i)$. We denote this concept class $\mathcal{C}_{\sigma,k}$. Our hypothesis concept class is also $\mathcal{C}_{\sigma,k}$.

Let $\boldsymbol{a} = (a_1, ..., a_k)$ and $\boldsymbol{\theta} = (\theta_1, ..., \theta_k)$; similarly for $\boldsymbol{b}, \boldsymbol{w}$ and our guess is $\hat{f}(x) = \sum_{i=1}^k a_i \sigma(x, \theta_i)$. We define Φ, the **potential function** *corresponding to the activation* σ, as

$$\Phi(\theta, w) = \mathop{\mathbb{E}}_{X \sim D}[\sigma(X, \theta)\sigma(X, w)].$$

We work directly with the true squared loss error $L(a, \theta) = \mathbb{E}_{x \sim \mathcal{D}}[(f - \hat{f})^2]$. To simplify L, we re-parametrize a by $-a$ and expand.

$$\begin{aligned} L(\boldsymbol{a}, \boldsymbol{\theta}) &= \mathop{\mathbb{E}}_{X \sim D}\left[\left(\sum_{i=1}^k a_i \sigma(X, \theta_i) + \sum_{i=1}^k b_i \sigma(X, w_i)\right)^2\right] \\ &= \sum_{i=1}^k \sum_{j=1}^k a_i a_j \Phi(\theta_i, \theta_j) + 2 a_i b_j \Phi(\theta_i, w_j) + b_i b_j \Phi(w_i, w_j), \end{aligned} \tag{1}$$

Given $\mathcal{D}$, the activation function σ, and the loss L, we attempt to show that we can use some variant of gradient descent to learn, with high probability, an ϵ-approximation of w_j for some (or all) j. Note that our loss is jointly convex, though it is quadratic in $\boldsymbol{a}$.

In this paper, we restrict our attention to translationally invariant activations and potentials. Specifically, we may write $\Phi = h(\theta - w)$ for some function $h(x)$. Furthermore, a translationally invariant function $\Phi(r)$ is *radial* if it is a function of $r = \|x - y\|$.

▶ Remark. Translationally symmetric potentials satisfy $\Phi(\theta, \theta)$ *is a positive constant.* We normalize $\Phi(\theta, \theta) = 1$ for the rest of the paper.

We assume that our input distribution $\mathcal{D} = \mathcal{N}(0, \mathbf{I_{d\times d}})$ is fixed as the standard Gaussian in $\mathbb{R}^d$. This assumption is not critical and a simpler distribution might lead to better bounds. However, for arbitrary distributions, there are hardness results for PAC-learning halfspaces [18].

We call a potential function **realizable** if it corresponds to some activation σ. The following theorem characterizes realizable translationally invariant potentials under standard Gaussian inputs. Proofs and a similar characterization for rotationally invariant potentials can be found in Appendix B.

▶ **Theorem 3.** *Let $\mathcal{M} = \mathbb{R}^d$ and Φ is square-integrable and $\mathfrak{F}(\Phi)$ is integrable. Then, Φ is realizable under standard Gaussian inputs if $\mathfrak{F}(\Phi)(\omega) \geq 0$ and the corresponding activation is $\sigma(x) = (2\pi)^{d/4} e^{x^T x/4} \mathfrak{F}^{-1}(\sqrt{\mathfrak{F}(\Phi)})(x)$, where $\mathfrak{F}$ is the generalized Fourier transform in $\mathbb{R}^d$.*

2.2 Electron-Proton Dynamics

By interpreting the pairwise potentials as electrostatic attraction potentials, we notice that our dynamics is similar to electron-proton type dynamics under potential Φ, where w_i are fixed point charges in $\mathbb{R}^d$ and θ_i are moving point charges in $\mathbb{R}^d$ that are trying to find w_i. The total force on each charge is the sum of the pairwise forces, determined by the gradient of Φ. We note that standard dynamics interprets the force between particles as an acceleration vector. In gradient descent, it is interpreted as a velocity vector.

▶ **Definition 4.** Given a potential Φ and particle locations $\theta_1, ..., \theta_k \in \mathbb{R}^d$ along with their respective charges $a_1, ..., a_k \in \mathbb{R}$. We define **Electron-Proton Dynamics** under Φ with some subset $S \subseteq [k]$ of fixed particles to be the solution $(\theta_1(t), ..., \theta_k(t))$ to the following system of differential equations: For each pair (θ_i, θ_j), there is a force from θ_j exerted on θ_i that is given by $\mathbf{F}_i(\theta_j) = a_i a_j \nabla_{\theta_i} \Phi(\theta_i, \theta_j)$ and

$$-\frac{d\theta_i}{dt} = \sum_{j \neq i} \mathbf{F}_i(\theta_j)$$

for all $i \notin S$, with $\theta_i(0) = \theta_i$. For $i \in S$, $\theta_i(t) = \theta_i$.

For the following theorem, we assume that $\boldsymbol{\theta}$ is fixed.

▶ **Theorem 5.** *Let Φ be a symmetric potential and L be as in (1). Running continuous gradient descent on $\frac{1}{2}L$ with respect to θ, initialized at $(\theta_1, ..., \theta_k)$ produces the same dynamics as Electron-Proton Dynamics under 2Φ with fixed particles at $w_1, ..., w_k$ with respective charges $b_1, .., b_k$ and moving particles at $\theta_1, ..., \theta_k$ with respective charges $a_1, ..., a_k$.*

3 Earnshaw's Theorem and Harmonic Potentials

When running gradient descent on a non-convex loss, we often can and do get stuck at a local minima. In this section, we use second-order information to deduce that for certain

classes of potentials, there are no spurious local minima. The potentials In this section are often *unbounded and un-realizable*. However, in the next section, we apply insights developed here to derive similar convergence results for approximations of these potentials.

Earnshaw's theorem in electrodynamics shows that there is no stable local minima for electron-proton dynamics. This hinges on the property that the electric potential $\Phi(\theta, w) = \|\theta - w\|^{2-d}, d \neq 2$ is harmonic, with $d = 3$ in natural setting. If $d = 2$, we instead have $\Phi(\theta, w) = -\ln(\|\theta - w\|)$. First, we notice that this is a symmetric loss, and our usual loss in (1) has constant terms that can be dropped to further simplify.

$$\overline{L}(a, \theta) = 2\sum_{i=1}^{k}\sum_{i<j} a_i a_j \Phi(\theta_i, \theta_j) + 2\sum_{i=1}^{k}\sum_{j=1}^{k} a_i b_j \Phi(\theta_i, w_j) \tag{2}$$

▶ **Definition 6.** $\Phi(\theta, w)$ is a **harmonic** potential on Ω if $\Delta_\theta \Phi(\theta, w) = 0$ for all $\theta \in \Omega$, except possibly at $\theta = w$.

▶ **Definition 7.** Let $\Omega \subseteq \mathbb{R}^d$ and consider a function $f : \Omega \to \mathbb{R}$. A critical point $x^* \in \Omega$ is a **local minimum** if there exists $\epsilon > 0$ such that $f(x^* + v) \geq f(x^*)$ for all $\|v\| \leq \epsilon$. It is a **strict local minimum** if the inequality is strict for all $\|v\| \leq \epsilon$.

▶ **Fact 8.** *Let x^* be a critical point of a function $f : \Omega \to \mathbb{R}$ such that f is twice differentiable at x^*. Then, if x^* is a local minimum then $\lambda_{min}(\nabla^2 f(x^*)) \geq 0$. Moreover, if $\lambda_{min}(\nabla^2 f(x^*)) > 0$, then x^* is a strict local minimum.*

Note that if $\lambda_{min}(\nabla^2 f(x^*)) < 0$ then moving along the direction of the corresponding eigenvector decreases f locally. If Φ is harmonic then it can be shown the trace of its Hessian is 0 so if there is any non zero eigenvalue then at least one eigenvalue is negative. This idea results in the following known theorem (see full proof in supplementary material) that is applicable to the electric potential function $1/r$ in 3-dimensions since is harmonic. It implies that a configuration of n electrons and n protons cannot be in a strict local minimum even if one of the mobile charges is isolated (however note that this potential function goes to ∞ at $r = 0$ and may not be realizable).

▶ **Theorem 9.** (Earnshaw's Theorem. See [2]) *Let $\mathcal{M} = \mathbb{R}^d$ and let Φ be harmonic and L be as in* (2)*. Then, L admits no differentiable strict local minima.*

Note that the Hessian of a harmonic potential can be identically zero. To avoid this possibility we generalize harmonic potentials.

3.1 λ-Harmonic Potentials

In order to relate our loss function with its Laplacian, we consider potentials that are non-negative eigenfunctions of the Laplacian operator. Since the zero eigenvalue case simply gives rise to harmonic potentials, we restrict our attention to positive eigenfunctions.

▶ **Definition 10.** A potential Φ is **λ-harmonic** on Ω if there exists $\lambda > 0$ such that for every $\theta \in \Omega$, $\Delta_\theta \Phi(\theta, w) = \lambda \Phi(\theta, w)$, except possibly at $\theta = w$.

Note that there are realizable versions of these potentials; for example $\Phi(a, b) = e^{-\|a-b\|_1}$ in $\mathbb{R}^1$. In the next section, we construct realizable potentials that are λ-harmonic almost everywhere except when θ and w are very close.

▶ **Theorem 11.** *Let Φ be λ-harmonic and L be as in* (1)*. Then, L admits no local minima $(\boldsymbol{a}, \boldsymbol{\theta})$, except when $L(\boldsymbol{a}, \boldsymbol{\theta}) = L(0, \boldsymbol{\theta})$ or $\theta_i = w_j$ for some i, j.*

Proof. Let $(\boldsymbol{a}, \boldsymbol{\theta})$ be a critical point of L. On the contrary, we assume that $\theta_i \neq w_j$ for all i, j. WLOG, we can partition $[k]$ into $S_1, ..., S_r$ such that for all $u \in S_i, v \in S_j$, we have $\theta_u = \theta_v$ iff $i = j$. Let $S_1 = \{\theta_1, \dots, \theta_l\}$. We consider changing all $\theta_1, \dots, \theta_l$ by the same v and define $H(\mathbf{a}, v) = L(\mathbf{a}, \theta_1 + v, ..., \theta_l + v, \theta_{l+1} \dots, \theta_k)$.

The optimality conditions on $\mathbf{a}$ are $0 = \frac{\partial L}{\partial a_i} = 2\sum_j a_j \Phi(\theta_i, \theta_j) + 2\sum_{j=1}^k b_j \Phi(\theta_i, w_j)$. Thus, by the definition of λ-harmonic potentials, we may differentiate as $\theta_i \neq w_j$ and compute the Laplacian as

$$\begin{aligned}\Delta_v H &= \lambda \sum_{i=1}^{l} a_i \left(2\sum_{j=1}^{k} b_j \Phi(\theta_i, w_j) + 2\sum_{j=l+1}^{k} a_j \Phi(\theta_i, \theta_j) \right) \\ &= \lambda \sum_{i=1}^{l} a_i \left(-2\sum_{j=1}^{l} a_j \Phi(\theta_i, \theta_j) \right) = -2\lambda \sum_{i=1}^{l} a_i \left(\sum_{j=1}^{l} a_j \right) = -2\lambda \left(\sum_{i=1}^{l} a_i \right)^2\end{aligned}$$

If $\sum_{i=1}^l a_i \neq 0$, then we conclude that the Laplacian is strictly negative, so we are not at a local minimum. Similarly, we can conclude that for each S_i, $\sum_{u \in S_i} a_u = 0$. In this case, since $\sum_{i=1}^k a_i \sigma(\theta_i, x) = 0$, $L(\boldsymbol{a}, \boldsymbol{\theta}) = L(0, \boldsymbol{\theta})$. ◀

4 Realizable Potentials with Convergence Guarantees

In this section, we derive convergence guarantees for realizable potentials that are almost λ-harmonic, specifically, they are λ-harmonic outside of a small neighborhood around the origin. First, we prove the existence of activation functions such that the corresponding potentials are almost λ-harmonic. Then, we reason about the Laplacian of our loss, as in the previous section, to derive our guarantees. We show that at a stable minima, each of the θ_i is close to some w_j in the target network. We may end up with a many to one mapping of the learned hidden weights to the true hidden weights, instead of a bijection. To make sure that $\|a\|$ remains controlled throughout the optimization process, we add a quadratic regularization term to L and instead optimize $G = L + \|a\|^2$.

Our optimization procedure is a slightly altered version of gradient descent, where we incorporate a second-order method (which we call Hessian descent as in Algorithm 1) that is used when the gradient is small and progress is slow. The descent algorithm (Algorithm 2) allows us to converge to points with small gradient and small negative curvature. Namely, for smooth functions, in poly$(1/\epsilon)$ iterations, we reach a point in $\mathcal{M}_{G,\epsilon}$, where

$$\mathcal{M}_{G,\epsilon} = \left\{ x \in \mathcal{M} \middle| \|\nabla G(x)\| \leq \epsilon \text{ and } \lambda_{min}(\nabla^2 G(x)) \geq -\epsilon \right\}$$

We show that if $(\boldsymbol{a}, \boldsymbol{\theta})$ is in $\mathcal{M}_{G,\epsilon}$ for ϵ small, then θ_i is close to w_j for some j. Finally, we show how to initialize $(\boldsymbol{a^{(0)}}, \boldsymbol{\theta^{(0)}})$ and run second-order GD to converge to $\mathcal{M}_{G,\epsilon}$, proving our main theorem.

▶ **Theorem 12.** *Let $\mathcal{M} = \mathbb{R}^d$ for $d \equiv 3 \mod 4$ and $k = \text{poly}(d)$. For all $\epsilon \in (0, 1)$, we can construct an activation σ_ϵ such that if $w_1, ..., w_k \in \mathbb{R}^d$ with w_i randomly chosen from $w_i \sim \mathcal{N}(\mathbf{0}, O(d \log d)\mathbf{I_{d \times d}})$ and $b_1, ..., b_k$ be randomly chosen at uniform from $[-1, 1]$, then with high probability, we can choose an initial point $(\boldsymbol{a^{(0)}}, \boldsymbol{\theta^{(0)}})$ such that after running SecondGD (Algorithm 2) on the regularized objective $G(\boldsymbol{a}, \boldsymbol{\theta})$ for at most $(d/\epsilon)^{O(d)}$ iterations, there exists an i, j such that $\|\theta_i - w_j\| < \epsilon$.*

We start by stating a lemma concerning the construction of an almost λ-harmonic function on $\mathbb{R}^d$. The construction is given in Appendix B and uses a linear combination of realizable

Algorithm 1 $x = HD(L, x_0, T, \alpha)$

Input: $L : \mathcal{M} \to \mathbb{R}$; $x_0 \in \mathcal{M}$; $T \in \mathbb{N}$; $\alpha \in \mathbb{R}$
Initialize $x \leftarrow x_0$
for $i = 1$ **to** T **do**
  Find unit eigenvector v_{min} corresponding to $\lambda_{min}(\nabla^2 f(x))$
  $\beta \leftarrow -\alpha\lambda_{min}(\nabla^2 f(x))\text{sign}(\nabla f(x)^T v_{min})$
  $x \leftarrow x + \beta v_{min}$

Algorithm 2 $x = SecondGD(L, x_0, T, \alpha, \eta, \gamma)$

Input: $L : \mathcal{M} \to \mathbb{R}$; $x_0 \in \mathcal{M}$; $T \in \mathbb{N}$; $\alpha, \eta, \gamma \in \mathbb{R}$
for $i = 1$ **to** T **do**
  if $\|\nabla L(x_{i-1})\| \geq \eta$ **then** $x_i \leftarrow x_{i-1} - \alpha\nabla L(x_{i-1})$
  else $x_i \leftarrow HD(L, x_{i-1}, 1, \alpha)$
  if $L(x_i) \geq L(x_{i-1}) - min(\alpha\eta^2/2, \alpha^2\gamma^3/2)$ **then return** x_{i-1}

potentials that correspond to an activation function of the indicator function of a n-sphere. By using Fourier analysis and Theorem 3, we can finish the construction of our almost λ-harmonic potential.

▶ **Lemma 13.** *Let $\mathcal{M} = \mathbb{R}^d$ for $d \equiv 3 \mod 4$. Then, for any $\epsilon \in (0, 1)$, we can construct a radial activation $\sigma_\epsilon(r)$ such that the corresponding radial potential $\Phi_\epsilon(r)$ is λ-harmonic for $r \geq \epsilon$.*

Furthermore, we have ${\Phi_\epsilon}^{(d-1)}(r) \geq 0$ for all $r > 0$, ${\Phi_\epsilon}^{(k)}(r) \geq 0$, and ${\Phi_\epsilon}^{(k+1)}(r) \leq 0$ for all $r > 0$ and $d - 3 \geq k \geq 0$ even.

When $\lambda = 1$, $|\Phi_\epsilon^{(k)}(r)| \leq O((d/\epsilon)^{2d})$ for all $0 \leq k \leq d - 1$. And when $r \geq \epsilon$, $\Omega(e^{-r}r^{2-d}(d/\epsilon)^{-2d}) \leq \Phi_\epsilon(r) \leq O((1 + r)^d e^{1-r}(r)^{2-d})$ and $\Omega(e^{-r}r^{1-d}(d/\epsilon)^{-2d}) \leq |\Phi'_\epsilon(r)| \leq O((d + r)(1 + r)^d e^{1-r}r^{1-d})$

Our next lemma use the almost λ-harmonic properties to show that at an almost stationary point of G, we must have converged close to some w_j as long as our charges a_i are not too small. The proof is similar to Theorem 11. Then, the following lemma relates the magnitude of the charges a_i to the progress made in the objective function.

▶ **Lemma 14.** *Let $\mathcal{M} = \mathbb{R}^d$ for $d \equiv 3 \mod 4$ and let G be the regularized loss corresponding to the activation σ_ϵ given by Lemma 13 with $\lambda = 1$. For any $\epsilon \in (0, 1)$ and $\delta \in (0, 1)$, if $(\boldsymbol{a}, \boldsymbol{\theta}) \in \mathcal{M}_{G,\delta}$, then for all i, either 1) there exists j such that $\|\theta_i - w_j\| < k\epsilon$ or 2) $a_i^2 < 2kd\delta$.*

▶ **Lemma 15.** *Assume the conditions of Lemma 14. If $\sqrt{G(\mathbf{a}, \boldsymbol{\theta})} \leq \sqrt{G(\mathbf{0}, \mathbf{0})} - \delta$ and $(\boldsymbol{a}, \boldsymbol{\theta}) \in \mathcal{M}_{G,\delta^2/(2k^3d)}$, then there exists some i, j such that $\|\theta_i - w_j\| < k\epsilon$.*

Finally, we guarantee that our initialization substantially decreases our objective function. Together with our previous lemmas, it will imply that we must be close to some w_j upon convergence. This is the overview of the proof of Theorem 12, presented below.

▶ **Lemma 16.** *Assume the conditions of Theorem 12 and Lemma 14. With high probability, we can initialize $(\boldsymbol{a}^{(\mathbf{0})}, \boldsymbol{\theta}^{(\mathbf{0})})$ such that $\sqrt{G(\mathbf{a}^{(\mathbf{0})}, \boldsymbol{\theta}^{(\mathbf{0})})} \leq \sqrt{G(\mathbf{0}, \mathbf{0})} - \delta$ with $\delta = (d/\epsilon)^{-O(d)}$.*

Proof of Theorem 12. Let our potential $\Phi_{\epsilon/k}$ be the one as constructed in Lemma 13 that is 1-harmonic for all $r \geq \epsilon/k$ and as always, $k = \text{poly}(d)$. First, by Lemma 16, we can

Algorithm 3 Node-wise Descent Algorithm

Input: $(\boldsymbol{a}, \boldsymbol{\theta}) = (a_1, ..., a_k, \theta_1, ..., \theta_k), a_i \in \mathbb{R}, \theta_i \in \mathcal{M}$; $T \in \mathbb{N}$; L; $\alpha, \eta, \gamma \in \mathbb{R}$;
for $i = 1$ **to** k **do**
  Initialize (a_i, θ_i)
  $(a_i, \theta_i) = SecondGD\left(L_{a_i,\theta_i}, (a_i, \theta_i), T, \alpha, \eta, \gamma\right)$
return $a = (a_1, ..., a_k), \theta = (\theta_1, ..., \theta_k)$

initialize $(\boldsymbol{a^{(0)}}, \boldsymbol{\theta^{(0)}})$ such that $\sqrt{G(\boldsymbol{a^{(0)}}, \boldsymbol{\theta^{(0)}})} \leq \sqrt{G(\mathbf{0}, \mathbf{0})} - \delta$ for $\delta = (d/\epsilon)^{-O(d)}$. If we set $\alpha = (d/\epsilon)^{-O(d)}$ and $\eta = \gamma = \delta^2/(2k^3 d)$, then running Algorithm 2 will terminate and return some $(\boldsymbol{a}, \boldsymbol{\theta})$ in at most $(d/\epsilon)^{O(d)}$ iterations. This is because our algorithm ensures that our objective function decreases by at least $\min(\alpha\eta^2/2, \alpha^2\gamma^3/2)$ at each iteration, $G(\mathbf{0}, \mathbf{0})$ is bounded by $O(k)$, and $G \geq 0$ is non-negative.

Let $\boldsymbol{\theta} = (\theta_1, ...\theta_k)$. If there exists θ_i, w_j such that $\|\theta_i - w_j\| < \epsilon$, then we are done. Otherwise, we claim that $(\boldsymbol{a}, \boldsymbol{\theta}) \in \mathcal{M}_{G,\delta^2/(2k^3 d)}$. For the sake of contradiction, assume otherwise. By our algorithm termination conditions, then it must be that after one step of gradient or Hessian descent from $(\boldsymbol{a}, \boldsymbol{\theta})$, we reach some $(\boldsymbol{a'}, \boldsymbol{\theta'})$ and $G(\boldsymbol{a'}, \boldsymbol{\theta'}) > G(\boldsymbol{a}, \boldsymbol{\theta}) - \min(\alpha\eta^2/2, \alpha^2\gamma^3/2)$.

Now, Lemma 13 ensures all first three derivatives of $\Phi_{\epsilon/k}$ are bounded by $O((dk/\epsilon)^{2d})$, except at $w_1, ..., w_k$. Furthermore, since there do not exist θ_i, w_j such that $\|\theta_i - w_j\| < \epsilon$, G is three-times continuously differentiable within a $\alpha(dk/\epsilon)^{2d} = (d/\epsilon)^{-O(d)}$ neighborhood of $\boldsymbol{\theta}$. Therefore, by Lemma 18 and 19 in the appendix, we must have $G(\boldsymbol{a'}, \boldsymbol{\theta'}) \leq G(\boldsymbol{a}, \boldsymbol{\theta}) - \min(\alpha\eta^2/2, \alpha^2\gamma^3/2)$, a contradiction. Lastly, since our algorithm maintains that our objective function is decreasing, so $\sqrt{G(\boldsymbol{a}, \boldsymbol{\theta})} \leq \sqrt{G(\mathbf{0}, \mathbf{0})} - \delta$. Finally, we conclude by Lemma 15. ◀

4.1 Node-by-Node Analysis

We cannot easily analyze the convergence of gradient descent to the global minima when all θ_i are simultaneously moving since the pairwise interaction terms between the θ_i present complications, even with added regularization. Instead, we run a greedy node-wise descent (Algorithm 3) to learn the hidden weights, i.e. we run a descent algorithm with respect to (a_i, θ_i) sequentially. The main idea is that after running SGD with respect to θ_1, θ_1 should be close to some w_j for some j. Then, we can carefully induct and show that θ_2 must be some other w_k for $k \neq j$ and so on.

Let $L_1(a_1, \theta_1)$ be the objective L restricted to a_1, θ_1 being variable, and $a_2, ..., a_k = 0$ are fixed. The tighter control on the movements of θ_1 allows us to remove our regularization. While our previous guarantees before allow us to reach a ϵ-neighborhood of w_j when running SGD on L_1, we will strengthen our guarantees to reach a $(d/\epsilon)^{-O(d)}$-neighborhood of w_j, by reasoning about the first derivatives of our potential in an ϵ-neighborhood of w_j. By similar argumentation as before, we will be able to derive the following convergence guarantees for node-wise training.

▶ **Theorem 17.** *Let $\mathcal{M} = \mathbb{R}^d$ and $d \equiv 3 \mod 4$ and let L be as in 1 and $k = \text{poly}(d)$. For all $\epsilon \in (0, 1)$, we can construct an activation σ_ϵ such that if $w_1, ..., w_k \in \mathbb{R}^d$ with w_i randomly chosen from $w_i \sim \mathcal{N}(\mathbf{0}, O(d \log d)\mathbf{I_{d\times d}})$ and $b_1, ..., b_k$ be randomly chosen at uniform from $[-1, 1]$, then with high probability, after running nodewise descent (Algorithm 3) on the objective L for at most $(d/\epsilon)^{O(d)}$ iterations, $(\boldsymbol{a}, \boldsymbol{\theta})$ is in a $(d/\epsilon)^{-O(d)}$ neighborhood of the global minima.*

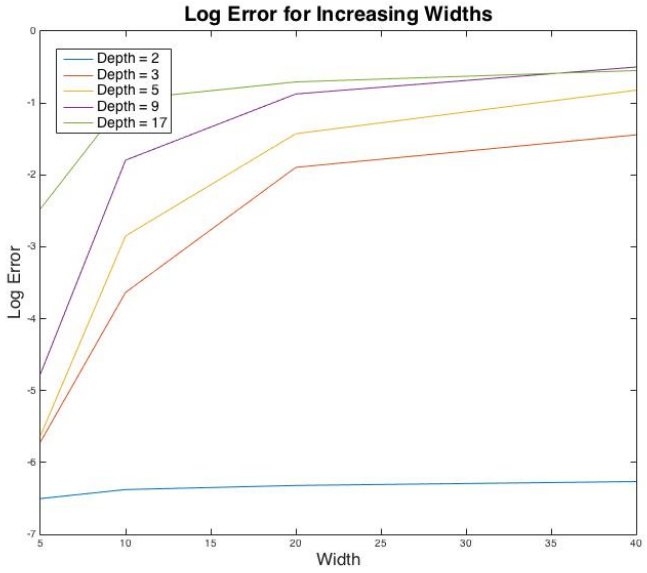


Figure 2 Test Error of Varying-Depth Networks vs. Width

Table 2 Test Error of Learning Neural Networks of Various Depth and Width

	Width 5	Width 10	Width 20	Width 40
Depth 2	0.0015	0.0017	0.0018	0.0019
Depth 3	0.0033	0.0264	0.1503	0.2362
Depth 5	0.0036	0.0579	0.2400	0.4397
Depth 9	0.0085	0.1662	0.4171	0.6071
Depth 17	0.0845	0.3862	0.4934	0.5777

5 Experiments

For our experiments, our training data is given by $(x_i, f(x_i))$, where x_i are randomly chosen from a standard Gaussian in $\mathbb{R}^d$ and f is a randomly generated neural network with weights chosen from a standard Gaussian. We run gradient descent (Algorithm 4) on the empirical loss, with stepsize around $\alpha = 10^{-5}$, for $T = 10^6$ iterations. The nonlinearity used at each node is sigmoid from -1 to 1, including the output node, unlike the assumptions in the theoretical analysis. A random guess for the network will result in a mean squared error of around 1. Our experiments (see Fig 1) show that for depth-2 neural networks, even with non-linear outputs, the training error diminishes quickly to under 0.002. This seems to hold even when the width, the number of hidden nodes, is substantially increased (even up to 125 nodes), but depth is held constant; although as the number of nodes increases, the rate of decrease is slower. This substantiates our claim that depth-2 neural networks are learnable.

However, it seems that for depth greater than 2, the test error becomes significant when width is high (see Fig 2). Even for depth 3 networks, the increase in depth impedes the learnability of the neural network and the training error does not get close enough to 0. It seems that for neural networks with greater depth, positive convergence results in practice are elusive. We note that we are using training error as a measure of success, so it's possible that the true underlying parameters are not learned.

References

1 Alexandr Andoni, Rina Panigrahy, Gregory Valiant, and Li Zhang. Learning polynomials with neural networks. In *International Conference on Machine Learning*, pages 1908–1916, 2014.

2 Vladimir I Arnold, Valery V Kozlov, and Anatoly I Neishtadt. Mathematical aspects of classical and celestial mechanics. *Encyclopaedia Math. Sci*, 3:1–291, 1985.

3 Sanjeev Arora, Aditya Bhaskara, Rong Ge, and Tengyu Ma. Provable bounds for learning some deep representations. In *ICML*, pages 584–592, 2014.

4 Peter Auer, Mark Herbster, and Manfred K. Warmuth. Exponentially many local minima for single neurons. In David S. Touretzky, Michael Mozer, and Michael E. Hasselmo, editors, *Advances in Neural Information Processing Systems 8, NIPS, Denver, CO, November 27-30, 1995*, pages 316–322. MIT Press, 1995. URL: `http://papers.nips.cc/paper/1028-exponentially-many-local-minima-for-single-neurons`.

5 Avrim Blum and Ronald L. Rivest. Training a 3-node neural network is np-complete. In David Haussler and Leonard Pitt, editors, *Proceedings of the First Annual Workshop on Computational Learning Theory, COLT '88, Cambridge, MA, USA, August 3-5, 1988.*, pages 9–18. ACM/MIT, 1988. URL: `http://dl.acm.org/citation.cfm?id=93033`.

6 Martin L Brady, Raghu Raghavan, and Joseph Slawny. Back propagation fails to separate where perceptrons succeed. *IEEE Transactions on Circuits and Systems*, 36(5):665–674, 1989.

7 Alon Brutzkus and Amir Globerson. Globally optimal gradient descent for a convnet with gaussian inputs. *arXiv preprint arXiv:1702.07966*, 2017.

8 Anna Choromanska, Mikael Henaff, Michael Mathieu, Gérard Ben Arous, and Yann LeCun. The loss surfaces of multilayer networks. In *AISTATS*, 2015.

9 Djork-Arné Clevert, Thomas Unterthiner, and Sepp Hochreiter. Fast and accurate deep network learning by exponential linear units (elus). *CoRR*, abs/1511.07289, 2015. `arXiv:1511.07289`.

10 Amit Daniely. Sgd learns the conjugate kernel class of the network. *arXiv preprint arXiv:1702.08503*, 2017.

11 Yann N Dauphin, Razvan Pascanu, Caglar Gulcehre, Kyunghyun Cho, Surya Ganguli, and Yoshua Bengio. Identifying and attacking the saddle point problem in high-dimensional non-convex optimization. In *Advances in neural information processing systems*, pages 2933–2941, 2014.

12 Rong Ge, Furong Huang, Chi Jin, and Yang Yuan. Escaping from saddle points-online stochastic gradient for tensor decomposition. In *COLT*, pages 797–842, 2015.

13 Rong Ge, Jason D Lee, and Tengyu Ma. Learning one-hidden-layer neural networks with landscape design. *arXiv preprint arXiv:1711.00501*, 2017.

14 Surbhi Goel, Varun Kanade, Adam Klivans, and Justin Thaler. Reliably learning the relu in polynomial time. *arXiv preprint arXiv:1611.10258*, 2016.

15 Majid Janzamin, Hanie Sedghi, and Anima Anandkumar. Generalization bounds for neural networks through tensor factorization. *CoRR*, abs/1506.08473, 2015. `arXiv:1506.08473`.

16 Chi Jin, Rong Ge, Praneeth Netrapalli, Sham M Kakade, and Michael I Jordan. How to escape saddle points efficiently. *arXiv preprint arXiv:1703.00887*, 2017.

17 Kenji Kawaguchi. Deep learning without poor local minima. In *Advances in Neural Information Processing Systems*, pages 586–594, 2016.

18 Adam R Klivans and Alexander A Sherstov. Cryptographic hardness for learning intersections of halfspaces. In *2006 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS'06)*, pages 553–562. IEEE, 2006.

19 Yuanzhi Li and Yang Yuan. Convergence analysis of two-layer neural networks with relu activation. *arXiv preprint arXiv:1705.09886*, 2017.

20 Roi Livni, Shai Shalev-Shwartz, and Ohad Shamir. On the computational efficiency of training neural networks. In *Advances in Neural Information Processing Systems*, pages 855–863, 2014.

21 Anish Shah, Eashan Kadam, Hena Shah, Sameer Shinde, and Sandip Shingade. Deep residual networks with exponential linear unit. In *Proceedings of the Third International Symposium on Computer Vision and the Internet*, pages 59–65. ACM, 2016.

22 Shai Shalev-Shwartz, Ohad Shamir, and Karthik Sridharan. Learning kernel-based halfspaces with the 0-1 loss. *SIAM Journal on Computing*, 40(6):1623–1646, 2011.

23 Daniel Soudry and Yair Carmon. No bad local minima: Data independent training error guarantees for multilayer neural networks. *CoRR*, abs/1605.08361, 2016. `arXiv:1605.08361`.

24 Yuandong Tian. An analytical formula of population gradient for two-layered relu network and its applications in convergence and critical point analysis. *arXiv preprint arXiv:1703.00560*, 2017.

25 Qiuyi Zhang, Rina Panigrahy, and Sushant Sachdeva. Electron-proton dynamics in deep learning. *CoRR*, abs/1702.00458, 2017. `arXiv:1702.00458`.

26 Yuchen Zhang, Jason D Lee, and Michael I Jordan. l1-regularized neural networks are improperly learnable in polynomial time. In *International Conference on Machine Learning*, pages 993–1001, 2016.

27 Yuchen Zhang, Jason D. Lee, Martin J. Wainwright, and Michael I. Jordan. Learning halfspaces and neural networks with random initialization. *CoRR*, abs/1511.07948, 2015. `arXiv:1511.07948`.

A Electron-Proton Dynamics

▶ **Theorem 5.** *Let Φ be a symmetric potential and L be as in* (1)*. Running continuous gradient descent on $\frac{1}{2}L$ with respect to θ, initialized at $(\theta_1, ..., \theta_k)$ produces the same dynamics as Electron-Proton Dynamics under 2Φ with fixed particles at $w_1, ..., w_k$ with respective charges $b_1, .., b_k$ and moving particles at $\theta_1, ..., \theta_k$ with respective charges $a_1, ..., a_k$.*

Proof. The initial values are the same. Notice that continuous gradient descent on $L(\boldsymbol{a}, \boldsymbol{\theta})$ with respect to θ produces dynamics given by $\frac{d\theta_i(t)}{dt} = -\nabla_{\theta_i} L(\boldsymbol{a}, \boldsymbol{\theta})$. Therefore,

$$\frac{d\theta_i(t)}{dt} = -2\sum_{j\neq i} a_i a_j \nabla_{\theta_i}\Phi(\theta_i, \theta_j) - 2\sum_{j=1}^{k} a_i b_j \nabla_{\theta_i}\Phi(\theta_i, w_j)$$

And gradient descent does not move w_i. By definition, the dynamics corresponds to Electron-Proton Dynamics as claimed. ◀

B Realizable Potentials

This section can be found in the full version of this paper on ArXiv [25].

C Earnshaw's Theorem

▶ **Theorem 9.** (Earnshaw's Theorem. See [2]) *Let $\mathcal{M} = \mathbb{R}^d$ and let Φ be harmonic and L be as in* (2)*. Then, L admits no differentiable strict local minima.*

Proof. If $(\boldsymbol{a}, \boldsymbol{\theta})$ is a differentiable strict local minima, then for any i, we must have

$$\nabla_{\theta_i} L = 0, \text{ and } \mathrm{Tr}(\nabla^2_{\theta_i} L) > 0.$$

Algorithm 4 $x = GD(L, x_0, T, \alpha)$

Input: $L : \mathcal{M} \to \mathbb{R}$; $x_0 \in \mathcal{M}$; $T \in \mathbb{N}$; $\alpha \in \mathbb{R}$
Initialize $x = x_0$
for $i = 1$ **to** T **do**
  $x = x - \alpha \nabla L(x)$
  $x = \Pi_{\mathcal{M}} x$

Since Φ is harmonic, we also have

$$\mathrm{Tr}(\nabla^2_{\theta_i} L(\theta_1, ..., \theta_n)) = \Delta_{\theta_i} L = 2\sum_{j \neq i} a_i a_j \Delta_{\theta_i} \Phi(\theta_i, \theta_j) + 2\sum_{j=1}^{k} a_i b_j \Delta_{\theta_i} \Phi(\theta_i, w_j) = 0,$$

which is a contradiction. In the first line, there is a factor of 2 by symmetry. ◀

D Descent Lemmas and Iteration Bounds

▶ **Lemma 18.** *Let $f : \Omega \to \mathbb{R}$ be a thrice differentiable function such that $|f(y)| \leq B_0, \|\nabla f(y)\| \leq B_1, \|\nabla^2 f(y)\| \leq B_2, \|\nabla^2 f(z) - \nabla^2 L(y)\| \leq B_3 \|z - y\|$ for all y, z in a (αB_1)-neighborhood of x. If $\|\nabla f(x)\| \geq \eta$ and x' is reached after one iteration of gradient descent (Algorithm 4) with stepsize $\alpha \leq \frac{1}{B_2}$, then $\|x' - x\| \leq \alpha B_1$ and $f(x') \leq f(x) - \alpha\eta^2/2$.*

Proof. The gradient descent step is given by $x' = x - \alpha \nabla f(x)$. The bound on $\|x' - x\|$ is clear since $\|\nabla f(x)\| \leq B_1$.

$$\begin{aligned} f(x') &\leq f(x) - \alpha \nabla f(x)^T \nabla f(x)^T + \alpha^2 \frac{B_2}{2} \|\nabla f(x)\|^2 \\ &\leq f(x) - (\alpha - \alpha^2 \frac{B_2}{2})\eta^2 \end{aligned}$$

For $0 \leq \alpha \leq \frac{1}{B_2}$, we have $\alpha - \alpha^2 B_2/2 \geq \alpha/2$, and our lemma follows. ◀

▶ **Lemma 19.** *Let $f : \Omega \to \mathbb{R}$ be a thrice differentiable function such that $|f(y)| \leq B_0, \|\nabla f(y)\| \leq B_1, \|\nabla^2 f(y)\| \leq B_2, \|\nabla^2 f(z) - \nabla^2 L(y)\| \leq B_3 \|z - y\|$ for all y, z in a (αB_2)-neighborhood of x. If $\lambda_{min}(\nabla^2 f(x)) \leq -\gamma$ and x' is reached after one iteration of Hessian descent (Algorithm 1) with stepsize $\alpha \leq \frac{1}{B_3}$, then $\|x' - x\| \leq \alpha B_2$ and $f(x') \leq f(x) - \alpha^2\gamma^3/2$.*

Proof. The gradient descent step is given by $x' = x + \beta v_{min}$, where v_{min} is the unit eigenvector corresponding to $\lambda_{min}(\nabla^2 f(x))$ and $\beta = -\alpha \lambda_{min}(\nabla^2 f(x)) sgn(\nabla f(x)^T v_{min})$. Our bound on $\|x' - x\|$ is clear since $|\lambda_{min}(\nabla^2 f(x))| \leq B_2$.

$$\begin{aligned} f(x') &\leq f(x) + \beta \nabla f(x)^T v_{min} + \beta^2 v_{min}^T \nabla^2 f(x) v_{min} + \frac{B_3}{6} |\beta|^3 \|v_{min}\|^3 \\ &\leq f(x) - |\beta|^2 \gamma + \frac{B_3}{6} |\beta|^3 \end{aligned}$$

The last inequality holds since the sign of β is chosen so that $\beta \nabla f(x)^T v_{min} \leq 0$. Now, since $|\beta| = \alpha\gamma \leq \frac{\gamma}{B_3}$, $-|\beta|^2\gamma + \frac{B_3}{6}|\beta|^3 \leq -\alpha^2\gamma^3/2$. ◀

E Convergence of Almost λ-Harmonic Potentials

▶ **Lemma 20.** *Let $\mathcal{M} = \mathbb{R}^d$ for $d \equiv 3 \mod 4$ and let G be the regularized loss corresponding to the activation σ_ϵ given by Lemma 13 with $\lambda = 1$. For any $\epsilon \in (0, 1)$ and $\delta \in (0, 1)$,*

if $(\boldsymbol{a}, \boldsymbol{\theta}) \in \mathcal{M}_{G,\delta}$*, then for all* i*, either 1) there exists* j *such that* $\|\theta_i - w_j\| < k\epsilon$ *or 2)* $a_i^2 < 2kd\delta$*.*

Proof. The proof is similar to Theorem 11. Let Φ_ϵ be the realizable potential in 13 such that $\Phi_\epsilon(r)$ is λ-harmonic when $r \geq \epsilon$ with $\lambda = 1$. Note that $\Phi_\epsilon(0) = 1$ is normalized. And let $(\boldsymbol{a}, \boldsymbol{\theta}) \in \mathcal{M}_{G,\delta}$.

WLOG, consider θ_1 and a initial set $S_0 = \{\theta_1\}$ containing it. For a finite set of points S and a point x, define $d(x, S) = \min_{y \in S} \|x - y\|$. Then, we consider the following set growing process. If there exists $\theta_i, w_i \notin S_j$ such that $d(\theta_i, S_j) < \epsilon$ or $d(w_i, S_j) < \epsilon$, add θ_i, w_i to S_j to form S_{j+1}. Otherwise, we stop the process. We grow S_0 to until the process terminates and we have the grown set S.

If there is some $w_j \in S$, then it must be the case that there exists $j_1, \cdots j_q$ such that $\|\theta_1 - \theta_{j_1}\| < \epsilon$ and $\|\theta_{j_i} - \theta_{j_{i+1}}\| < \epsilon$, and $\|\theta_{j_q} - w_j\| < \epsilon$ for some w_j. So, there exists j, such that $\|\theta_1 - w_j\| < k\epsilon$.

Otherwise, notice that for each $\theta_i \in S$, $\|w_j - \theta_i\| \geq \epsilon$ for all j, and $\|\theta_i - \theta_j\| \geq \epsilon$ for all $\theta_j \notin S$. WLOG, let $S = \{\theta_1, \ldots, \theta_l\}$.

We consider changing all $\theta_1, \ldots, \theta_l$ by the same v and define

$$H(\mathbf{a}, v) = G(\mathbf{a}, \theta_1 + v, ..., \theta_l + v, \theta_{l+1} \ldots, \theta_k).$$

The optimality conditions on $\mathbf{a}$ are

$$\left|\frac{\partial H}{\partial a_i}\right| = |4a_i + 2\sum_{j \neq i} a_j \Phi_\epsilon(\theta_i, \theta_j) + 2\sum_{j=1}^{k} b_j \Phi_\epsilon(\theta_i, w_j)| \leq \delta$$

Next, since $\Phi_\epsilon(r)$ is λ-harmonic for $r \geq \epsilon$, we may calculate the Laplacian of H as

$$\begin{aligned}
\Delta_v H &= \sum_{i=1}^{l} \lambda \left(2\sum_{j=1}^{k} a_i b_j \Phi_\epsilon(\theta_i, w_j) + 2\sum_{j=l+1}^{k} a_i a_j \Phi_\epsilon(\theta_i, \theta_j) \right) \\
&\leq \sum_{i=1}^{l} \lambda \left(-4a_i^2 - 2\sum_{j=1, j \neq i}^{l} a_i a_j \Phi_\epsilon(\theta_i, \theta_j) \right) + \delta \sum_{i=1}^{l} \lambda |a_i| \\
&= -2\lambda\, \mathbb{E}\left[\left(\sum_{i=1}^{l} a_i \sigma(\theta_i, X) \right)^2 \right] - 2\lambda \sum_{i=1}^{l} a_i^2 + \delta\lambda \sum_{i=1}^{l} |a_i|
\end{aligned}$$

The second line follows from our optimality conditions and the third line follows from completing the square. Since $(\boldsymbol{a}, \boldsymbol{\theta}) \in \mathcal{M}_{G,\delta}$, we have $\Delta_v H \geq -2kd\delta$. Let $S = \sum_{i=1}^{l} a_i^2$. Then, by Cauchy-Schwarz, we have $-2\lambda S + \delta\lambda\sqrt{k}\sqrt{S} \geq -2kd\delta$. When $S \geq \delta^2 k$, we see that $-\lambda S \geq -2\lambda S + \delta\lambda\sqrt{k}\sqrt{S} \geq -2kd\delta$. Therefore, $S \leq 2kd\delta/\lambda$.

We conclude that $S \leq \max(\delta^2 k, 2kd\delta/\lambda) \leq 2kd\delta/\lambda$ since $\delta \leq 1 \leq 2d/\lambda$ and $\lambda = 1$. Therefore, $a_i^2 \leq 2kd\delta$. ◀

▶ **Lemma 21.** *Assume the conditions of Lemma 14. If* $\sqrt{G(\mathbf{a}, \boldsymbol{\theta})} \leq \sqrt{G(\mathbf{0}, \mathbf{0})} - \delta$ *and* $(\boldsymbol{a}, \boldsymbol{\theta}) \in \mathcal{M}_{G,\delta^2/(2k^3 d)}$*, then there exists some* i, j *such that* $\|\theta_i - w_j\| < k\epsilon$*.*

Proof. If there does not exists i, j such that $\|\theta_i - w_j\| < k\epsilon$, then by Lemma 14, this implies $a_i^2 < \delta^2/k^2$ for all i. Now, for a integrable function $f(x)$, $\|f\|_X = \sqrt{\mathbb{E}_X[f(X)^2]}$ is a norm. Therefore, if $f(x) = \sum_i b_i \sigma(w_i, x)$ be our true target function, we conclude that by triangle

inequality

$$\sqrt{G(\boldsymbol{a},\boldsymbol{\theta})} \geq \left\| \sum_{i=1}^{k} a_i \sigma(\theta_i, x) - f(x) \right\|_X \geq \|f(x)\|_X - \sum_{i=1}^{k} \|a_i \sigma(\theta_i, x)\|_X \geq \sqrt{G(\mathbf{0},\mathbf{0})} - \delta$$

This gives a contradiction, so we conclude that there must exist i, j such that θ_i is in a $k\epsilon$ neighborhood of w_j. ◀

▶ **Lemma 22.** *Assume the conditions of Theorem 12 and Lemma 14. With high probability, we can initialize* $(\boldsymbol{a^{(0)}}, \boldsymbol{\theta^{(0)}})$ *such that* $\sqrt{G(\mathbf{a^{(0)}}, \boldsymbol{\theta^{(0)}})} \leq \sqrt{G(\mathbf{0},\mathbf{0})} - \delta$ *with* $\delta = (d/\epsilon)^{-O(d)}$.

Proof. Consider choosing $\theta_1 = \mathbf{0}$ and then optimizing a_1. Given θ_1, the loss decrease is:

$$G(a_1, \mathbf{0}) - G(\mathbf{0},\mathbf{0}) = \min_{a_1} 2a_1^2 + 2\sum_{j=1}^{k} a_1 b_j \Phi_\epsilon(\mathbf{0}, w_j) = -\frac{1}{2}\left(\sum_{j=1}^{k} b_j \Phi_\epsilon(\mathbf{0}, w_j)\right)^2$$

Because w_j are random Gaussians with variance $O(d \log d)$, we have $\|w_j\| \leq O(d \log d)$ with high probability for all j. By Lemma 13, our potential satisfies $\Phi_\epsilon(\mathbf{0}, w_j) \geq (d/\epsilon)^{-O(d)}$. And since b_j are uniformly chosen in $[-1, 1]$, we conclude that with high probability over the choices of b_j, $-\frac{1}{2}\left(\sum_{j=1}^{k} b_j \Phi(\theta_1, w_j)\right)^2 \geq (d/\epsilon)^{-O(d)}$ by appealing to Chebyshev's inequality on the squared term.

Therefore, we conclude that with high probability, $G(a_1, \mathbf{0}) \leq G(\mathbf{0},\mathbf{0}) - \frac{1}{2}(d/\epsilon)^{-O(d)}$. Let $\sqrt{G(a_1,\mathbf{0})} = \sqrt{G(\mathbf{0},\mathbf{0})} - \Delta \geq 0$. Squaring and rearranging gives $\Delta \geq \frac{1}{4\sqrt{G(\mathbf{0},\mathbf{0})}}(d/\epsilon)^{-O(d)}$. Since $G(\mathbf{0},\mathbf{0}) \leq O(k) = O(\text{poly}(d))$, we are done. ◀

E.1 Node by Node Analysis

The proofs in this section can be found in the full version of this paper on ArXiv [25].

▶ **Lemma 23.** *Let* $\mathcal{M} = \mathbb{R}^d$ *for* $d \equiv 3 \mod 4$ *and let* L_1 *be the loss restricted to* (a_1, θ_1) *corresponding to the activation function* σ_ϵ *given by Lemma 13 with* $\lambda = 1$. *For any* $\epsilon \in (0,1)$ *and* $\delta \in (0,1)$, *we can construct* σ_ϵ *such that if* $(a_1, \theta_1) \in \mathcal{M}_{L_1,\delta}$, *then for all* i, *either 1) there exists* j *such that* $\|\theta_1 - w_j\| < \epsilon$ *or 2)* $a_1^2 < 2d\delta$.

▶ **Lemma 24.** *Assume the conditions of Lemma 23. If* $\sqrt{L_1(a_1,\theta_1)} \leq \sqrt{L_1(0,0)} - \delta$ *and* $(a_1, \theta_1) \in \mathcal{M}_{G,\delta^2/(2d)}$, *then there exists some* j *such that* $\|\theta_1 - w_j\| < \epsilon$.

▶ **Lemma 25.** *Assume the conditions of Theorem 27 and Lemma 23. If* $\|\theta_1 - w_j\| \leq d$ *and* $|b_j| \geq 1/\text{poly}(d)$ *and* $|a_1 - a_1^*(\theta_1)| \leq (d/\epsilon)^{-O(d)}$ *is almost optimal and for* i, $\|w_i - w_j\| \geq \Omega(d \log d)$, *then* $-\nabla_{\theta_1} L_1 = \zeta \frac{w_j - \theta_1}{\|\theta_1 - w_j\|} + \xi$ *with* $\zeta \geq \frac{1}{\text{poly}(d)}(d/\epsilon)^{-8d}$ *and* $\xi \leq (d/\epsilon)^{-O(d)}$.

▶ **Lemma 26** (Node-wise Initialization)**.** *Assume the conditions of Theorem 27 and Lemma 23. With high probability, we can initialize* $(a_1^{(0)}, \theta_1^{(0)})$ *such that* $\sqrt{L(a_1^{(0)}, \theta_1^{(0)})} \leq \sqrt{L(0,0)} - \delta$ *with* $\delta = \frac{1}{\text{poly}(d)}(d/\epsilon)^{-18d}$ *in time* $\log(d)^{O(d)}$.

▶ **Lemma 27.** *Assume the conditions of Lemma 23. Also, assume* $b_1, ..., b_k$ *are any numbers in* $[-1, 1]$ *and* $w_1, ..., w_k \in \mathbb{R}^d$ *satisfy* $\|w_i\| \leq O(d \log d)$ *for all* i *and there exists some* $|b_j| \geq 1/\text{poly}(d)$ *with* $\|w_i - w_j\| \geq \Omega(d \log d)$ *for all* i.

Then with high probability, we can choose an initial point $(a_1^{(0)}, \theta_1^{(0)})$ *such that after running SecondGD (Algorithm 2) on the restricted regularized objective* $L_1(a_1, \theta_1)$ *for at most* $(d/\epsilon)^{O(d)}$ *iterations, there exists some* w_j *such that* $\|\theta_1 - w_j\| < \epsilon$. *Furthermore, if* $|b_j| \geq 1/\text{poly}(d)$ *and* $\|w_i - w_j\| \geq \Omega(d \log d)$ *for all* i, *then* $\|\theta_1 - w_j\| < (d/\epsilon)^{-O(d)}$ *and* $|a + b_j| < (d/\epsilon)^{-O(d)}$.

▶ **Theorem 17.** *Let $\mathcal{M} = \mathbb{R}^d$ and $d \equiv 3 \mod 4$ and let L be as in 1 and $k = \text{poly}(d)$. For all $\epsilon \in (0,1)$, we can construct an activation σ_ϵ such that if $w_1, ..., w_k \in \mathbb{R}^d$ with w_i randomly chosen from $w_i \sim \mathcal{N}(\mathbf{0}, O(d \log d)\mathbf{I_{d \times d}})$ and $b_1, ..., b_k$ be randomly chosen at uniform from $[-1, 1]$, then with high probability, after running nodewise descent (Algorithm 3) on the objective L for at most $(d/\epsilon)^{O(d)}$ iterations, $(\boldsymbol{a}, \boldsymbol{\theta})$ is in a $(d/\epsilon)^{-O(d)}$ neighborhood of the global minima.*

F Common Activations

First, we consider the sign activation function. Under restrictions on the size of the input dimension or the number of hidden units, we can prove convergence results under the sign activation function, as it gives rise to a harmonic potential.

▶ **Assumption 1.** *All output weights $b_i = 1$ and therefore the output weights $a_i = -b_i = -1$ are fixed throughout the learning algorithm.*

▶ **Lemma 28.** *Let $\mathcal{M} = S^1$ and let Assumption 1 hold. Let L be as in (2) and σ is the sign activation function. Then L admits no strict local minima, except at the global minima.*

We cannot simply analyze the convergence of GD on all θ_i simultaneously since as before, the pairwise interaction terms between the θ_i present complications. Therefore, we now only consider the convergence guarantee of gradient descent on the first node, θ_1, to some w_j, while the other nodes are inactive (i.e. $a_2, ..., a_k = 0$). In essence, we are working with the following simplified loss function.

$$L(a_1, \theta_1) = a_1^2 \Phi(\theta_1, \theta_1) + 2 \sum_{j=1}^{k} a_1 b_j \Phi(\theta_1, w_j) \tag{3}$$

▶ **Lemma 29.** *Let $\mathcal{M} = S^1$ and L be as in (3) and σ is the sign activation function. Then, almost surely over random choices of $b_1, ..., b_k$, all local minima of L are at $\pm w_j$.*

For the polynomial activation and potential functions, we also can show convergence under orthogonality assumptions on w_j. Note that the realizability of polynomial potentials is guaranteed in Section B.

▶ **Theorem 30.** *Let $\mathcal{M} = S^{d-1}$. Let $w_1, ..., w_k$ be orthonormal vectors in $\mathbb{R}^d$ and Φ is of the form $\Phi(\theta, w) = (\theta^T w)^l$ for some fixed integer $l \geq 3$. Let L be as in (3). Then, all critical points of L are not local minima, except when $\theta_1 = w_j$ for some j.*

F.1 Convergence of Sign Activation

▶ **Lemma 31.** *Let $\mathcal{M} = S^1$ and let Assumption 1 hold. Let L be as in (2) and σ is the sign activation function. Then L admits no strict local minima, except at the global minima.*

Proof. We will first argue that unless all the electrons and protons have matched up as a permutation it cannot be a strict local minimum and then argue that the global minimum is a strict local minimum.

First note that if some electron and proton have merged, we can remove such pairs and argue about the remaining configuration of charges. So WLOG we assume there are no such overlapping electron and proton.

First consider the case when there is an isolated electron e and there is no charge diagonally opposite to it. In this case look at the two semicircles on the left and the right half of the circle around the isolated electron – let q_1 and q_2 be the net charges in the left and the right semi-circles. Note that $q_1 \neq q_2$ since they are integers and $q_1 + q_2 = +1$ which is odd. So by moving the electron slightly to the side with the larger charge you decrease the potential.

If there is a proton opposite the isolated electron the argument becomes simpler as the proton benefits the motion of the electron in either the left or right direction. So the only way the electron does not benefit by moving in either direction is that $q_1 = -1$ and $q_2 = -1$ which is impossible.

If there is an electron opposite the isolated electron then the combination of these two diagonally opposing electrons have a zero effect on every other charge. So it is possible rotate this pair jointly keeping them opposed in any way and not change the potential. So this is not a strict local minimum.

Next if there is a clump of isolated electrons with no charge on the diagonally opposite point then again as before if $q_1 \neq q_2$ we are done. If $q_1 = q_2$ then the the electrons in the clump locally are unaffected by the remaining charges. So now by splitting the clump into two groups and moving them apart infinitesimally we will decrease the potential.

Now if there is only protons in the diagonally opposite position an isolated electron again we are done as in the case when there is one electron diagonally opposite one proton.

Finally if there is only electrons diagonally opposite a clump of electrons again we are done as we have found at least one pair of opposing electrons that can be jointly rotated in any way.

Next we will argue that a permutation matching up is a strict local minumum. For this we will assume that no two protons are diagonally opposite each other (as they can be removed without affecting the function). Now given a perfect matching up of electrons and protons, if we perturb the electrons in any way infinitesimally, then any isolated clump of electrons can be moved slightly to the left or right to improve the potential. ◀

▶ **Lemma 32.** *Let $\mathcal{M} = S^1$ and L be as in* (3) *and σ is the sign activation function. Then, almost surely over random choices of $b_1, ..., b_k$, all local minima of L are at $\pm w_j$.*

Proof. In S^1, notice that the pairwise potential function is $\Phi(\theta, w) = 1 - 2\cos^{-1}(\theta^T w)/\pi = 1 - 2\alpha/\pi$, where α is the angle between θ, w. So, let us parameterize in polar coordinates, calling our true parameters as $\widetilde{w_1}, ..., \widetilde{w_k} \in [0, 2\pi]$ and rewriting our loss as a function of $\widetilde{\theta} \in [0, 2\pi]$.

Since Φ is a linear function of the angle between θ, w_j, each w_j exerts a constant gradient on $\widetilde{\theta}$ towards $\widetilde{w_j}$, with discontinuities at $\widetilde{w_j}, \pi + \widetilde{w_j}$. Almost surely over $b_1, .., b_k$, the gradient is non-zero almost everywhere, except at the discontinuities, which are at $\widetilde{w_j}, \pi + \widetilde{w_j}$ for some j. ◀

F.2 Convergence of Polynomial Potentials

▶ **Theorem 30.** *Let $\mathcal{M} = S^{d-1}$. Let $w_1, ..., w_k$ be orthonormal vectors in $\mathbb{R}^d$ and Φ is of the form $\Phi(\theta, w) = (\theta^T w)^l$ for some fixed integer $l \geq 3$. Let L be as in* (3)*. Then, all critical points of L are not local minima, except when $\theta_1 = w_j$ for some j.*

Proof. WLOG, we can consider $w_1, ..., w_d$ to be the basis vectors $e_1, ..., e_d$. Note that this is a manifold optimization problem, so our optimality conditions are given by introducing a

Lagrange multiplier λ, as in [12].

$$\frac{\partial L}{\partial a} = 2\sum_{i=1}^{d} ab_i(\theta_i)^l + 2a = 0$$

$$(\nabla_\theta L)_i = 2ab_i l(\theta_i)^{l-1} - 2\lambda\theta_i = 0$$

where λ is chosen that minimizes

$$\lambda = \arg\min_{\lambda} \sum_i (ab_i l(\theta_i)^{l-1} - \lambda\theta_i)^2 = \sum ab_i l(\theta_i)^l$$

Therefore, either $\theta_i = 0$ or $b_i(\theta_i)^{l-2} = \lambda/(al)$. From [12], we consider the constrained Hessian, which is a diagonal matrix with diagonal entry:

$$(\nabla^2 L)_{ii} = 2ab_i l(l-1)(\theta_i)^{l-2} - 2\lambda$$

Assume that there exists $\theta_i, \theta_j \neq 0$, then we claim that θ is not a local minima. First, our optimality conditions imply $b_i(\theta_i)^{l-2} = b_j(\theta_j)^{l-2} = \lambda/(al)$. So,

$$(\nabla^2 L)_{ii} = (\nabla^2 L)_{jj} = 2ab_i l(l-1)(\theta_i)^{l-2} - 2\lambda$$

$$= 2(l-2)\lambda = -2(l-2)la^2$$

Now, there must exist a vector $v \in S^{d-1}$ such that $v_k = 0$ for $k \neq i, j$ and $v^T\theta = 0$, so v is in the tangent space at θ. Finally, $v^T(\nabla^2 L)v = -2(l-2)la^2 < 0$, implying θ is not a local minima when $a \neq 0$. Note that $a = 0$ occurs with probability 0 since our objective function is non-increasing throughout the gradient descent algorithm and is almost surely initialized to be negative with a optimized upon initialization, as by observed before. ◀

Under a node-wise descent algorithm, we can show polynomial-time convergence to global minima under orthogonality assumptions on w_j for these polynomial activations/potentials. We will not include the proof but it follows from similar techniques presented for nodewise convergence in Section E.

Accelerated Extra-Gradient Descent: A Novel Accelerated First-Order Method*

Jelena Diakonikolas[1] and Lorenzo Orecchia[2]

1 **Department of Computer Science, Boston University, Boston MA 02215, USA**
jelenad@bu.edu

2 **Department of Computer Science, Boston University, Boston MA 02215, USA**
orecchia@bu.edu

Abstract

We provide a novel accelerated first-order method that achieves the asymptotically optimal convergence rate for smooth functions in the first-order oracle model. To this day, Nesterov's Accelerated Gradient Descent (AGD) and variations thereof were the only methods achieving acceleration in this standard blackbox model. In contrast, our algorithm is significantly different from AGD, as it relies on a *predictor-corrector approach* similar to that used by Mirror-Prox [18] and Extra-Gradient Descent [14] in the solution of convex-concave saddle point problems. For this reason, we dub our algorithm Accelerated Extra-Gradient Descent (AXGD).

Its construction is motivated by the discretization of an accelerated continuous-time dynamics [15] using the classical method of implicit Euler discretization. Our analysis explicitly shows the effects of discretization through a conceptually novel primal-dual viewpoint. Moreover, we show that the method is quite general: it attains optimal convergence rates for other classes of objectives (e.g., those with generalized smoothness properties or that are non-smooth and Lipschitz-continuous) using the appropriate choices of step lengths. Finally, we present experiments showing that our algorithm matches the performance of Nesterov's method, while appearing more robust to noise in some cases.

1998 ACM Subject Classification F.2.1 Theory of Computation: Analysis of Algorithms and Problem Complexity, Numerical Algorithms and Problems, G.1.6 Mathematics of Computing: Numerical Analysi, Optimization, Convex programming, Gradient methods

Keywords and phrases Acceleration, dynamical systems, discretization, first-order methods

Digital Object Identifier 10.4230/LIPIcs.ITCS.2018.23

1 Introduction

First-order methods for convex optimization have come to play an important role in the design of algorithms and in Theoretical Computer Science in general, with applications including numerical methods [30, 13], graph algorithms [12, 29], submodular optimization [8] and complexity theory [11].

A classical setting for convex optimization is that of smooth optimization, i.e., minimizing a convex differentiable function f over a convex set $X \subseteq \mathbb{R}^n$, with the smoothness assumption

* Part of this work was done while the authors were visiting the Simons Institute for the Theory of Computing. It was partially supported by NSF grant #CCF-1718342 and by the DIMACS/Simons Collaboration on Bridging Continuous and Discrete Optimization through NSF grant #CCF-1740425.

that the gradient of f be L-Lipschitz continuous[1] for some positive real L, i.e.:

$$\forall\, x, y \in X,\ \|\nabla f(x) - \nabla f(y)\|_* \leq L \cdot \|x - y\|.$$

In this setting, it is also assumed that the algorithm can access the input function f only via queries to a first-order oracle, i.e., a *blackbox* that on input $x \in X$, returns the vector $\nabla f(x)$ in constant time.[2]

Smooth optimization is of particular interest because it is the simplest setting in which the phenomenon of *acceleration* arises, i.e., the optimal algorithms in the blackbox model achieve an error that scales as $O(1/t^2)$, where t is the number of queries [22]. This should be compared to the convergence of steepest-descent methods, which attempt to locally minimize the first-order approximation to the function and only yield $O(1/t)$-convergence [3, 28]. Acceleration has proved an active topic of algorithmic research, both for the promise of obtaining generic speed-ups for problems having some smoothness condition and for the unintuitive nature of the fact that faster algorithms can be obtained by not moving in the direction of steepest-descent.

Recently, a number of papers have helped demystify the concept behind accelerated algorithms by providing interpretations based on continuous dynamics and their discretization [15, 33, 31], geometric ideas [5], and on trading off the performances of two slower first-order methods [1]. Despite these efforts, to this day, Nesterov's Accelerated Gradient Descent (AGD) methods remain the only paradigm [22, 23] through which to obtain accelerated algorithms in the blackbox model and in related settings, where all existing accelerated algorithms are variations of Nesterov's general method [32].

Our Main Contributions

We present a novel accelerated first-order method that achieves the optimal convergence rate for smooth functions and is significantly different from Nesterov's method, as it relies on a predictor-corrector approach, similar to that of Mirror-Prox [18] and Extra-Gradient Descent [14]. For this reason, we name our method *Accelerated Extra-Gradient Descent* (AXGD). Our derivation of the AXGD algorithm is based on the discretization of a recently proposed continuous-time accelerated algorithm [15, 33]. The continuous-time view is particularly helpful in clarifying the relation between AGD, AXGD, and Mirror-Prox. Following [15], given a gradient field ∇f and a prox function ψ, it is possible to define two continuous-time evolutions: the mirror-descent dynamics and the accelerated-mirror-descent dynamics (see Section 2.2). With this setup, Nesterov's AGD can be seen as a variant of the classical forward-Euler discretization applied to the accelerated-mirror-descent dynamics. In contrast, Mirror-Prox and extra-gradient methods arise from an approximate backward-Euler discretization [9] on the mirror-descent dynamics. Finally, our algorithm AXGD is the result of an approximate backward-Euler discretization of the accelerated mirror-descent dynamics.

Another conceptual contribution of our paper is the application of a primal-dual viewpoint on the convergence of first-order methods, both in continuous and discrete time. At every time instant t, our algorithm explicitly maintains a current primal solution $\mathbf{x}^{(t)}$ and a current dual solution $\mathbf{z}^{(t)}$, the latter in the form of a convex combination of gradients of the convex

[1] Lipschitz continuity is defined w.r.t to a pair of dual norms $\|\cdot\|, \|\cdot\|_*$. At a first reading, these can be taken as $\|\cdot\|_2$.

[2] In general, we may assume that the blackbox also returns the function value $f(x)$. However, for the general class of problems we consider this information is not necessary and the gradient suffices [28]. For intuition about this, see the expression for the change in duality gap in Equation 4.

objective, i.e., a lower-bounding hyperplane. This primal-dual pair of solutions yields, for every t, both an upper bound U_t and a lower bound L_t on the optimum: $U_t \geq f(\mathbf{x}^*) \geq L_t$. In all cases, we obtain convergence bounds by explicitly quantifying the rate at which the duality gap $G_t = U_t - L_t$ goes to zero. We believe that this primal-dual viewpoint makes the analysis and design of first-order methods easier to carry out. We provide its application to proving other classical results in first-order methods, including Mirror Descent, Mirror-Prox, and Frank-Wolfe algorithms in the upcoming manuscript [7].

Other Technical Contributions

In Section 2.6, we provide a unified convergence proof for standard smooth functions (as defined above) and for functions with Hölder-continuous gradients, a more general notion of smoothness [20]. While this paper focuses on the standard smooth setup, the same techniques easily yield results matching those of AGD methods for the strongly-convex-and-smooth case. Indeed, it is possible to prove that our method is universal, in the sense of Nesterov [21], meaning that it can be composed with a line-search algorithm to yield near-optimal algorithms even when the smoothness parameters of the functions are unknown. We illustrate this phenomenon by showing that (AXGD) also achieves the optimal rate for the optimization of Lipschitz-continuous convex functions, a non-smooth problem.

Finally, we present a suite of experiments comparing AGD, AXGD, and standard gradient methods, showing that the performance of AXGD closely matches that of AGD methods. We also explore the empirical performance of AXGD in the practically and theoretically relevant case in which the queried gradients are corrupted by noise. We show that AXGD exhibits better stability properties than AGD in some cases, leading to a number of interesting theoretical questions on the convergence of AXGD.

1.1 Related Work

In his seminal work [22, 23], Nesterov gave a method for the minimization of convex functions that are smooth with respect to the Euclidean norm, where the function is accessed through a first-order oracle. Nesterov's method converges quadratically faster than gradient descent, at a rate of $O(\frac{1}{t^2})$, which has been shown to be asymptotically optimal [23] for smooth functions in this standard blackbox model [28]. More recently, Nesterov generalized this method to allow non-Euclidean norms in the definition of smoothness [25]. We refer to this generalization of Nesterov's method and to instantiations thereof as AGD methods. Accelerated gradient methods have been widely extended and modified for different settings, including composite optimization [27, 16], cubic regularization [26], and universal methods [21]. They have also found a number of fundamental applications in many algorithmic areas, including machine learning (see [4]) and discrete optimization [17].

An important application of AGD methods concerns the solution of various convex-concave saddle point problems. While these are examples of non-smooth problems, for which the optimal rate is known to be $\Omega(\frac{1}{\sqrt{k}})$ [20], Nesterov showed that the saddle-point structure can be exploited by smoothing the original problem and applying AGD methods on the resulting smooth function [25]. This approach [25, 24] yields an $O(\frac{1}{k})$-convergence for convex-concave saddle point problems with smooth gradients. Surprisingly, at around the same time, Nemirovski [18] gave a very different algorithm, known as Mirror-Prox, which achieves the same complexity for the saddle point problem. Mirror-Prox does not rely on the algorithm or analysis underlying AGD, but is based instead on the idea of an *extra-gradient* step, i.e., a correction step that is performed at every iteration to speed up convergence.

Mirror-Prox can be viewed as an approximate solution to the implicit Euler discretization of the standard mirror descent dynamics of Nemirovski and Yudin [20]. In this fashion, our AXGD algorithm resembles Mirror-Prox as it also makes use of an approximate implicit Euler step to discretize a *different*, accelerated dynamic.

A number of interpretations have been proposed to explain the phenomenon of acceleration in first-order methods. Tseng gives a formal framework that unifies all the different instantiations of AGD methods [32]. More recently, Allen-Zhu and Orecchia [1] cast AGD methods as the result of coupling mirror descent and gradient descent steps. Bubeck *et al.* give an elegant geometric interpretation of the Euclidean instantiation of Nesterov's method [5]. At the same time, Su *et al.* [31], Krichene *et al.* [15], and Wibisono *et al.* [33] have provided characterizations of accelerated methods as discretizations of certain families of ODEs related to the gradient flow of the objective f. Our algorithm is strongly influenced by these works: in particular, the starting point for the derivation of AXGD is the continuous-time accelerated-mirror-descent (AMD) dynamics [15].

1.2 Preliminaries

We focus on continuous and differentiable functions defined on a closed convex set $X \subseteq \mathbb{R}^n$. We assume that there is an arbitrary (but fixed) norm $\|\cdot\|$ associated with the space, and all the statements about function properties are stated with respect to that norm. We also define the dual norm $\|\cdot\|_*$ in the standard way: $\|\mathbf{z}\|_* = \sup\{\langle \mathbf{z}, \mathbf{x}\rangle : \|\mathbf{x}\| = 1\}$. The following definitions will be useful in our analysis, and thus we state them here for completeness.

▶ **Definition 1.** A function $f : X \to \mathbb{R}$ is convex on X, if for all $\mathbf{x}, \hat{\mathbf{x}} \in X$: $f(\hat{\mathbf{x}}) \geq f(\mathbf{x}) + \langle \nabla f(\mathbf{x}), \hat{\mathbf{x}} - \mathbf{x}\rangle$.

▶ **Definition 2.** A function $f : X \to \mathbb{R}$ is smooth on X with smoothness parameter L and with respect to a norm $\|\cdot\|$, if for all $\mathbf{x}, \hat{\mathbf{x}} \in X$: $f(\hat{\mathbf{x}}) \leq f(\mathbf{x}) + \langle \nabla f(\mathbf{x}), \hat{\mathbf{x}} - \mathbf{x}\rangle + \frac{L}{2}\|\hat{\mathbf{x}} - \mathbf{x}\|^2$.

Definition 2 can equivalently be stated as: $\|\nabla f(\mathbf{x}) - \nabla f(\hat{\mathbf{x}})\|_* \leq L\|\mathbf{x} - \hat{\mathbf{x}}\|$.

▶ **Definition 3.** A function $f : X \to \mathbb{R}$ is strongly convex on X with strong convexity parameter σ and with respect to a norm $\|\cdot\|$, if for all $\mathbf{x}, \hat{\mathbf{x}} \in X$: $f(\hat{\mathbf{x}}) \geq f(\mathbf{x}) + \langle \nabla f(\mathbf{x}), \hat{\mathbf{x}} - \mathbf{x}\rangle + \frac{\sigma}{2}\|\hat{\mathbf{x}} - \mathbf{x}\|^2$.

▶ **Definition 4.** (Bregman Divergence) $D_\psi(\mathbf{x}, \hat{\mathbf{x}}) \stackrel{\text{def}}{=} \psi(\mathbf{x}) - \psi(\hat{\mathbf{x}}) - \langle \nabla\psi(\hat{\mathbf{x}}), \mathbf{x} - \hat{\mathbf{x}}\rangle$.

▶ **Definition 5.** (Convex Conjugate) Function ψ^* is the convex conjugate of $\psi : X \to \mathbb{R}$, if $\psi^*(\mathbf{z}) = \max_{\mathbf{x}\in X}\{\langle \mathbf{z}, \mathbf{x}\rangle - \psi(\mathbf{x})\}$, $\forall \mathbf{z} \in \mathbb{R}$.

In the rest of the paper, we will assume that $\psi(\mathbf{x})$ is continuously differentiable, so that Fenchel-Moreau Theorem implies that $\psi^{**} = \psi$.[3] We are interested in minimizing a convex function f over $X \subseteq \mathbb{R}^n$. We let $\mathbf{x}^* = \arg\min_{\mathbf{x}\in X} f(\mathbf{x})$.

We will refer to any step that decreases the value of f as a gradient descent step. In the special case of a smooth function f the gradient descent step from a point $\mathbf{x} \in X$ will be given as $\text{Grad}(\mathbf{x}) = \arg\min_{\hat{\mathbf{x}}\in X}\{f(\mathbf{x}) + \langle \nabla f(\mathbf{x}), \hat{\mathbf{x}} - \mathbf{x}\rangle + \frac{L}{2}\|\hat{\mathbf{x}} - \mathbf{x}\|^2\}$.

We will assume that there is a strongly-convex differentiable function $\psi : X \to \mathbb{R}$ such that $\max_{\mathbf{x}\in X}\{\langle \mathbf{z}, \mathbf{x}\rangle - \psi(\mathbf{x})\}$ is easily solvable, possibly in a closed form. Notice that this

[3] Note that Fenchel-Moreau Theorem requires ψ to only be lower-semicontinuous for $\psi^{**} = \psi$ to hold, which is a weaker property than continuity or continuous differentiability.

problem defines the convex conjugate of $\psi(\cdot)$, i.e., $\psi^*(\mathbf{z}) = \max_{\mathbf{x}\in X}\{\langle \mathbf{z}, \mathbf{x}\rangle - \psi(\mathbf{x})\}$. The following standard fact will be extremely useful in carrying out the analysis of the algorithms in this paper.

▶ **Fact 6.** Let $\psi : X \to \mathbb{R}$ be a differentiable strongly-convex function. Then:

$$\nabla\psi^*(\mathbf{z}) = \arg\max_{\mathbf{x}\in X}\{\langle \mathbf{z}, \mathbf{x}\rangle - \psi(\mathbf{x})\}.$$

Additional useful properties of Bregman divergence are provided in Appendix A.

2 Accelerated Extra-Gradient Descent

In this section, we describe the AXGD method and analyze its convergence. For comparison, steps of AGD and AXGD are shown next to each other in the box below. In continuous time, both algorithms follow the same dynamics. However, due to the different discretization methods used in constructing AGD and AXGD, they follow different discrete-time updates. In particular, we show in [7] that AGD can be interpreted as performing explicit (forward) Euler discretization plus a gradient step to correct the discretization error. In contrast, AXGD uses an approximate implementation of implicit (backward) Euler discretization to directly control the discretization error.

Accelerated Gradient Descent (AGD)

$$\begin{aligned}\mathbf{x}^{(k+1)} &= \frac{A_k}{A_{k+1}}\hat{\mathbf{x}}^{(k)} + \frac{a_{k+1}}{A_{k+1}}\nabla\psi^*(\mathbf{z}^{(k)}),\\ \mathbf{z}^{(k+1)} &= \mathbf{z}^{(k)} - a_{k+1}\nabla f(\mathbf{x}^{(k+1)}),\\ \hat{\mathbf{x}}^{(k+1)} &= \mathrm{Grad}(\mathbf{x}^{(k+1)}).\end{aligned} \tag{1}$$

Accelerated Extra-Gradient Descent (AXGD)

$$\begin{aligned}\hat{\mathbf{x}}^{(k)} &= \frac{A_k}{A_{k+1}}\mathbf{x}^{(k)} + \frac{a_{k+1}}{A_{k+1}}\nabla\psi^*(\mathbf{z}^{(k)}),\\ \hat{\mathbf{z}}^{(k)} &= \mathbf{z}^{(k)} - a_{k+1}\nabla f(\hat{\mathbf{x}}^{(k)}),\\ \mathbf{x}^{(k+1)} &= \frac{A_k}{A_{k+1}}\mathbf{x}^{(k)} + \frac{a_{k+1}}{A_{k+1}}\nabla\psi^*(\hat{\mathbf{z}}^{(k)}),\\ \mathbf{z}^{(k+1)} &= \mathbf{z}^{(k)} - a_{k+1}\nabla f(\mathbf{x}^{(k+1)}).\end{aligned} \tag{2}$$

The idea behind AXGD is similar to the dual-averaging version of Nemirovski's mirror prox algorithm [18, 7], with the main difference coming from the discretization of the accelerated dynamics in Equation (5) (as opposed to the standard mirror descent dynamics used in [18]). As we will show, an exact implicit Euler step would have $\nabla\psi^*(\mathbf{z}^{(k+1)})$ instead of $\nabla\psi^*(\hat{\mathbf{z}}^{(k)})$ in the third line of AXGD. However, obtaining $\mathbf{x}^{(k+1)}$ in a such a manner could be computationally prohibitive since $\mathbf{z}^{(k+1)}$ implicitly depends on $\mathbf{x}^{(k+1)}$ through its gradient. Instead, we opt for an extra prox-step $\nabla\psi^*(\hat{\mathbf{z}}^{(k)})$ that adds the gradient at an intermediate point $\hat{\mathbf{x}}^{(k)}$ constructed using $\mathbf{x}^{(k)}$ and $\mathbf{z}^{(k)}$ from the previous iteration. Thanks to this extra-gradient step, AXGD can correct the discretization error without using a gradient step.

Convergence proof for AXGD together with the sufficient conditions for obtaining optimal convergence bounds are provided in Section 2.4. For example, Theorem 11 shows that when the objective function is smooth, AXGD converges at the optimal rate of $1/k^2$. The analysis of AGD is provided in [7].

2.1 Approximate Optimality Gap

The analysis relies on the construction of an approximate optimality gap G_t, which is defined as the difference of an upper bound U_t and a lower bound L_t to the optimal function value $f(\mathbf{x}^*)$. In particular, for an increasing function of time t, $\alpha^{(t)}$, the convergence analysis will work on establishing the following:

Invariance condition: $\alpha^{(t)}G_t$ is non-increasing with time t.

Such a condition immediately implies: $G_t \leq \frac{\alpha^{(t_0)}}{\alpha^{(t)}} G_{t_0}$, leading to the $\frac{1}{\alpha^{(t)}}$ convergence rate. We sketch the main ideas that relate to the accelerated methods and AXGD in particular here for completeness, while the more general arguments that recover a number of known first-order methods are provided in [7].

We now describe the upper bound and the lower bound choices, which will take the same form in both continuous time and discrete time domains. To do so, we will rely on the Lebesgue-Stieltjes integration, which allows us to treat continuous and discrete choice of $\alpha^{(t)}$ in a unified manner. Observe that when $\alpha^{(t)}$ is a discrete measure, $\dot{\alpha}^{(t)}$ is a train of (scaled) Dirac Delta functions. Denote $A^{(t)} = \int_{t_0}^{t} d\alpha^{(\tau)} = \int_{t_0}^{t} \dot{\alpha}^{(\tau)} d\tau$.

Upper Bound

As $\mathbf{x}^* \in X$ is the minimizer of $f(\cdot)$, $f(\mathbf{x})$ for any $\mathbf{x} \in X$ constitutes a valid upper bound. In particular, our choice of the upper bound will be $U_t = f(\mathbf{x}^{(t)})$, where $\mathbf{x}^{(t)}$ is the solution maintained by the algorithm at time t.

Lower Bound

More interesting than the upper bound is the construction of a lower bound to $f(\mathbf{x}^*)$. From convexity of f, we have the standard lower-bounding hyperplanes $\forall \mathbf{x}, \hat{\mathbf{x}} \in X$: $f(\mathbf{x}) \geq f(\hat{\mathbf{x}}) + \langle \nabla f(\hat{\mathbf{x}}), \mathbf{x} - \hat{\mathbf{x}} \rangle$. A natural choice of a lower bound to the optimum at time $t \geq t_0$, is obtained by averaging such hyperplanes over $[t_0, t]$ according to the measure α:

$$f(\mathbf{u}) \geq \frac{\int_{t_0}^{t} f(\mathbf{x}^{(\tau)}) d\alpha^{(\tau)}}{A^{(t)}} + \frac{\int_{t_0}^{t} \left\langle \nabla f(\mathbf{x}^{(\tau)}), \mathbf{u} - \mathbf{x}^{(\tau)} \right\rangle d\alpha^{(\tau)}}{A^{(t)}}, \ \forall \mathbf{u} \in X.$$

While we could take the minimum over $\mathbf{u} \in X$ on the right-hand side of this equation as our notion of lower bound, this choice has two serious drawbacks. First, it is non-smooth, and in general not even differentiable, as a function of t. Second, in continuous-time, it is not defined for our initial time t_0, meaning that we do not have a natural concept of initial lower bound and initial duality gap. (In the discrete time, we can ensure that α contains a Dirac Delta function at t_0, which overcomes this issue.) We address the first problem by applying regularization, i.e., by adding to both sides of the inequality a regularizer term that is strongly-convex in $\mathbf{x}$ and then minimizing the right-hand side with respect to $\mathbf{u} \in X$.[4] Without loss of generality, the regularizer can be taken to be the Bregman divergence of a σ-strongly convex function ψ taken from an input point $\mathbf{x}^{(t_0)}$. This yields:

$$f(\mathbf{x}^*) + \frac{D_\psi(\mathbf{x}^*, \mathbf{x}^{(t_0)})}{A^{(t)}} \geq \frac{\int_{t_0}^{t} f(\mathbf{x}^{(\tau)}) d\alpha^{(\tau)}}{A^{(t)}} + \frac{\min_{\mathbf{u} \in X} \left\{ \int_{t_0}^{t} \left\langle \nabla f(\mathbf{x}^{(\tau)}), \mathbf{u} - \mathbf{x}^{(\tau)} \right\rangle d\alpha^{(\tau)} + D_\psi(\mathbf{u}, \mathbf{x}^{(t_0)}) \right\}}{A^{(t)}}.$$

To address the second problem, we mix into the α-combination of hyperplanes the optimal lower bound $f(\mathbf{x}^*)$ with weight $\alpha^{(t)} - A^{(t)}$ (which is just zero in the discrete time, as in that case $A^{(t)} = \alpha^{(t)}$). Rescaling the normalization factor, we obtain our notion of *regularized*

[4] This is similar to the well-known Moreau-Yosida regularization.

lower bound:

$$L_t \stackrel{\text{def}}{=} \frac{\int_{t_0}^{t} f(\mathbf{x}^{(\tau)}) d\alpha^{(\tau)}}{\alpha^{(t)}} + \frac{\min_{\mathbf{u}\in X} \left\{ \int_{t_0}^{t} \left\langle \nabla f(\mathbf{x}^{(\tau)}), \mathbf{u} - \mathbf{x}^{(\tau)} \right\rangle d\alpha^{(\tau)} + D_\psi(\mathbf{u}, \mathbf{x}^{(t_0)}) \right\}}{\alpha^{(t)}} + \frac{(\alpha^{(t)} - A^{(t)}) f(\mathbf{x}^*) - D_\psi(\mathbf{x}^*, \mathbf{x}^{(t_0)})}{\alpha^{(t)}}. \tag{3}$$

2.2 Accelerated Mirror Descent in Continuous Time

We now show that the accelerated dynamics can be obtained by enforcing the invariance condition from previous subsection with $\alpha^{(t)} G_t$ being constant; i.e., we enforce that $\frac{d}{dt}(\alpha^{(t)} G_t) = 0$. Towards that goal, assume that $\alpha^{(t)}$ is continuously differentiable, and observe that $\alpha^{(t)} - A^{(t)} = \alpha^{(t_0)}$ is constant. To simplify the notation when taking the time derivative of $\alpha^{(t)} G^{(t)}$, we first show the following:

▶ **Proposition 7.** Let $\mathbf{z}^{(t)} = \nabla\psi(\mathbf{x}^{(t_0)}) - \int_{t_0}^{t} \nabla f(\mathbf{x}^{(\tau)}) d\alpha^{(\tau)}$. Then:

$$\nabla\psi^*(\mathbf{z}^{(t)}) = \arg\min_{\mathbf{u}\in X} \left\{ \int_{t_0}^{t} \left\langle \nabla f(\mathbf{x}^{(\tau)}), \mathbf{u} - \mathbf{x}^{(\tau)} \right\rangle d\alpha^{(\tau)} + D_\psi(\mathbf{u}, \mathbf{x}^{(t_0)}) \right\}.$$

I.e., $\nabla\psi^*(\mathbf{z}^{(t)})$ is the argument of the minimum appearing in the definition of lower bound L_t. The proof is simple and is provided in the appendix.

Recalling that $U_t = f(\mathbf{x}^{(t)})$ and using (3) and Danskin's theorem (which allows us to differentiate inside the min):

$$\begin{aligned} \frac{d}{dt}(\alpha^{(t)} G_t) &= \frac{d}{dt}(\alpha^{(t)} f(\mathbf{x}^{(t)})) - \dot\alpha^{(t)} f(\mathbf{x}^{(t)}) - \dot\alpha^{(t)} \left\langle \nabla f(\mathbf{x}^{(t)}), \nabla\psi^*(\mathbf{z}^{(t)}) - \mathbf{x}^{(t)} \right\rangle \\ &= \left\langle \nabla f(\mathbf{x}^{(t)}), \alpha^{(t)} \mathbf{x}^{(t)} - \dot\alpha^{(t)} \left(\nabla\psi^*(\mathbf{z}^{(t)}) - \mathbf{x}^{(t)} \right) \right\rangle. \end{aligned} \tag{4}$$

Hence, to obtain $\frac{d}{dt}(\alpha^{(t)} G_t) = 0$, it suffices to set $\alpha^{(t)} \mathbf{x}^{(t)} = \dot\alpha^{(t)} (\nabla\psi^*(\mathbf{z}^{(t)}) - \mathbf{x}^{(t)})$, resulting in the accelerated dynamics from [15]:

$$\begin{aligned} \dot{\mathbf{z}}^{(t)} &= -\dot\alpha^{(t)} \nabla f(\mathbf{x}^{(t)}), \\ \dot{\mathbf{x}}^{(t)} &= \dot\alpha^{(t)} \frac{\nabla\psi^*(\mathbf{z}^{(t)}) - \mathbf{x}^{(t)}}{\alpha^{(t)}}, \end{aligned} \tag{5}$$

$\mathbf{z}^{(t_0)} = \nabla\psi(\mathbf{x}^{(t_0)})$, $\mathbf{x}^{(t_0)} \in X$ is an arbitrary initial point.

It is not hard to see that (5) constructs a sequence of points $\mathbf{x}^{(t)}$ that are feasible, that is, $\mathbf{x}^{(t)} \in X$. This is because $\mathbf{x}^{(t)}$ can equivalently be written as $\frac{d}{dt}(\alpha^{(t)} \mathbf{x}^{(t)}) = \dot\alpha^{(t)} \nabla\psi^*(\mathbf{z}^{(t)})$, which, after integrating over $\tau \in [t_0, t]$, gives $\mathbf{x}^{(t)} = \frac{\alpha^{(t_0)}}{\alpha^{(t)}} \mathbf{x}^{(t_0)} + \frac{1}{\alpha^{(t)}} \int_{t_0}^{t} \nabla\psi^*(\mathbf{z}^{(\tau)}) d\alpha^{(\tau)}$ – a convex combination of $\mathbf{x}^{(t_0)}$ and $\nabla\psi^*(\mathbf{z}^{(\tau)})$ for $\tau \in [t_0, t]$. By (5), $\mathbf{x}^{(t_0)} \in X$, while $\nabla\psi^*(\mathbf{z}^{(\tau)}) \in X$ by Proposition 7.

We immediately obtain the following continuous-time convergence guarantee:

▶ **Lemma 8.** *Let $\mathbf{x}^{(t)}$ evolve according to (5). Then, $\forall t \geq t_0$:*

$$f(\mathbf{x}^{(t)}) - f(\mathbf{x}^*) \leq \frac{\alpha^{(t_0)}(f(\mathbf{x}^{(t_0)}) - f(\mathbf{x}^*)) + D_\psi(\mathbf{x}^*, \mathbf{x}^{(t_0)})}{\alpha^{(t)}}.$$

Proof. We have already established that $\frac{d}{dt}(\alpha^{(t)} G^{(t)}) = 0$, and, therefore, $f(\mathbf{x}^{(t)}) - f(\mathbf{x}^*) \leq G_t = \frac{\alpha^{(t_0)}}{\alpha^{(t)}} G_{t_0}$. Observing that $G_{t_0} = f(\mathbf{x}^{(t_0)}) - f(\mathbf{x}^*) + D_\psi(\mathbf{x}^*, \mathbf{x}^{(t_0)})/\alpha^{(t_0)}$, the proof follows. ◀

2.3 Discretization

As discussed in Section 2.1, our construction of the approximate optimality gap is valid both in the continuous time and in the discrete time domain. To understand where the discretization error occurs, we make the following observations. First, the upper bound does not involve any integration, and thus cannot incur a discretization error. In the lower bound (3), the role of the first integral is only to perform weighted averaging, which is the same in the continuous time and in the discrete time, and, therefore, does not incur a discretization error. The terms that are not integrated over look the same whether or not $\alpha^{(t)}$ is discrete. Therefore, the only term that can incur the discretization error is the integral under the min: $I^{(t_0,t)} = \int_{t_0}^{t} \left\langle \nabla f(\mathbf{x}^{(\tau)}), \nabla \psi^*(\mathbf{z}^{(t)}) - \mathbf{x}^{(\tau)} \right\rangle d\alpha^{(\tau)}$.

As mentioned before, when α is a discrete measure, we can express it as $\alpha^{(t)} = \sum_{i=1}^{\infty} a_i \delta(t - (t_0 + i - 1))$, where $\delta(\cdot)$ denotes the Dirac Delta function and a_i's are positive. Then $A^{(t)} = \int_{t_0}^{t} d\alpha^{(\tau)} = \sum_{i:t_0+i-1\le t} a_i$. To simplify the notation, we will use $i \in \mathbb{Z}_+$ to denote the discrete time points corresponding to $t_0 + i - 1$ on the continuous line. Therefore, the discretization error incurred in $A^{(t)}L_t$ between the discrete time points i and $i+1$ (understood as integrating from i^+ to $(i+1)^+$) is $I^{(i,i+1)} - I_c^{(i,i+1)}$, where $I_c^{(i,i+1)}$ is the continuous approximation of $I^{(i,i+1)}$ (i.e., we allow continuous integration rules in $I_c^{(i,i+1)}$). We can now establish the following bound on the discretization error.

▶ **Lemma 9.** *Let $A_{i+1}G_{i+1} - A_iG_i \equiv E_{i+1}$ be the discretization error. Then*

$$G_k = \frac{A_1}{A_k} G_1 + \frac{\sum_{i=1}^{k} E_i}{A_k}$$

and

$$E_{i+1} \le \left\langle \nabla f(\mathbf{x}^{(i+1)}), A^{(i+1)}\mathbf{x}^{(i+1)} - A^{(i)}\mathbf{x}^{(i)} - a_{i+1}\nabla\psi^*(\mathbf{z}^{(i+1)}) \right\rangle - D_{\psi^*}(\mathbf{z}^{(i)}, \mathbf{z}^{(i+1)}).$$

Proof. The first part of the lemma follows by summing over $1 \le i \le k$. For the second part, we have already argued that $E_{i+1} = I_c^{(i,i+1)} - I^{(i,i+1)}$. For the discrete integral $I^{(i,i+1)}$, as $\dot\alpha^{(t)}$ just samples the function under the integral at point $i+1$, we have:

$$I^{(i,i+1)} = a_{i+1} \left\langle \nabla f(\mathbf{x}^{(i+1)}), \nabla\psi^*(\mathbf{z}^{(i+1)}) - \mathbf{x}^{(i+1)} \right\rangle. \tag{6}$$

For the continuous integral, using (5) and integration by parts:

$$\begin{aligned} I_c^{(i,i+1)} &= \int_i^{i+1} \alpha^{(\tau)} \left\langle \nabla f(\mathbf{x}^{(\tau)}), \dot{\mathbf{x}}^{(\tau)} \right\rangle d\tau \\ &\quad + \int_i^{i+1} \left\langle \nabla f(\mathbf{x}^{(\tau)}), \nabla\psi^*(\mathbf{z}^{(i+1)}) - \nabla\psi^*(\mathbf{z}^{(\tau)}) \right\rangle d\alpha^{(\tau)} \\ &= A^{(i)}(f(\mathbf{x}^{(i+1)}) - f(\mathbf{x}^{(i)})) - \int_i^{i+1} \left\langle \dot{\mathbf{z}}^{(\tau)}, \nabla\psi^*(\mathbf{z}^{(i+1)}) - \nabla\psi^*(\mathbf{z}^{(\tau)}) \right\rangle d\tau \\ &= A^{(i)}(f(\mathbf{x}^{(i+1)}) - f(\mathbf{x}^{(i)})) - D_{\psi^*}(\mathbf{z}^{(i)}, \mathbf{z}^{(i+1)}), \end{aligned} \tag{7}$$

where we have used $\dot{\mathbf{z}}^{(\tau)} = -\dot\alpha^{(\tau)}\nabla f(\mathbf{x}^{(\tau)})$, $\nabla_{\mathbf{z}^{(\tau)}} D_{\psi^*}(\mathbf{z}^{(\tau)}, \mathbf{z}^{(i+1)}) = \nabla\psi^*(\mathbf{z}^{(\tau)}) - \nabla\psi^*(\mathbf{z}^{(i+1)})$, and $D_{\psi^*}(\mathbf{z}^{(i)}, \mathbf{z}^{(i)}) = 0$.

By convexity of f, $f(\mathbf{x}^{(i+1)}) - f(\mathbf{x}^{(i)}) \le \left\langle \nabla f(\mathbf{x}^{(i+1)}), \mathbf{x}^{(i+1)} - \mathbf{x}^{(i)} \right\rangle$. Combining with (6) and (7):

$$E_{i+1} \le \left\langle \nabla f(\mathbf{x}^{(i+1)}), A^{(i+1)}\mathbf{x}^{(i+1)} - A^{(i)}\mathbf{x}^{(i)} - a_{i+1}\nabla\psi^*(\mathbf{z}^{(i+1)}) \right\rangle - D_{\psi^*}(\mathbf{z}^{(i)}, \mathbf{z}^{(i+1)}),$$

as claimed. ◀

We remark that the same result for the discretization error can be obtained by directly computing $A_{i+1}G_{i+1} - A_iG_i$ under a discrete measure α (where all the integrals in the definition of the duality gap are replaced by summations). We have chosen to work with the integration error described above to demonstrate the cause of the discretization error.

We now describe how AXGD cancels out the discretization error by (approximately) implementing implicit Euler discretization of $\dot{\mathbf{x}}^{(t)}$.

Implicit Euler Discretization

Implicit Euler discretization is an abstract discretization method which defines the next iterate $\mathbf{x}^{(k+1)}$ implicitly as a function of the gradient at $\mathbf{x}^{(k+1)}$. In the case of the AMD dynamics, implicit Euler discretization yields the following algorithm: let $\mathbf{x}^{(1)} \in X$ be an arbitrary initial point that satisfies $\mathbf{x}^{(1)} = \nabla\psi^*(\mathbf{z}^{(1)})$, where $\mathbf{z}^{(1)} = \nabla\psi(\mathbf{x}^{(1)}) - \nabla f(\mathbf{x}^{(1)})$; for all $k \geq 1$

$$\begin{cases} \mathbf{z}^{(k+1)} = \mathbf{z}^{(k)} - a_{k+1}\nabla f(\mathbf{x}^{(k+1)}), \\ \mathbf{x}^{(k+1)} = \frac{A_k}{A_{k+1}}\mathbf{x}^{(k)} + \frac{a_{k+1}}{A_{k+1}}\nabla\psi^*(\mathbf{z}^{(k+1)}) \end{cases} \tag{8}$$

Observe that $\mathbf{x}^{(k+1)}$ in (8) exactly sets the inner product in E_{i+1} (Lemma 9) to zero, leaving only the negative term $-D_{\psi^*}(\mathbf{z}^{(i)}, \mathbf{z}^{(i+1)})$. While this discretization is not computationally feasible in practice, as it requires solving for the implicitly defined $\mathbf{x}^{(k+1)}$, it also boasts a negative discretization error, i.e., it converges faster than the continuous-time AMD. Ultimately, we will use this extra slack to trade-off the error arising from an *approximate* implicit discretization.

2.4 Convergence of AXGD

A standard way to implement implicit Euler discretization in the solution of ODEs [9] is to replace the exact solution of the implicit equation with a small number of fixed point iterations of the same equation. In our case, the implicit equation can be written as:

$$\mathbf{x}^{(k+1)} = \frac{A_k}{A_{k+1}}\mathbf{x}^{(k)} + \frac{a_{k+1}}{A_{k+1}}\nabla\psi^*(\mathbf{z}^{(k)} - a_{k+1}\nabla f(\mathbf{x}^{(k+1)})).$$

Two steps of the fixed-point iteration yield the following updates, which are exactly those performed by AXGD:

$$\begin{cases} \hat{\mathbf{x}}^{(k)} = \frac{A_k}{A_{k+1}}\mathbf{x}^{(k)} + \frac{a_{k+1}}{A_{k+1}}\nabla\psi^*(\mathbf{z}^{(k)}). \\ \mathbf{x}^{(k+1)} = \frac{A_k}{A_{k+1}}\mathbf{x}^{(k)} + \frac{a_{k+1}}{A_{k+1}}\nabla\psi^*(\mathbf{z}^{(k)} - a_{k+1}\nabla f(\hat{\mathbf{x}}^{(k)})) \end{cases}$$

We can now analyze AXGD as producing an approximate solution to the implicit Euler discretization problem. The following lemma gives a general bound on the convergence of AXGD for a convex and differentiable $f(\cdot)$ without additional assumptions. The only (mild) difference is replacing $D_\psi(\mathbf{x}, \mathbf{x}^{(1)})$ and $D_\psi(\mathbf{x}^*, \mathbf{x}^{(1)})$ by $D_\psi(\mathbf{x}, \hat{\mathbf{x}}^{(0)})$ and $D_\psi(\mathbf{x}^*, \hat{\mathbf{x}}^{(0)})$, since we start from the "intermediate" point $\hat{\mathbf{x}}^{(0)}$. This change is only important for bounding the initial gap G_1; everything else is the same as before.

▶ **Lemma 10.** *Consider the* AXGD *algorithm as described in Equation (2), starting from an arbitrary point $\hat{\mathbf{x}}^{(0)}$ with $\mathbf{z}^{(0)} = \nabla\psi(\hat{\mathbf{x}}^{(0)})$ and $A_0 = 0$. Then the error from Lemma 9 is bounded by:*

$$\begin{aligned} E_{i+1} \leq & a_{i+1}\left\langle \nabla f(\mathbf{x}^{(i+1)}) - \nabla f(\hat{\mathbf{x}}^{(i)}), \nabla\psi^*(\hat{\mathbf{z}}^{(i)}) - \nabla\psi^*(\mathbf{z}^{(i+1)})\right\rangle \\ & - D_{\psi^*}(\hat{\mathbf{z}}^{(i)}, \mathbf{z}^{(i+1)}) - D_{\psi^*}(\mathbf{z}^{(i)}, \hat{\mathbf{z}}^{(i)}). \end{aligned}$$

Proof. From Lemma 9:

$$\begin{aligned}E_{i+1} \leq & a_{i+1}\left\langle \nabla f(\mathbf{x}^{(i+1)}), \nabla\psi^*(\hat{\mathbf{z}}^{(i)}) - \nabla\psi^*(\mathbf{z}^{(i+1)})\right\rangle - D_{\psi^*}(\mathbf{z}^{(i)}, \mathbf{z}^{(i+1)}) \\ = & a_{i+1}\left\langle \nabla f(\mathbf{x}^{(i+1)}) - \nabla f(\hat{\mathbf{x}}^{(i)}) + \nabla f(\hat{\mathbf{x}}^{(i)}), \nabla\psi^*(\hat{\mathbf{z}}^{(i)}) - \nabla\psi^*(\mathbf{z}^{(i+1)})\right\rangle \\ & - D_{\psi^*}(\mathbf{z}^{(i)}, \mathbf{z}^{(i+1)}).\end{aligned}$$

We now use the fact that $a_{i+1}f(\hat{\mathbf{x}}^{(i)}) = \mathbf{z}^{(i)} - \hat{\mathbf{z}}^{(i)}$ together with the standard triangle-inequality for Bregman divergences (see Proposition 17) to show that:

$$\begin{aligned}a_{i+1}\left\langle \nabla f(\hat{\mathbf{x}}^{(i)}), \nabla\psi^*(\hat{\mathbf{z}}^{(i)}) - \nabla\psi^*(\mathbf{z}^{(i+1)})\right\rangle &= \left\langle \mathbf{z}^{(i)} - \hat{\mathbf{z}}^{(i)}, \nabla\psi^*(\hat{\mathbf{z}}^{(i)}) - \nabla\psi^*(\mathbf{z}^{(i+1)})\right\rangle \\ &= D_{\psi^*}(\mathbf{z}^{(i)}, \mathbf{z}^{(i+1)}) - D_{\psi^*}(\hat{\mathbf{z}}^{(i)}, \mathbf{z}^{(i+1)}) - D_{\psi^*}(\mathbf{z}^{(i)}, \hat{\mathbf{z}}^{(i)}),\end{aligned}$$

Combining the results of the last two equations, we get the claimed bound on the error. ◀

2.5 Smooth Minimization with AXGD

We show that AXGD achieves the asymptotically optimal convergence rate of $1/k^2$ for the minimization of an L-smooth convex objective $f(\cdot)$ by applying Lemma 10. The crux of the proof is that we can take sufficiently large steps while keeping the error from Lemma 10 non-positive. In other words, we are able to move quickly through the continuous evolution of AMD by taking large discrete steps.

▶ **Theorem 11.** *Let $f : X \to \mathbb{R}$ be an L-smooth convex function and let $\mathbf{x}^{(k)}, \mathbf{z}^{(k)}, \hat{\mathbf{x}}^{(k)}, \hat{\mathbf{z}}^{(k)}$ be updated according to the* AXGD *algorithm in Equation (2), starting from an arbitrary initial point $\hat{\mathbf{x}}^{(0)} \in X$ with the following initial conditions: $\mathbf{z}^{(0)} = \nabla\psi(\hat{\mathbf{x}}^{(0)})$ and $A_0 = 0$. Let $\psi : X \to \mathbb{R}$ be σ-strongly convex. If $\frac{a_k^2}{A_k} \leq \frac{\sigma}{L}$, then for all $k \geq 1$,*

$$f(\mathbf{x}^{(k)}) - f(\mathbf{x}^*) \leq \frac{D_\psi(\mathbf{x}^*, \hat{\mathbf{x}}^{(0)})}{A_k}.$$

In particular, if $a_k = \frac{k+1}{2} \cdot \frac{\sigma}{L}$ and $\psi(\mathbf{x}) = \frac{\sigma}{2}\|\mathbf{x}\|^2$, then:

$$f(\mathbf{x}^{(k)}) - f(\mathbf{x}^*) \leq \frac{2L}{(k+1)^2}\|\mathbf{x}^* - \hat{\mathbf{x}}^{(0)}\|^2.$$

Proof. The proof follows directly by applying Lemma 10 and using L-smoothness of f and σ-strong convexity of ψ. In particular, by Cauchy-Schwartz inequality and smoothness:

$$\begin{aligned}&\left\langle \nabla f(\mathbf{x}^{(k+1)}) - \nabla f(\hat{\mathbf{x}}^{(k)}), \nabla\psi^*(\hat{\mathbf{z}}^{(k)}) - \nabla\psi^*(\mathbf{z}^{(k+1)})\right\rangle \\ &\qquad \leq L\|\mathbf{x}^{(k+1)} - \hat{\mathbf{x}}^{(k)}\| \cdot \|\nabla\psi^*(\mathbf{z}^{(k+1)}) - \nabla\psi^*(\hat{\mathbf{z}}^{(k)})\|,\end{aligned}$$

and, by Proposition 16

$$\begin{aligned}&D_{\psi^*}(\hat{\mathbf{z}}^{(k)}, \mathbf{z}^{(k+1)}) + D_{\psi^*}(\mathbf{z}^{(k)}, \hat{\mathbf{z}}^{(k)}) \\ &\qquad \geq \frac{\sigma}{2}\left(\|\nabla\psi^*(\hat{\mathbf{z}}^{(k)}) - \nabla\psi^*(\mathbf{z}^{(k+1)})\|^2 + \|\nabla\psi^*(\mathbf{z}^{(k)}) - \nabla\psi^*(\hat{\mathbf{z}}^{(k)})\|^2\right).\end{aligned} \tag{9}$$

From the definition of the steps, $\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)} = \frac{a_{k+1}}{A_{k+1}}(\nabla\psi^*(\hat{\mathbf{z}}^{(k)}) - \nabla\psi^*(\mathbf{z}^{(k)}))$, and, therefore:

$$E_{k+1} \leq \frac{a_{k+1}^2}{A_{k+1}}L \cdot pq - \frac{\sigma}{2}(p^2 + q^2),$$

where $p = \|\nabla\psi^*(\hat{\mathbf{z}}^{(k)}) - \nabla\psi^*(\mathbf{z}^{(k+1)})\|$ and $q = \|\nabla\psi^*(\mathbf{z}^{(k)}) - \nabla\psi^*(\hat{\mathbf{z}}^{(k)})\|$. Since, for any p, q, $p^2 + q^2 - 2\alpha pq \geq 0$ whenever $\alpha \leq 1$, it follows that $E_{k+1} \leq 0$ whenever $\frac{{a_{k+1}}^2}{A_{k+1}} \frac{L}{\sigma} \leq 1$, which is true by the theorem assumptions. In particular, for $a_k = \frac{k+1}{2} \cdot \frac{\sigma}{L}$, $A_k = \frac{\sigma}{L}(\frac{(k+1)(k+2)}{4}) \geq \frac{\sigma}{L}\frac{(k+1)^2}{4}$. This proves that $f(\mathbf{x}^{(k)}) - f(\mathbf{x}^*) \leq \frac{G_1}{A_k}$. It remains to bound G_1. This a simple computation, shown in the appendix, which yields: $G_1 \leq \frac{1}{A_1} D_\psi(\mathbf{x}^*, \hat{\mathbf{x}}^{(0)})$. ◀

2.6 Generalized Smoothness: Hölder-Continuous Gradients

Suppose that $f(\cdot)$ has Hölder-continuous gradients, namely, $f(\cdot)$ then satisfies:

$$\|\nabla f(\hat{\mathbf{x}}) - \nabla f(\mathbf{x})\| \leq L_\nu \|\hat{\mathbf{x}} - \mathbf{x}\|^\nu, \tag{10}$$

which also implies:

$$\forall \mathbf{x}, \hat{\mathbf{x}} \in X : \; f(\hat{\mathbf{x}}) \leq f(\mathbf{x}) + \langle \nabla f(\mathbf{x}), \hat{\mathbf{x}} - \mathbf{x} \rangle + \frac{L_\nu}{1+\nu} \|\hat{\mathbf{x}} - \mathbf{x}\|^{1+\nu}, \tag{11}$$

where $\nu \in (0, 1]$, $L_\nu \in \mathbb{R}_{++}$. In particular, if $\nu = 1$, then $f(\cdot)$ is L_ν-smooth. Thus, the functions with Hölder-continuous gradients represent a class of functions with generalized/relaxed smoothness properties.

The lower iteration complexity bound for (unconstrained) minimization of convex functions with Hölder-continuous gradients was established in [20] and equals $O\left(L_\nu D_1^{1+\nu} \epsilon^{-\frac{2}{1+3\nu}}\right)$, where D_1 is the distance from the initial point to the optimal solution. A matching upper bound was obtained in [19].

To recover the optimal convergence rate in the minimization of convex functions with Hölder-continuous gradients, as before, we bound the discretization error from Lemma 10. Before doing so, we will need the following technical proposition (which appears in a similar form as Lemma 3.1 a) in [18]).

▶ **Proposition 12.**

$$a_{i+1} \left\langle \nabla f(\mathbf{x}^{(i+1)}) - \nabla f(\hat{\mathbf{x}}^{(i)}), \nabla\psi^*(\mathbf{z}^{(i+1)}) - \nabla\psi^*(\hat{\mathbf{z}}^{(i)}) \right\rangle \leq \sigma^{-1} {a_{i+1}}^2 \|\nabla f(\mathbf{x}^{(i+1)}) - \nabla f(\hat{\mathbf{x}}^{(i)})\|^2.$$

The proof is provided in the appendix.

▶ **Theorem 13.** *Let $f(\cdot)$ be a convex function that satisfies (10), and let $\psi(\cdot)$ be σ-strongly convex. Let $\mathbf{x}^{(k)}, \mathbf{z}^{(k)}, \hat{\mathbf{x}}^{(k)}, \hat{\mathbf{z}}^{(k)}$ be updated according to the* AXGD *algorithm in Equation (2), starting from an arbitrary initial point $\hat{\mathbf{x}}^{(0)} \in X$ with the following initial conditions: $\mathbf{z}^{(0)} = \nabla\psi(\hat{\mathbf{x}}^{(0)})$ and $A_0 = 0$. Let $a_k = c\frac{\sigma}{L_\nu} D^{1-\nu} k^{\frac{-1+3\nu}{2}}$, where $D = \max_{\mathbf{x},\hat{\mathbf{x}} \in X} \|\mathbf{x} - \hat{\mathbf{x}}\|$ and $c = 2^{\frac{3\nu(\nu+1)-1}{2}}$. Then, $\forall k \geq 1$:*

$$f(\mathbf{x}^{(k)}) - f(\mathbf{x}^*) \leq 2^{\frac{1-3\nu(\nu+1)}{2}} \frac{L_\nu}{\sigma} \frac{D^{\nu-1} D_\psi(\mathbf{x}^*, \hat{\mathbf{x}}^{(0)})}{k^{\frac{1+3\nu}{2}}}.$$

In particular, if $\psi(\mathbf{x}) = \frac{\sigma}{2}\|\mathbf{x}\|^2$, then:

$$f(\mathbf{x}^{(k)}) - f(\mathbf{x}^*) \leq 2^{\frac{1-3\nu(\nu+1)}{2}} L_\nu D^{1+\nu} k^{-\frac{1+3\nu}{2}}.$$

Proof. We prove the theorem by bounding the discretization error E_{i+1} from Lemma 10. Applying Propositions 16 and 12:

$$\begin{aligned}
E_{i+1} =& a_{i+1}\left\langle \nabla f(\mathbf{x}^{(i+1)}) - \nabla f(\hat{\mathbf{x}}^{(i)}), \nabla\psi^*(\hat{\mathbf{z}}^{(i)}) - \nabla\psi^*(\mathbf{z}^{(i+1)})\right\rangle \\
&- D_{\psi^*}(\hat{\mathbf{z}}^{(i)}, \mathbf{z}^{(i+1)}) - D_{\psi^*}(\mathbf{z}^{(i)}, \hat{\mathbf{z}}^{(i)}) \\
\leq& \sigma^{-1}{a_{i+1}}^2\|\nabla f(\mathbf{x}^{(i+1)}) - \nabla f(\hat{\mathbf{x}}^{(i)})\|^2 \\
&- \frac{\sigma}{2}\left(\|\nabla\psi^*(\hat{\mathbf{z}}^{(i)}) - \nabla\psi^*(\mathbf{z}^{(i+1)})\|^2 + \|\nabla\psi^*(\mathbf{z}^{(i)}) - \nabla\psi^*(\hat{\mathbf{z}}^{(i)})\|^2\right) \\
\leq& \sigma^{-1}{a_{i+1}}^2{L_\nu}^2\|\mathbf{x}^{(i+1)} - \hat{\mathbf{x}}^{(i)}\|^{2\nu} - \frac{\sigma}{2}\|\nabla\psi^*(\mathbf{z}^{(i)}) - \nabla\psi^*(\hat{\mathbf{z}}^{(i)})\|^2 \\
\leq& \sigma^{-1}{L_\nu}^2\frac{{a_{i+1}}^{2+2\nu}}{{A_{i+1}}^{2\nu}}\|\nabla\psi^*(\mathbf{z}^{(i)}) - \nabla\psi^*(\hat{\mathbf{z}}^{(i)})\|^{2\nu} - \frac{\sigma}{2}\|\nabla\psi^*(\mathbf{z}^{(i)}) - \nabla\psi^*(\hat{\mathbf{z}}^{(i)})\|^2, \quad (12)
\end{aligned}$$

where the second inequality is by (10) and the third inequality is by the step definition (2).

Taking $a_k = c\frac{\sigma}{L_\nu}D^{1-\nu}k^{\frac{-1+3\nu}{2}}$, where $c = 2^{\frac{3\nu(\nu+1)-1}{2}}$, it follows that $A_k = \sum_{i=1}^k a_i \geq \sum_{i=\lceil k/2\rceil}^k a_i \geq \frac{c}{2}D^{1-\nu}\frac{\sigma}{L_\nu}\left(\frac{k}{2}\right)^{\frac{1+3\nu}{2}}$. Therefore, the expression in (12) is at least:

$$\Big(-c^2 2^{3\nu(\nu+1)}(k+1)^{\nu-1} + \frac{1}{2}\Big)\sigma\|\nabla\psi^*(\mathbf{z}^{(k)}) - \nabla\psi^*(\hat{\mathbf{z}}^{(k)})\|^2 \geq 0,$$

as $(k+1)^{\nu-1} \leq 1$. Therefore, we have that $G_k \leq \frac{A^{(1)}}{A^{(k)}}G_1$, and using similar arguments to bound the initial gap G_1, the proof follows. ◀

2.7 Non-Smooth Minimization: Lipschitz-Continuous Objective

We now show that we can recover the well-known $\frac{1}{\sqrt{k}}$ convergence rate for the class of non-smooth L-Lipschitz objectives by using AXGD. This is summarized in the following theorem. We note that, as in the analysis of classical mirror descent (see, e.g., [3]), the factor $\log(k)$ can be removed if we fix the approximation error (and, consequently, the number of steps k) in advance.

▶ **Theorem 14.** *Let $f(\cdot)$ be a Lipschitz-continuous function with parameter L. Let $\mathbf{x}^{(k)}$, $\mathbf{z}^{(k)}$, $\hat{\mathbf{x}}^{(k)}$, $\hat{\mathbf{z}}^{(k)}$ be updated according to the* AXGD *algorithm in Equation (2), starting from an arbitrary initial point $\hat{\mathbf{x}}^{(0)} \in X$ with the following initial conditions: $\mathbf{z}^{(0)} = \nabla\psi(\hat{\mathbf{x}}^{(0)})$ and $A_0 = 0$. If $a_k = \frac{\sqrt{\sigma}}{2\sqrt{2}L}\sqrt{\frac{D_\psi(\mathbf{x}^*, \hat{\mathbf{x}}^{(0)})}{k}}$, then, $\forall k \geq 1$:*

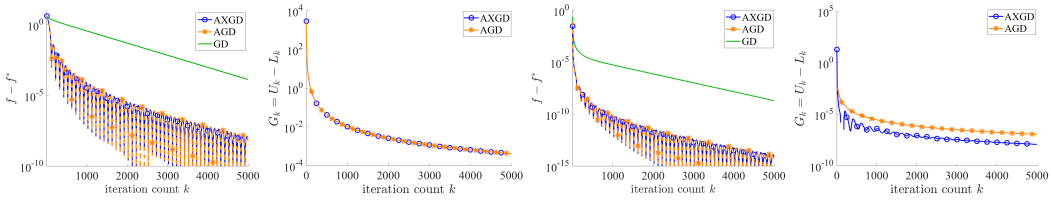
$$f(\mathbf{x}^{(k)}) - f(\mathbf{x}^*) \leq 8(2 + \log(k))\frac{L\cdot\sqrt{D_\psi(\mathbf{x}^*, \hat{\mathbf{x}}^{(0)})}}{\sqrt{\sigma}\sqrt{k}}.$$

In particular, for $\psi(\mathbf{x}) = \frac{\sigma}{2}\|\mathbf{x}\|^2$:

$$f(\mathbf{x}^{(k)}) - f(\mathbf{x}^*) \leq 4\sqrt{2}(2 + \log(k))\frac{L\cdot\|\mathbf{x}^* - \hat{\mathbf{x}}^{(0)}\|}{\sqrt{k}}.$$

Proof. As before, we bound the discretization error from Lemma 10. As $f(\cdot)$ is L-Lipschitz, using Proposition 16:

$$\begin{aligned}
E_{i+1} \leq& a_{i+1}\left\langle \nabla f(\mathbf{x}^{(i+1)}) - \nabla f(\hat{\mathbf{x}}^{(i)}), \nabla\psi^*(\hat{\mathbf{z}}^{(i)}) - \nabla\psi^*(\mathbf{z}^{(i+1)})\right\rangle \\
&- D_{\psi^*}(\hat{\mathbf{z}}^{(i)}, \mathbf{z}^{(i+1)}) - D_{\psi^*}(\mathbf{z}^{(i)}, \hat{\mathbf{z}}^{(i)}) \\
\leq& 2a_{i+1}L\|\nabla\psi^*(\mathbf{z}^{(i+1)}) - \nabla\psi^*(\hat{\mathbf{z}}^{(i)})\| - \frac{\sigma}{2}\|\nabla\psi^*(\mathbf{z}^{(i+1)}) - \nabla\psi^*(\hat{\mathbf{z}}^{(i)})\|^2 \\
\leq& \frac{8(a^{(i+1)}L)^2}{\sigma},
\end{aligned}$$



(a) unconstrained **(b)** unconstrained **(c)** simplex **(d)** simplex

Figure 1 (a),(c) Exact and (b),(d) approximate duality gaps for AGD and AXGD with exact gradients.

where we have used the inequality $2xy - x^2 \leq y^2$, $\forall x, y$. As $\sigma \geq L$ and

$$A_k \cdot \frac{2\sqrt{2}L}{\sqrt{\sigma D_\psi(\mathbf{x}^*, \hat{\mathbf{x}}^{(0)})}} = \sum_{i=1}^{k} \frac{1}{\sqrt{k}} \geq \sum_{i=\lceil k/2 \rceil}^{k} \frac{1}{\sqrt{k}} \geq \frac{1}{2} \cdot \sqrt{\frac{k}{2}},$$

we have that

$$\sum_{i=1}^{k} \frac{E_i}{A_k} \leq 8 \cdot \frac{L \cdot \sqrt{D_\psi(\mathbf{x}^*, \hat{\mathbf{x}}^{(0)})}}{\sqrt{\sigma}\sqrt{k}} (\log(k) + 1),$$

which, after bounding the initial gap by similar arguments, completes the proof. ◀

3 Experiments

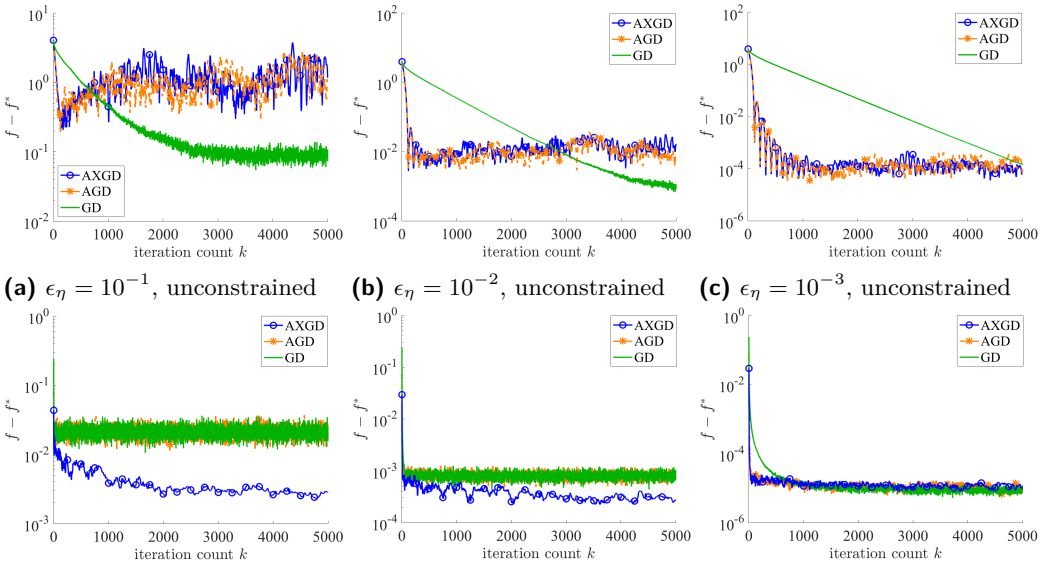
We now illustrate the performance of AGD and AXGD for (i) an unconstrained problem over $\mathbb{R}^n$ with the objective function $f(\mathbf{x}) = \frac{1}{2}\langle \mathbf{A}\mathbf{x}, \mathbf{x}\rangle - \langle \mathbf{b}, \mathbf{x}\rangle$, and (ii) for the problem with the same objective and unit simplex as the feasible region, where $\mathbf{A}$ is the Laplacian of a cycle graph[5] and $\mathbf{b}$ is a vector whose first element is one and the remaining elements are zero. This example is known as a "hard" instance for smooth minimization – it is typically used in proving the lower iteration complexity bound for first-order methods (see, e.g., [28]). We also include Gradient Descent (GD) in the exact gap graphs for comparison. In the experiments, we take $n = 100$ and $\sigma = L$ $(= 4)$. We use the ℓ_2 norm in the gradient steps.

In the figures, f denotes the objective value at the upper-bound point and f^* denotes the optimal objective value, so that $f - f^*$ is the true distance to the optimum (the exact gap). Fig. 1 shows the distance to the optimum and the approximate duality gap $G_k = U_k - L_k$ obtained using our analysis. We can observe that AGD and AXGD exhibit similar performance in these examples. The approximate gap overestimates the actual duality gap, however, the difference between the two decreases with the number of iterations.

Acceleration and Noise

We now consider the setting in which the gradients output by our oracle are corrupted by additive noise, which has significant applications in practice [10] and theory [2]. We note

[5] Namely, the sum of a tridiagonal matrix $\mathbf{B}$ with 2's on its main diagonal and -1's on its remaining two diagonals and a matrix $\mathbf{C}$ whose all elements are zero except for the $\mathbf{C}_{1,n} = \mathbf{C}_{n,1} = -1$.



(a) $\epsilon_\eta = 10^{-1}$, unconstrained **(b)** $\epsilon_\eta = 10^{-2}$, unconstrained **(c)** $\epsilon_\eta = 10^{-3}$, unconstrained

(d) $\epsilon_\eta = 10^{-1}$, simplex **(e)** $\epsilon_\eta = 10^{-2}$, simplex **(f)** $\epsilon_\eta = 10^{-3}$, simplex

Figure 2 Exact gap for additive Gaussian noise in the gradients with zero mean and covariance $\epsilon_\eta I$ (a)-(c) in the unconstrained-region case and (d)-(f) in unit simplex.

that this model is fundamentally different from the inexact model considered by Devolder *et al.* [6], for which tight lower bounds preventing acceleration exist.[6]

Specifically, we experimentally evaluate the performance of AGD and AXGD under additive Gaussian noise. Fig. 2 illustrates the performance of AGD and AXGD when the gradients are corrupted by zero-mean additive Gaussian noise with covariance matrix $\epsilon_\eta I$, where I is the identity matrix. When the region is unconstrained (top row in Fig. 2), both AGD and AXGD exhibit high sensitivity to noise. The GD method overall exhibits higher tolerance to noise (at the expense of slower convergence). In the case of the unit simplex region (bottom row in Fig. 2), all the algorithms appear more tolerant to noise than in the unconstrained case. Interestingly, on this example AXGD exhibits higher tolerance to noise than GD and AGD, both in terms of mean and in terms of variance. Explaining this phenomenon analytically is an interesting question that merits further investigation.

4 Conclusion

We have presented a novel accelerated method – AXGD– that combines ideas from the Nesterov's AGD and Nemirovski's mirror prox. AXGD achieves optimal convergence rates for a range of convex optimization problems, such as the problems with the (i) smooth objectives, (ii) objectives with Hölder-continuous gradients, (iii) and non-smooth Lipschitz-continuous objectives. In the constrained-regime experiments from Section 3, the method demonstrates favorable performance compared to AGD when subjected to zero-mean Gaussian noise.

There are several directions that merit further investigation. A more thorough analytical and experimental study of acceleration when the gradients are corrupted by noise is of

[6] In [6], it is assumed that a function $f(\cdot)$ is associated with a (δ, L) oracle, such that $f(\hat{\mathbf{x}}) \leq f(\mathbf{x}) + \langle \nabla f(\mathbf{x}), \hat{\mathbf{x}} - \mathbf{x} \rangle + \frac{L}{2}\|\hat{\mathbf{x}} - \mathbf{x}\|^2 + \frac{\delta}{2}$, $\forall \mathbf{x}, \hat{\mathbf{x}} \in X$. Such a model seems more suitable for incorrectly specified functions (e.g., non-smooth functions treated as being smooth) and adversarially perturbed functions.

particular interest, since the gradients can often come from noise-corrupted measurements. Further, our experiments from Fig. 2 suggest that there are cases that incur a trade-off between noise tolerance and acceleration. A systematic study of this trade-off is thus another important direction, since it would guide the choice of accelerated/non-accelerated algorithms in practice depending on the application. Finally, it is interesting to investigate whether restart schemes can improve the algorithms' noise tolerance, since in the noiseless setting several restart schemes are known to improve the convergence of AGD in practice.

References

1. Zeyuan Allen-Zhu and Lorenzo Orecchia. Linear coupling: An ultimate unification of gradient and mirror descent. In *Proc. ITCS'17*, 2017.
2. Raef Bassily, Adam Smith, and Abhradeep Thakurta. Private empirical risk minimization: Efficient algorithms and tight error bounds. In *Proc. IEEE FOCS'14*, 2014.
3. Aharon Ben-Tal and Arkadi Nemirovski. *Lectures on modern convex optimization: Analysis, algorithms, and engineering applications*. MPS-SIAM Series on Optimization. SIAM, 2001.
4. Sébastien Bubeck. Theory of convex optimization for machine learning. *CoRR*, abs/1405.4980, 2014. `arXiv:1405.4980v1`.
5. Sébastien Bubeck, Yin Tat Lee, and Mohit Singh. A geometric alternative to nesterov's accelerated gradient descent. *CoRR*, abs/1506.08187, 2015. `arXiv:1506.08187`.
6. Olivier Devolder, François Glineur, and Yurii Nesterov. First-order methods of smooth convex optimization with inexact oracle. *Math. Prog.*, 146(1-2):37–75, 2014.
7. Jelena Diakonikolas and Lorenzo Orecchia. The approximate gap technique: A unified approach to optimal first-order methods, 2017. Manuscript.
8. A. Ene and H. L. Nguyen. Constrained submodular maximization: Beyond 1/e. In *Proc. IEEE FOCS'16*, 2016.
9. E Hairer, SP Nørsett, and G Wanner. *Solving Ordinary Differential Equations I (2nd Revised. Ed.): Nonstiff Problems*. Springer Ser. Comput. Math. Springer-Verlag New York, Inc., 1993.
10. Moritz Hardt. Robustness vs acceleration, 2014. URL: `http://blog.mrtz.org/2014/08/18/robustness-versus-acceleration.html`.
11. Rahul Jain, Zhengfeng Ji, Sarvagya Upadhyay, and John Watrous. QIP = PSPACE. *Journal of the ACM (JACM)*, 58(6):30, 2011.
12. Jonathan A. Kelner, Yin Tat Lee, Lorenzo Orecchia, and Aaron Sidford. An almost-linear-time algorithm for approximate max flow in undirected graphs, and its multicommodity generalizations. In *Proc. ACM-SIAM SODA'14*, 2014.
13. Jonathan A. Kelner, Lorenzo Orecchia, Aaron Sidford, and Zeyuan Allen Zhu. A simple, combinatorial algorithm for solving SDD systems in nearly-Linear time. In *Proc. ACM STOC'13*, 2013.
14. G. M. Korpelevich. The extragradient method for finding saddle points and other problems. *Matekon : translations of Russian & East European mathematical economics*, 13(4):35–49, 1977.
15. Walid Krichene, Alexandre Bayen, and Peter L Bartlett. Accelerated mirror descent in continuous and discrete time. In *Proc. NIPS'15*, 2015.
16. Guanghui Lan. An optimal method for stochastic composite optimization. *Math. Prog.*, 133(1-2):365–397, January 2011.
17. Yin Tat Lee, Satish Rao, and Nikhil Srivastava. A new approach to computing maximum flows using electrical flows. In *Proc. ACM STOC '13*, 2013.

18 Arkadi Nemirovski. Prox-method with rate of convergence $O(1/t)$ for variational inequalities with Lipschitz continuous monotone operators and smooth convex-concave saddle point problems. *SIAM J. Optimiz.*, 15(1):229–251, 2004.

19 Arkadi S Nemirovski and Yurii Evgen'evich Nesterov. Optimal methods of smooth convex minimization. *Zh. Vychisl. Mat. i Mat. Fiz.*, 25(3):356–369, 1985.

20 Arkadii Nemirovskii and David Borisovich Yudin. *Problem complexity and method efficiency in optimization*. Wiley, 1983.

21 Yu Nesterov. Universal gradient methods for convex optimization problems. *Math. Prog.*, 152(1-2):381–404, 2015.

22 Yurii Nesterov. A method of solving a convex programming problem with convergence rate $O(1/k^2)$. In *Doklady AN SSSR (translated as Soviet Mathematics Doklady)*, volume 269, pages 543–547, 1983.

23 Yurii Nesterov. *Introductory Lectures on Convex Programming Volume: A Basic course*, volume I. Kluwer Academic Publishers, 2004.

24 Yurii Nesterov. Excessive gap technique in nonsmooth convex minimization. *SIAM J. Optimiz.*, 16(1):235–249, January 2005.

25 Yurii Nesterov. Smooth minimization of non-smooth functions. *Math. Prog.*, 103(1):127–152, December 2005.

26 Yurii Nesterov. Accelerating the cubic regularization of Newton's method on convex problems. *Math. Prog.*, 112(1):159–181, 2008.

27 Yurii Nesterov. Gradient methods for minimizing composite functions. *Math. Prog.*, 140(1):125–161, 2013.

28 Yurii Nesterov. *Introductory lectures on convex optimization: A basic course*. Springer Science & Business Media, 2013.

29 Jonah Sherman. Nearly maximum flows in nearly linear time. In *Proc. IEEE FOCS'13*, 2013.

30 Daniel A. Spielman and Shang-Hua Teng. Nearly-linear time algorithms for graph partitioning, graph sparsification, and solving linear systems. In *Proc. ACM STOC '04*, 2004.

31 Weijie Su, Stephen Boyd, and Emmanuel Candes. A differential equation for modeling nesterov's accelerated gradient method: Theory and insights. In *Proc. NIPS'14*, 2014.

32 Paul Tseng. On accelerated proximal gradient methods for convex-concave optimization, 2008.

33 Andre Wibisono, Ashia C Wilson, and Michael I Jordan. A variational perspective on accelerated methods in optimization. In *Proc. Natl. Acad. Sci. U.S.A.*, 2016.

A Properties of the Bregman Divergence

The following properties of Bregman divergence will be useful in our analysis.

▶ **Proposition 15.** $D_\psi(\nabla\psi^*(\mathbf{z}), \mathbf{x}) = D_{\psi^*}(\nabla\psi(\mathbf{x}), \mathbf{z})$, $\forall \mathbf{x}, \mathbf{z}$.

Proof. From the definition of ψ^* and Fact 6,

$$\psi^*(\mathbf{z}) = \langle \nabla\psi^*(\mathbf{z}), \mathbf{z}\rangle - \psi(\nabla\psi^*),\ \forall \mathbf{z}. \tag{13}$$

Similarly, as in the light of Fenchel-Moreau Theorem $\psi^{**} = \psi$,

$$\psi(\mathbf{x}) = \langle \nabla\psi(\mathbf{x}), \mathbf{x}\rangle - \psi^*(\nabla\psi(\mathbf{x})),\ \forall \mathbf{x}. \tag{14}$$

Using the definition of $D_\psi(\nabla\psi^*(\mathbf{z}), \mathbf{x})$ and Fact 6:

$$\begin{aligned} D_\psi(\nabla\psi^*(\mathbf{z}), \mathbf{x}) &= \psi(\nabla\psi^*(\mathbf{z})) - \psi(\mathbf{x}) - \langle \nabla\psi(\mathbf{x}), \nabla\psi^*(\mathbf{z}) - \mathbf{x}\rangle \\ &= \psi(\nabla\psi^*(\mathbf{z})) + \psi^*(\nabla\psi(\mathbf{x})) - \langle \nabla\psi(\mathbf{x}), \nabla\psi^*(\mathbf{z})\rangle\,. \end{aligned} \tag{15}$$

Similarly, using the definition of $D_{\psi^*}(\nabla\psi(\mathbf{x}), \mathbf{z})$ combined with (13):

$$\begin{aligned} D_{\psi^*}(\nabla\psi(\mathbf{x}), \mathbf{z}) &= \psi^*(\nabla\psi(\mathbf{x})) - \psi^*(\mathbf{z}) - \langle \nabla\psi^*(\mathbf{z}), \nabla\psi(\mathbf{x}) - \mathbf{z}\rangle \\ &= \psi^*(\nabla\psi(\mathbf{x})) + \psi(\nabla\psi^*(\mathbf{z})) - \langle \nabla\psi^*(\mathbf{z}), \nabla\psi(\mathbf{x})\rangle\,. \end{aligned} \tag{16}$$

Comparing (15) and (16), the proof follows. ◀

▶ **Proposition 16.** If $\psi(\cdot)$ is σ-strongly convex, then $D_{\psi^*}(\mathbf{z}, \hat{\mathbf{z}}) \geq \frac{\sigma}{2}\|\nabla\psi^*(\mathbf{z}) - \nabla\psi^*(\hat{\mathbf{z}})\|^2$.

Proof. Using the definition of $D_{\psi^*}(\mathbf{z}, \hat{\mathbf{z}})$ and (13), we can write $D_{\psi^*}(\mathbf{z}, \hat{\mathbf{z}})$ as:

$$D_{\psi^*}(\mathbf{z}, \hat{\mathbf{z}}) = \psi(\nabla\psi^*(\hat{\mathbf{z}})) - \psi(\nabla\psi^*(\mathbf{z})) - \langle \mathbf{z}, \nabla\psi^*(\hat{\mathbf{z}}) - \nabla\psi^*(\mathbf{z})\rangle\,.$$

Since $\psi(\cdot)$ is σ-strongly convex, it follows that:

$$D_{\psi^*}(\mathbf{z}, \hat{\mathbf{z}}) \geq \frac{\sigma}{2}\|\nabla\psi^*(\hat{\mathbf{z}}) - \nabla\psi^*(\mathbf{z})\|^2 + \langle \nabla\psi(\nabla\psi^*(\mathbf{z})) - \mathbf{z}, \nabla\psi^*(\hat{\mathbf{z}}) - \nabla\psi^*(\mathbf{z})\rangle\,.$$

As, from Fact 6, $\nabla\psi^*(\mathbf{z}) = \arg\max_{\mathbf{x}\in X}\{\langle \mathbf{x}, \mathbf{z}\rangle - \psi(\mathbf{x})\}$, by the first-order optimality condition

$$\langle \nabla\psi(\nabla\psi^*(\mathbf{z})) - \mathbf{z}, \nabla\psi^*(\hat{\mathbf{z}}) - \nabla\psi^*(\mathbf{z})\rangle \geq 0,$$

completing the proof. ◀

The Bregman divergence $D_{\psi^*}(\mathbf{x}, \mathbf{y})$ captures the difference between $\psi^*(\mathbf{x})$ and its first order approximation at $\mathbf{y}$. Notice that, for a differentiable ψ^*, we have:

$$\nabla_{\mathbf{x}} D_{\psi^*}(\mathbf{x}, \mathbf{y}) = \nabla\psi^*(\mathbf{x}) - \nabla\psi^*(\mathbf{y}).$$

The Bregman divergence $D_{\psi^*}(\mathbf{x}, \mathbf{y})$ is a convex function of $\mathbf{x}$. Its Bregman divergence is itself.

▶ **Proposition 17.** For all $\mathbf{x}, \mathbf{y}, \mathbf{z} \in X$

$$D_{\psi^*}(\mathbf{x}, \mathbf{y}) = D_{\psi^*}(\mathbf{z}, \mathbf{y}) + \langle \nabla\psi^*(\mathbf{z}) - \nabla\psi^*(\mathbf{y}), \mathbf{x} - \mathbf{z}\rangle + D_{\psi^*}(\mathbf{x}, \mathbf{z}).$$

B Omitted Proofs from Section 2

▶ **Proposition 7** (restated). *Let* $\mathbf{z}^{(t)} = \nabla\psi(\mathbf{x}^{(t_0)}) - \int_{t_0}^{t} \nabla f(\mathbf{x}^{(\tau)}) d\alpha^{(\tau)}$. *Then:*

$$\nabla\psi^*(\mathbf{z}^{(t)}) = \arg\min_{\mathbf{u}\in X}\left\{\int_{t_0}^{t} \left\langle \nabla f(\mathbf{x}^{(\tau)}), \mathbf{u} - \mathbf{x}^{(\tau)}\right\rangle d\alpha^{(\tau)} + D_{\psi}(\mathbf{u}, \mathbf{x}^{(t_0)})\right\}.$$

Proof. From the definition of Bregman divergence:

$$\begin{aligned} &\arg\min_{\mathbf{u}\in X}\left\{\int_{t_0}^{t} \left\langle \nabla f(\mathbf{x}^{(\tau)}), \mathbf{u} - \mathbf{x}^{(\tau)}\right\rangle d\alpha^{(\tau)} + D_{\psi}(\mathbf{u}, \mathbf{x}^{(t_0)})\right\} \\ &= \arg\min_{\mathbf{u}\in X}\left\{\int_{t_0}^{t} \left\langle \nabla f(\mathbf{x}^{(\tau)}), \mathbf{u} - \mathbf{x}^{(\tau)}\right\rangle d\alpha^{(\tau)} + \psi(\mathbf{u}) - \psi(\mathbf{x}^{(t_0)}) - \left\langle \nabla\psi(\mathbf{x}^{(t_0)}), \mathbf{u} - \mathbf{x}^{(t_0)}\right\rangle\right\} \\ &= \arg\min_{\mathbf{u}\in X}\left\{\left\langle \int_{t_0}^{t} \nabla f(\mathbf{x}^{(\tau)}) d\alpha^{(\tau)} - \nabla\psi(\mathbf{x}^{(t_0)}), \mathbf{u}\right\rangle + \psi(\mathbf{u})\right\}. \end{aligned}$$

Using the definition of $\mathbf{z}^{(t)}$ and Fact 6, the proof follows. ◀

Remaining Proof of Theorem 11 (The Bound on G_1). To bound G_1, we recall the definition of L_1:

$$\begin{aligned}L_1 &= f(\mathbf{x}^{(1)}) + \min_{\mathbf{x}\in X}\left\{\left\langle \nabla f(\mathbf{x}^{(1)}), \mathbf{x}-\mathbf{x}^{(1)}\right\rangle + \frac{1}{A_1}D_\psi(\mathbf{x},\hat{\mathbf{x}}^{(0)})\right\} - \frac{1}{A_1}D_\psi(\mathbf{x}^*,\hat{\mathbf{x}}^{(0)})\\ &= f(\mathbf{x}^{(1)}) + \left\langle \nabla f(\mathbf{x}^{(1)}), \nabla\psi^*(\mathbf{z}^{(1)})-\mathbf{x}^{(1)}\right\rangle\\ &\quad + \frac{1}{A_1}D_\psi(\nabla\psi^*(\mathbf{z}^{(1)}),\hat{\mathbf{x}}^{(0)}) - \frac{1}{A_1}D_\psi(\mathbf{x}^*,\hat{\mathbf{x}}^{(0)}).\end{aligned}$$

As $a_1 = A_1$, $\mathbf{x}^{(1)} = \nabla\psi^*(\hat{\mathbf{z}}^{(0)})$, and $a_1\nabla f(\hat{\mathbf{x}}^{(0)}) = \mathbf{z}^{(0)} - \hat{\mathbf{z}}^{(0)}$, using Proposition 17, we have that:

$$\begin{aligned}&\left\langle \nabla f(\hat{\mathbf{x}}^{(0)}), \nabla\psi^*(\mathbf{z}^{(1)})-\mathbf{x}^{(1)}\right\rangle\\ &\quad= \frac{1}{A_1}\left\langle \mathbf{z}^{(0)}-\hat{\mathbf{z}}^{(0)}, \nabla\psi^*(\mathbf{z}^{(1)})-\nabla\psi^*(\hat{\mathbf{z}}^{(0)})\right\rangle\\ &\quad= \frac{1}{A_1}\left(D_{\psi^*}(\mathbf{z}^{(0)},\hat{\mathbf{z}}^{(0)}) - D_{\psi^*}(\mathbf{z}^{(0)},\mathbf{z}^{(1)}) + D_{\psi^*}(\hat{\mathbf{z}}^{(0)},\mathbf{z}^{(1)})\right).\end{aligned} \tag{17}$$

On the other hand, by smoothness of $f(\cdot)$ and the initial condition:

$$\begin{aligned}&\left\langle \nabla f(\mathbf{x}^{(1)}) - \nabla f(\hat{\mathbf{x}}^{(0)}), \nabla\psi^*(\mathbf{z}^{(1)})-\mathbf{x}^{(1)}\right\rangle\\ &\quad\geq -L\|\nabla\psi^*(\hat{\mathbf{z}}^{(0)}) - \hat{\mathbf{x}}^{(0)}\|\|\nabla\psi^*(\mathbf{z}^{(1)})-\mathbf{x}^{(1)}\|.\end{aligned} \tag{18}$$

Finally, by Proposition 15 and the initial condition $\mathbf{z}^{(0)} = \nabla\psi(\hat{\mathbf{x}}^{(0)})$, we have that $D_{\psi^*}(\mathbf{z}^{(0)},\mathbf{z}^{(1)}) = D_\psi(\nabla\psi^*(\mathbf{z}^{(1)}),\hat{\mathbf{x}}^{(0)})$. Combining with (17), (18), and $G_1 = U_1 - L_1 = f(\mathbf{x}^{(1)}) - L_1$:

$$\begin{aligned}G_1 \leq& L\|\nabla\psi^*(\hat{\mathbf{z}}^{(0)}) - \hat{\mathbf{x}}^{(0)}\|\cdot\|\nabla\psi^*(\mathbf{z}^{(1)})-\mathbf{x}^{(1)}\|\\ &- \frac{1}{A_1}\left(D_{\psi^*}(\mathbf{z}^{(0)},\hat{\mathbf{z}}^{(0)}) + D_{\psi^*}(\hat{\mathbf{z}}^{(0)},\mathbf{z}^{(1)})\right) + \frac{1}{A_1}D_\psi(\mathbf{x}^*,\hat{\mathbf{x}}^{(0)})\\ =& L\|\nabla\psi^*(\hat{\mathbf{z}}^{(0)}) - \hat{\mathbf{x}}^{(0)}\|\cdot\|\nabla\psi^*(\mathbf{z}^{(1)})-\mathbf{x}^{(1)}\|\\ &- \frac{1}{A_1}\left(D_\psi(\nabla\psi^*(\hat{\mathbf{z}}^{(0)}),\hat{\mathbf{x}}^{(0)}) + D_{\psi^*}(\hat{\mathbf{z}}^{(0)},\mathbf{z}^{(1)})\right) + \frac{1}{A_1}D_\psi(\mathbf{x}^*,\hat{\mathbf{x}}^{(0)})\\ \leq& L\|\nabla\psi^*(\hat{\mathbf{z}}^{(0)}) - \hat{\mathbf{x}}^{(0)}\|\cdot\|\nabla\psi^*(\mathbf{z}^{(1)})-\mathbf{x}^{(1)}\|\\ &- \frac{\sigma}{2A_1}\left(\|\nabla\psi^*(\hat{\mathbf{z}}^{(0)}) - \hat{\mathbf{x}}^{(0)}\|^2 + \|\nabla\psi^*(\mathbf{z}^{(1)})-\mathbf{x}^{(1)}\|^2\right) + \frac{1}{A_1}D_\psi(\mathbf{x}^*,\hat{\mathbf{x}}^{(0)})\\ \leq& \frac{1}{A_1}D_\psi(\mathbf{x}^*,\hat{\mathbf{x}}^{(0)}),\end{aligned}$$

where we have used Proposition 15, $\mathbf{x}^{(1)} = \nabla\psi^*(\hat{\mathbf{z}}^{(0)})$, and $\frac{{a_1}^2}{A_1} = A_1 \leq \frac{\sigma}{L}$. ◀

▶ **Proposition 12** (restated)**.**

$$\begin{aligned}a_{i+1}&\left\langle \nabla f(\mathbf{x}^{(i+1)}) - \nabla f(\hat{\mathbf{x}}^{(i)}), \nabla\psi^*(\mathbf{z}^{(i+1)}) - \nabla\psi^*(\hat{\mathbf{z}}^{(i)})\right\rangle\\ &\leq \sigma^{-1}{a_{i+1}}^2\|\nabla f(\mathbf{x}^{(i+1)}) - \nabla f(\hat{\mathbf{x}}^{(i)})\|^2.\end{aligned}$$

Proof. From the first order optimality condition in Fact 6, $\forall \mathbf{x},\mathbf{y}\in X$:

$$\left\langle \nabla\psi(\nabla\psi^*(\mathbf{z}^{(i+1)})) - \mathbf{z}^{(i+1)}, \mathbf{x} - \nabla\psi^*(\mathbf{z}^{(i+1)})\right\rangle \geq 0, \text{ and} \tag{19}$$

$$\left\langle \nabla\psi(\nabla\psi^*(\hat{\mathbf{z}}^{(i)})) - \hat{\mathbf{z}}^{(i)}, \mathbf{y} - \nabla\psi^*(\hat{\mathbf{z}}^{(i)})\right\rangle \geq 0. \tag{20}$$

Letting $\mathbf{x} = \nabla\psi^*(\hat{\mathbf{z}}^{(i)})$, $\mathbf{y} = \nabla\psi^*(\mathbf{z}^{(i+1)})$, and summing (19) and (20):

$$\begin{aligned}\left\langle \hat{\mathbf{z}}^{(i)} - \mathbf{z}^{(i+1)}, \nabla\psi^*(\hat{\mathbf{z}}^{(i)}) - \nabla\psi^*(\mathbf{z}^{(i+1)}) \right\rangle \\ \geq \left\langle \nabla\psi(\nabla\psi^*(\hat{\mathbf{z}}^{(i)})) - \nabla\psi(\nabla\psi^*(\mathbf{z}^{(i+1)})), \nabla\psi^*(\hat{\mathbf{z}}^{(i)}) - \nabla\psi^*(\mathbf{z}^{(i+1)}) \right\rangle \\ \geq \sigma\|\nabla\psi^*(\hat{\mathbf{z}}^{(i)}) - \nabla\psi^*(\mathbf{z}^{(i+1)})\|^2, \qquad (21)\end{aligned}$$

where (21) follows by the σ-strong convexity of $\psi(\cdot)$. Using the Cauchy-Schwartz inequality and dividing both sides by $\|\nabla\psi^*(\hat{\mathbf{z}}^{(i)}) - \nabla\psi^*(\mathbf{z}^{(i+1)})\|$ gives $\|\hat{\mathbf{z}}^{(i)} - \mathbf{z}^{(i+1)}\| \geq \sigma\|\nabla\psi^*(\hat{\mathbf{z}}^{(i)}) - \nabla\psi^*(\mathbf{z}^{(i+1)})\|$.

Since, by the step definition (2), $\hat{\mathbf{z}}^{(i)} - \mathbf{z}^{(i+1)} = a_{i+1}(\nabla f(\mathbf{x}^{(i+1)}) - \nabla f(\hat{\mathbf{x}}^{(i)}))$, applying Cauchy-Schwartz Inequality to $a_{i+1}\left\langle \nabla f(\mathbf{x}^{(i+1)}) - \nabla f(\hat{\mathbf{x}}^{(i)}), \nabla\psi^*(\mathbf{z}^{(i+1)}) - \nabla\psi^*(\hat{\mathbf{z}}^{(i)}) \right\rangle$ completes the proof. ◀

Alternating Minimization, Scaling Algorithms, and the Null-Cone Problem from Invariant Theory*†

Peter Bürgisser[1], Ankit Garg[2], Rafael Oliveira[3], Michael Walter[4], and Avi Wigderson[5]

1 **Institut für Mathematik, Technische Universität Berlin, Germany**
pbuerg@math.tu-berlin.de
2 **Microsoft Research New England, Cambridge, USA**
garga@microsoft.com
3 **Department of Computer Science, University of Toronto, Canada and Department of Computer Science, Princeton University, USA**
rafael@cs.toronto.edu
4 **QuSoft, Korteweg-de Vries Institute for Mathematics, Institute of Physics, and Institute for Logic, Language and Computation, University of Amsterdam, The Netherlands and Stanford Institute for Theoretical Physics, Stanford University, USA**
m.walter@uva.nl
5 **Institute for Advanced Study, Princeton, USA**
avi@ias.edu

Abstract

Alternating minimization heuristics seek to solve a (difficult) global optimization task through iteratively solving a sequence of (much easier) local optimization tasks on different parts (or blocks) of the input parameters. While popular and widely applicable, very few examples of this heuristic are rigorously shown to converge to optimality, and even fewer to do so efficiently.

In this paper we present a general framework which is amenable to rigorous analysis, and expose its applicability. Its main feature is that the local optimization domains are each a group of invertible matrices, together naturally acting on tensors, and the optimization problem is minimizing the norm of an input tensor under this joint action. The solution of this optimization problem captures a basic problem in Invariant Theory, called the *null-cone problem*.

This algebraic framework turns out to encompass natural computational problems in combinatorial optimization, algebra, analysis, quantum information theory, and geometric complexity theory. It includes and extends to high dimensions the recent advances on (2-dimensional) *operator scaling* [14, 11, 22].

Our main result is a fully polynomial time approximation scheme for this general problem, which may be viewed as a multi-dimensional scaling algorithm. This directly leads to progress on some of the problems in the areas above, and a unified view of others. We explain how faster convergence of an algorithm for the same problem will allow resolving central open problems.

Our main techniques come from Invariant Theory, and include its rich *non-commutative duality theory*, and new bounds on the bitsizes of coefficients of *invariant polynomials*. They enrich the algorithmic toolbox of this very computational field of mathematics, and are directly related to some challenges in geometric complexity theory (GCT).

1998 ACM Subject Classification F.2.1 Numerical Algorithms and Problems, G.1.6 Optimization

* This work was partially supported by an NWO Veni grant no. 680-47-459 and a DFG grant BU 1371 2-2.
† A full version of the paper is available at https://arxiv.org/abs/1711.08039, [6].

9th Innovations in Theoretical Computer Science Conference (ITCS 2018).
Editor: Anna R. Karlin; Article No. 24; pp. 24:1–24:20

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

Keywords and phrases alternating minimization, tensors, scaling, quantum marginal problem, null cone, invariant theory, geometric complexity theory

Digital Object Identifier 10.4230/LIPIcs.ITCS.2018.24

1 Introduction and summary of results

Alternating minimization refers to a large class of heuristics commonly used in optimization, information theory, statistics and machine learning. It addresses optimization problems of the following general form. Given a function

$$f\colon \mathcal{X}_1 \times \cdots \times \mathcal{X}_d \to \mathbb{R} \tag{1}$$

the goal is to find a global optimum

$$\inf_{x_1 \in \mathcal{X}_1, \ldots, x_d \in \mathcal{X}_d} f(x_1, \ldots, x_d). \tag{2}$$

While both the function f and its domain may be extremely complicated, in particular non-convex, the decomposition of the domain to d *blocks* is such that the *local* optimization problems are all feasible. More precisely, for every $i \in [d]$, and for every choice of $\alpha_j \in \mathcal{X}_j$ with $j \neq i$, computing

$$\inf_{x_i \in \mathcal{X}_i} f(\alpha_1, \ldots, \alpha_{i-1}, x_i, \alpha_{i+1}, \ldots, \alpha_d)$$

is easy.

A natural heuristic in such cases is to repeatedly make local improvements to different coordinates. Namely, we start from some arbitrary vector $\alpha^{(0)}$, and generate a sequence $\alpha^{(0)}, \alpha^{(1)}, \ldots, \alpha^{(t)}, \ldots$, such that $\alpha^{(t+1)}$ is obtained by solving the above local optimization problem for some $i = i^{(t)}$, freeing up this variable while fixing all other coordinates according to $\alpha^{(t)}$. The argmin of the optimum replaces the ith coordinate in $\alpha^{(t)}$ to create $\alpha^{(t+1)}$.

There is a vast number of situations which are captured by this natural framework. For one example (we'll see more), consider the famous Lemke-Howson [28] algorithm for finding a Nash-equilibrium in a 2-player game. Here $d = 2$, and starting from an arbitrary pair of strategies, proceed by alternatingly finding a "best response" strategy for one player given the strategy of the other. This local optimization is simply a linear program that can be efficiently computed. As is well known, this algorithm always converges, but in some games it requires exponential time!

So, the basic questions which arise in some settings are under which conditions does this general heuristic converge at all, and even better, when does it converge efficiently. Seemingly the first paper to give *provable* sufficient conditions (the "5-point property") for convergence in a *general* setting was Csiszár and Tusnády [8]. An application is computing the distance (in their case, KL-divergence, but other metrics can be considered) between two convex sets in $\mathbb{R}^n$, as well as the two closest points achieving that distance. Again, the local problem (fixing one point and finding the closest to it in the other convex set) is a simple convex program that has a simple efficient algorithm. For two affine subspaces and the ℓ_2-distance, von Neumann's alternating projection method [36] is an important special case with numerous applications. As mentioned, in the past three decades numerous papers studied various variations and gave conditions for convergence, especially in cases where $d > 2$ (often called also "block-coordinate descent") – we cite only a few recent ones [46, 37, 44] which address a variety of problems and have many references. Much much fewer cases

are known in which convergence is fast, namely requires only a polynomial number of local iterations. Some examples include the recent works on matrix completion [23, 17]. Our aim is to develop techniques that expand *efficient* convergence results in the following algebraic setting, which as we shall see is surprisingly general and has many motivations.

We will consider the case of minimizing (1) with very specific domain and function, that we in turn explain and then motivate (so please bear with this necessary notation). First, each of the blocks $\mathcal{X}_i$ is a special linear group, $\mathrm{SL}_{n_i}(\mathbb{C})$ (which we will also abbreviate by $\mathrm{SL}(n_i)$), namely the invertible complex matrices of some size n_i and determinant one. Note that this domain is quite complex, as these groups are certainly not convex, and not even compact. To explain the function f to be minimized over this domain, consider the natural linear transformation (basis change) of the vector space $\mathbb{C}^{n_i}$ by matrices $A_i \in \mathrm{SL}(n_i)$. Thus, the product group $G := \mathrm{SL}(n_1) \times \mathrm{SL}(n_2) \times \cdots \times \mathrm{SL}(n_d)$ acts on tensors $X \in V := \mathrm{Ten}(n_0, n_1, n_2, \ldots, n_d) = \mathbb{C}^{n_0} \otimes \mathbb{C}^{n_1} \otimes \mathbb{C}^{n_2} \otimes \cdots \otimes \mathbb{C}^{n_d}$ of order $d + 1$[1], where (the basis change) A_i is applied to all vectors (slices, "fibers") of X along the ith dimension. Now the objective function $f = f_X$ depends on an input tensor $X \in V$; for any vector of matrices $A = (A_1, A_2, \ldots A_d) \in G$, $f_X(A)$ is defined simply as the ℓ_2-norm squared[2] of $A \cdot X$, the tensor resulting by the action of A on X. Summarizing this description, for input $X \in V$ we would like to compute (or approximate) the quantity we call *capacity* of X, denoted $\mathrm{cap}(X)$ defined by

$$\mathrm{cap}(X) := \inf_{A \in G} \|A \cdot X\|_2^2. \tag{3}$$

In particular, we would like to decide the *null-cone problem*: is the capacity of X zero or not?

While this formulation readily lends itself to an alternating minimization procedure (the local problems have an easily computed closed-form solution), the algorithms we will develop will be related to a dual optimization (scaling) problem that also has a similar form. *Our main result will be a fully polynomial time approximation scheme (FPTAS) for that dual problem!* This will also allow us to solve the null-cone problem. However we defer this discussion and turn to motivations.

Now where do such problems and this notion of capacity naturally arise? Let us list a few sources, revealing how fundamental this framework really is. Some of these connections go back over a century and some were discovered in the past couple of years. Also, some are less direct than others, and some overlap. Some of them, and others, will be elaborated on throughout the paper.

- **Combinatorial Optimization.** Computing matroid intersection (for linear matroids over $\mathbb{Q}$) easily reduces to checking zeroness of capacity for $d = 2$ for very "simple" inputs X. This was discovered in [14], who showed it remains true for an implicit version of this problem for which there was no known polynomial time algorithm. His alternating minimization algorithm (called today *operator scaling*) is efficient for such simple inputs. It also generalizes similar algorithms for such problems as perfect matchings in bipartite graphs and maximum flow in networks, where again zeroness of capacity captures the decision versions of the problems. The associated alternating minimization algorithm gives a very different and yet efficient way than is taught in algorithms texts for these classical problems (these go by the name of *matrix scaling*, originated in [38] and described explicitly in [30]).

[1] Equivalently, this means that we study n_0-tuples of tensors of order d.
[2] Namely, the sum of squares of the absolute values of the entries.

- **Non-commutative algebra.** The most basic computational problem in this field is the *word problem*, namely, given an expression over the free skew field (namely, a formula over $\mathbb{Q}$ in non-commutative variables), is it identically zero[3]. As shown in [11] this problem is reducible to testing if the capacity is 0, for $d = 2$. [11] significantly extends [14], proving that his algorithm is actually an FPTAS for *all* inputs, resolving the $d = 2$ of our general problem above, and providing the first polynomial time algorithm for this basic word problem (the best previous algorithm, deterministic or probabilistic, required exponential time)[4].
- **Analysis.** We move from efficiently testing *identities* to efficiently testing *inequalities*. The Brascamp-Lieb inequalities [4, 29] are an extremely broad class capturing many famous inequalities in geometry, probability, analysis and information theory (including Hölder, Loomis-Whitney, Shearer, Brunn-Minkowski, ...). A central theorem in this field [29] expresses feasibility, and the tightest possible constant if feasible, for every instantiation of such inequality in terms of capacity (defined somewhat differently, but directly related to ours). In [12] a direct alternating minimization algorithm[5] is given which efficiently approximates it on every input! Previous techniques to even bound capacity were completely ineffective, using compactness arguments.
- **Quantum Information Theory and Geometric Complexity Theory.** We finally move to arbitrary d. The *quantum marginal problem* (generalizing the classical *marginal problem* in probability theory) asks if a collection of marginals (namely, density matrices) on d subsystems of a given quantum system are *consistent*, namely is there a density matrix on the whole system whose partial traces on the given subsystems result in these marginals [27]. When the state of the whole system is pure and the marginals are proportional to identity matrices, each subsystem is maximally entangled with the others. It is an important task to distill entanglement, that is, given a tensor, to transform it into such a *locally maximally entangled* state by SLOCC[6] operations [2, 26]. It turns out that this amounts precisely to the capacity optimization problem, and the matrices which minimize it solve this distillation problem. One can thus view our FPTAS as achieving approximate entanglement distillation (see, e.g., [41, 43, 42]). We note that for $d = 3$, the non-zeroness of capacity captures a refinement of the *asymptotic positivity* of special *Kronecker coefficients*, a central notion in representation theory and geometric complexity theory (GCT) of [33, 5, 20]. We refer to [6] for more detail.
- **Invariant Theory.** We will be brief, and soon discuss this area at length, as it underlies most of the work in this paper. Invariant theory studies symmetries, namely group actions on sets, and their invariants. The action of our particular group G above on the particular linear space of tensors V is an example of the main object of study in invariant theory: actions of reductive[7] groups G on linear spaces V. A central notion in the (geometric) study of such actions is the *null cone*; it consists of all elements in $v \in V$ which G maps arbitrarily close to $0 \in V$. Here the connection to our problem is most direct: the null cone of the action in our framework is *precisely* the input tensors X whose capacity is 0! We stress that previous algorithms for the null cone (for general reductive actions) required

[3] This is an analog of the PIT problem in algebraic complexity, for non-commutative formulas with division.
[4] This problem was solved for finite characteristics by different methods in [22].
[5] Which can be viewed also as a reduction to the operator scaling algorithm in [11].
[6] Acronym for *Stochastic Local Operations and Classical Communication*; these are the natural actions when each of the subsystems is owned by a different party.
[7] A somewhat technical term which includes all classical linear groups.

doubly exponential time (e.g., [39, Algorithm 4.6.7]). In contrast, our algorithm (for the general framework above) decides if a given tensor is in the null cone in *singly* exponential time! We will also discuss special cases where our FPTAS achieves polynomial time. The same null-cone problem is a direct challenge among the algorithmic problems suggested by Mulmuley towards the GCT program in [32].

We now turn to discuss at more length the relevant features of invariant theory, explaining in particular the dual optimization problem that our algorithm actually solves. Then we will state our main results.

1.1 Invariant theory

Invariant theory is an area of mathematics in which computational methods and efficient algorithms are extremely developed (and sought). The reader is referred to the excellent texts [9, 39] for general background, focused on algorithms. Using invariant theory was key for the recent development of efficient algorithms for operator scaling mentioned above [11, 22], while these algorithms in turn solved basic problems in this field itself. This paper proceeds to expand the applications of invariant theory for algorithms (and computational complexity!) and to improve the efficiency of algorithms for basic problems in the field.

Invariant theory is an extremely rich theory, starting with seminal works of Cayley [7][8], to compute invariants of $\mathrm{SL}(n)$, of Hilbert [18, 19][9] and others in the 19th century. Again, we focus on linear actions of reductive groups G on vector spaces V. It turns out that in this linear setting the relevant invariants are polynomial functions of the variables (entries of vectors in V with respect to some fixed basis), and the central problem is to understand the *invariant ring*, namely all polynomials which are left invariant (compute the same value) under the action of G. Two familiar examples, which illustrate how good an understanding we can hope for, are the following:

- When $G = S_n$, the group of permutations of the coordinates of an n-vector in $V = \mathbb{F}^n$, the ring of invariants are (naturally) all *symmetric polynomials*, which is (finitely!) generated by the n elementary symmetric polynomials.
- When $G = \mathrm{SL}(n) \times \mathrm{SL}(n)$[10] acts on $n \times n$ matrices by changing basis of the rows and columns respectively, the ring of invariants is generated by one polynomial in the entries of this matrix: the *determinant*.

One of Hilbert's celebrated results in [18, 19] was that the invariant ring is *always* finitely generated. Further understanding calls for listing generators, finding their algebraic relations, making them as low degree as possible, as few as possible, and as easily computable as possible. Many new motivations for these arose from the GCT program mentioned above (and below).

The action of G naturally carves the space V into *orbits*, namely sets of the form $G \cdot v$ consisting of all images of $v \in V$ under the action of G. Understanding when are two vectors in the same orbit captures numerous natural problems, e.g., *graph isomorphism* (when is a graph in the orbit of another, under vertex renaming), *module isomorphism* (when is a vector of matrices in the orbit of another, under simultaneous conjugation), and others. Over the complex numbers $\mathbb{C}$ it is more natural to consider *orbit closures* and answer similar problems regarding membership of vectors in such orbit closures. For an important example,

[8] Which describes his ingenious algorithm, the Omega-process.
[9] In which he proves his celebrated Nullstellensatz and Finite Basis theorem.
[10] Here $\mathrm{SL}(n)$ is the group of invertible matrices of determinant 1.

the central approach of Mulmuley and Sohoni's GCT program [33] to Valiant's conjecture that VP≠VNP is to translate it to the question of whether the permanent is in the orbit closure of (a polynomially larger) determinant under a linear action on the matrix entries. Understanding orbit closures is the subject of *Geometric Invariant Theory*, starting with the seminal work of Mumford [34] in the middle of the last century. He proved that answering such questions is *equivalent* to computing invariants. That is, the orbit closures of two vectors $v, w \in V$ intersect if and only if $p(v) = p(w)$ for all invariant polynomials p.

An important special case is to understand the orbit closures that contain the zero vector $0 \in V$. Note that by the theorem above, these are all vectors $v \in V$ for which $p(v) = 0$ holds for all invariant polynomials p without constant term. This is an algebraic variety (nothing more than a zero set a system of polynomials), which is called the *null cone* of the action of G on V. Understanding the null cone is also central to geometric complexity theory from a topological viewpoint when *projectivizing* the space of orbits (making a point of each one, "modding out" by the group action).

A key to understanding the null cone is a duality theory for the group action, which may be viewed as a non-commutative analog of the duality theory for linear programming. We will elaborate on this duality theory, essential to our work, in Sections 3 and 4 and only sketch the essentials here. It is clear that a vector $v \in V$ is in the null cone if its capacity is equal to zero, and so can be "certified" by exhibiting a sequence of group elements g_t such that $\|g_t \cdot v\|$ approaches zero. How can we "certify" that a vector $v \in V$ is *not* in the null cone? The capacity formulation means that there must be some fixed $\delta > 0$ such that, for every element $w = g \cdot v$ in the orbit of v, $\|w\| \geq \delta$. Consider a point w attaining this minimum distance to 0 (assume it exists). There is a way to write down a non-commutative analog of the "Lagrange multiplier" conditions giving equations saying essentially that the derivative of the group action on w in any direction is zero. It turns out that the distance to satisfying these equations is *another, dual* optimization problem. In particular, to "certify" non-membership of v in the null cone, it suffices to exhibit a sequence of group elements g_t such that $g_t \cdot v$ approaches distance 0 from satisfying these equations. Again, we give plenty more detail on that in Sections 3 and 4.

What is remarkable is that in our setting, when G is a product of d groups as above, the new optimization has the exact same form as the original one we started with – only with a different function to minimize! Moreover, the set of equations decouples to d subsets, and optimizing each *local* one is efficiently computable. In short, this makes our new optimization problem again amenable to alternating minimization. Further, the conditions that we need to satisfy may be viewed as "scaling conditions", which for $d = 2$ can be shown to be equivalent to the matrix scaling conditions in the commutative case, and to the operator scaling conditions in the non-commutative case. Thus, these scaling algorithms in combinatorial optimization to "doubly stochastic" position naturally arise from general considerations of duality in geometric invariant theory. We will continue to borrow this name and call a tensor that satisfies the minimization conditions *"d-stochastic"*, and we quantify how well a tensor Y satisfies d-stochasticity by a distance measure denoted $\mathrm{ds}(Y)$. For an input tensor X, we then seek to compute the minimal distance to d-stochasticity, denoted $\mathrm{dds}(X)$, which is the infimum of $\mathrm{ds}(Y)$ over all Y in the orbit of X (and formally defined in Theorem 6). Thus, summarizing, with G and V as before Eq. (3), our new, dual optimization problem is

$$\mathrm{dds}(X) := \inf_{A \in G} \mathrm{ds}(A \cdot X). \tag{4}$$

In our framework, $\mathrm{ds}(Y)$ has a very simple form. Taking the quantum information theory view of the the problem (described above in the motivation), the tensor Y captures a quantum

system of d local parts; then $\mathrm{ds}(Y)$ is simply the total ℓ_2-distance squared of the d subsystems to being maximally mixed (that is, proportional to normalized identity density matrices).

We now proceed to describe our main results: the problems we address, our algorithms for them, and their complexities. More formal statements of all will appear in the technical sections.

1.2 Our results

We first describe our technical results and then the conceptual contribution of our paper.

Technical results

We fix G and V as above. While we work over the complex numbers, we will think of an input tensor $X \in V$ as an array of integers (one can consider rational numbers, but this is no gain in generality) represented in binary. The input size parameters will be the number of tensor entries $n = n_0 \times n_1 \times \cdots \times n_d$, and the maximum binary length of each entry, which will be denoted by b. So the total input length may be thought of as nb.

Recall again that we have considered two dual optimization problems[11] above for which the input is X:

$$\mathrm{cap}(X) = \inf_{A \in G} \|A \cdot X\|_2^2, \qquad \mathrm{dds}(X) = \inf_{A \in G} \mathrm{ds}(A \cdot X). \tag{3,4}$$

These lead to both exact and approximate computational problems for which we give new algorithms.

The exact *null-cone problem* we will consider is to test if the input tensor X is in the null cone of the group action G. As we discussed, X is in the null cone iff $\mathrm{cap}(X) = 0$ and iff $\mathrm{dds}(X) > 0$ (!).

We will give two different *exponential* time algorithms for this problem. We note that the previous best algorithms (which work in the greater generality of all reductive group actions) required *doubly exponential* time! These algorithms may be very useful for the study of invariants for the actions of "small", specific groups. We will discuss these algorithms soon.

The approximate problem we will consider will be to approximate $\mathrm{dds}(X)$[12]. Our approximation of dds runs in polynomial time in the input length nb and the (additive) approximation ϵ – thus a FPTAS is our main result[13], and we state it first.

▶ **Theorem 1** (Main theorem). *There is a* $\mathrm{poly}(n, b, \frac{1}{\varepsilon})$ *time deterministic algorithm (Algorithm 1) that, given a tensor* $X \in V$ *with integer coordinates of bit size bounded by* b*, either identifies that* X *is in the null cone or outputs a "scaling"* $Y \in G \cdot X$ *such that* $\mathrm{ds}(Y) < \varepsilon$.

[11] While we do not know of any precise relationship between the value of these optimization problems, the vanishing behaviour of these two problems, as explained above, is dual to each other.

[12] One could also consider the same for $\mathrm{cap}(X)$, but unlike $\mathrm{dds}(X)$, we have no faster algorithm for approximating $\mathrm{cap}(X)$ than computing it exactly.

[13] Some readers might object to our use of the term FPTAS for our algorithm since we have an additive approximation guarantee whereas the term is commonly used for multiplicative guarantees. However, note that a multiplicative guarantee for either $\mathrm{cap}(X)$ or $\mathrm{dds}(X)$ will need to determine their vanishing behaviour. This captures the null-cone problem, a polynomial time algorithm for which is the main open problem left open by this work. Also note that we don't have any guarantee on approximating $\mathrm{dds}(X)$ if $\mathrm{dds}(X) > 0$, which is also left as an interesting open problem. However, approximating $\mathrm{dds}(X)$ when it is 0 is more fundamental because of the connection to the null cone.

We present the algorithm right below in Section 3.1 and analyze it in Section 3.2. It is a realization of the alternating minimization scheme discussed before, and may be viewed as a *scaling algorithm*. Indeed, it generalizes the FPTAS given in [11] for $d = 2$, which is the operator scaling case. While the analysis is similar in spirit, and borrows many ideas from [11], we note that for higher dimensions $d > 2$, the understanding of invariant polynomials was much poorer. Thus, new ideas are needed, and in particular we give explicit[14] generators (via classical Invariant Theory techniques) of the invariant ring (for all degrees), all of which have small coefficients, and we bound the running time in a way which is *independent* of degree bounds.

Moving to the algorithms for the exact null-cone problem, as discussed, we have two exponential time ones, improving the known doubly exponential ones.

The first algorithm is a natural instantiation of the FPTAS above, with the right choice of ε. Specifically, we prove that when if X is not in the null cone, $\mathrm{dds}(X) \geq \exp(-O(n \log n))$. So picking $\varepsilon = \exp(-O(n \log n))$ suffices. This algorithm is described in Theorem 8.

Our second algorithm is much more general, and applies to the action of a product of general linear groups on *any* vector space V, not just the the set of tensors. Here, instead of giving explicit generators, we simply prove their existence, via Cayley's famous *Omega-process*. Namely, we prove that in this very general case there are always generating invariants which have both exponential degree and whose coefficients have exponential bit length. These bounds turns out to be sufficient to carry out our time analysis. The bound on the size of coefficients is (to the best of our knowledge) a new structural result in Invariant Theory, and what our analysis demonstrates is its usefulness for solving computational problems.

The corollaries of these main results to specific areas, which we described in some of the motivation bullets earlier in the introduction, are presented in the different technical sections.

Conceptual contributions

We believe that the general framework we present, namely an *algebraic framework of alternating minimization*, establishes an important connection between the fields of optimization and invariant theory. As we saw, this framework is very general and encompasses many natural problems in several fields. It exposes the importance of *symmetry* in many of these problems, and so invites the use of tools from invariant theory which studies symmetry to the analysis of algorithms. This is akin to the GCT program, which connects algebraic complexity theory with invariant theory and representation theory, exposing the importance of using symmetry to proving lower bounds.

At the same time we expose basic computational problems of invariant theory, which are in great need of better algorithms and algorithmic techniques from computer science and optimization to solve them. The advent of operating scaling already pointed out to this power in their utility of alternate minimization and scaling *numerical* algorithms to a field in which most work was *symbolic*, and we expand on this theme in this paper. But there are many other optimization methods which can be useful there, and we note only one other example, that may in particular be useful for the very null-cone problem we study. While the problem of capacity optimization is not convex under the usual Euclidean metric, it is actually *geodesically convex* [24, 35, 45] if one moves to the natural (hyperbolic) metric on the group. This opens up this problem (and related ones) to avenues of attack from techniques of classical optimization (gradient and other descent methods, interior point methods, etc.)

[14] The generators are explicitly defined but may not be efficiently computable by algebraic circuits.

when generalized and adapted to such metrics. We believe that the area of geodesic convex optimization, which itself in its nascent stages (from an algorithmic standpoint), is likely to have a significant impact on algorithmic invariant theory. See more on algorithmic techniques for geodesic convexity in the work of [47] and the references therein.

2 Null-cone problem as an optimization problem

In the introduction, we defined the *null cone* as the set of tensors X such that the zero vector $0 \in V$ lies in the orbit closure $\overline{G \cdot X}$, i.e., there exists a sequence of group elements $A^{(1)}, A^{(2)}, \ldots$ sending the vector to zero, $\lim_{j\to\infty} A^{(j)} \cdot X = 0$. Thus the *null-cone problem* amounts to the optimization problem (3): A tensor $X \in V = \mathrm{Ten}(n_0, n_1, \ldots, n_d)$ is in the null cone if and only if the *capacity* is equal to zero, namely:

$$\mathrm{cap}(X) = \inf_{A\in G} \|A \cdot X\|_2^2 = \min_{Y\in\overline{G\cdot X}} \|Y\|_2^2 = 0.$$

▶ Remark. We are denoting the above optimization problem by $\mathrm{cap}(X)$, short for *capacity*, to remain consistent with similar notions defined in previous papers like *[16, 14, 15, 11]*. In most of these cases, the capacity notion that they consider is also looking at the optimization problem: "minimum ℓ_2-norm in an orbit closure" for specific group actions. As phrased in these papers, they might look different (e.g., they involve determinants in place of ℓ_2 norms) but there is a tight connection between the two via the AM-GM inequality (i.e., the inequality of arithmetic and geometric means) in one direction and a single alternating minimization step in the other direction.

A fundamental theorem of Hilbert [19] and Mumford [34] gives an alternative characterization of the null cone. It states that the null cone is precisely the set of tensors X on which all invariant polynomials P (without constant term) vanish. This is the starting point to the field of Geometric Invariant Theory.

It is instructive to see why $P(X) = 0$ must hold for any tensor X in the null cone: Since P is G-invariant, $P(Y) = P(X)$ for all Y in the G-orbit of X, and, by continuity, also for all Y in the closure of the orbit. If the tensor X is in the null cone, then $Y = 0 \in V$ is in its closure. But then $P(X) = P(0) = 0$, since the polynomial P has no constant term. In [6] we derive a more subtle, *quantitative* version of this observation. It will be a fundamental ingredient to the analysis of our algorithm.

Since the group $G = \mathrm{SL}(n_1) \times \cdots \times \mathrm{SL}(n_d)$ consists of tuples $A = (A_1, \ldots, A_d)$, an alternating minimization algorithm suggests itself for solving the optimization problem $\mathrm{cap}(X)$: Fix an index $i \in [d]$ and optimize only over a single A_i, leaving the $(A_j)_{j\neq i}$ unchanged. This gives rise to an optimization problem of the following form:

$$\inf_{A_S\in\mathrm{SL}(n_S)} \|(A_S \otimes I_{n_T}) \cdot Y\|_2^2$$

Here $n_S = n_i$, $n_T = \prod_{j\neq i} n_j$ and $Y = A \cdot X$. This optimization problem has a closed form solution in terms of the following quantum mechanic analog of the marginals of a probability distribution:

▶ **Definition 2** (Quantum marginal)**.** Given a matrix ρ acting on $\mathcal{H}_S \otimes \mathcal{H}_T$, where $\mathcal{H}_S$ and $\mathcal{H}_T$ are Hilbert spaces of dimensions n_S and n_T, respectively (e.g., $\mathcal{H}_S = \mathbb{C}^{n_S}$ and $\mathcal{H}_T = \mathbb{C}^{n_T}$). Then there is a unique matrix ρ_S acting on $\mathcal{H}_S$, called the *quantum marginal* (or reduced density matrix) of S, such that

$$\mathrm{tr}\left[(M_S \otimes I_T)\,\rho\right] = \mathrm{tr}\left[M_S \rho_S\right] \tag{5}$$

for every M_S acting on $\mathcal{H}_S$.

This point of view gives rise to an important interpretation of our result [6].

It can be easily seen that if ρ is positive semidefinite (PSD) then ρ_S is PSD, and that it has the same trace as ρ. It is given by the following explicit formula

$$\rho_S = \sum_{k=1}^{n_T} (I_S \otimes e_k)^\dagger \, \rho \, (I_S \otimes e_k) \, .$$

Here the e_k's are the elementary column vectors of dimension n_T.

Now observe that

$$\|(A_S \otimes I_T) \cdot Y\|_2^2 = \mathrm{tr}\left[\left(A_S^\dagger A_S \otimes I_T\right) YY^\dagger\right] .$$

Let ρ_i be the partial trace of $\rho = YY^\dagger$ obtained by tracing out all the Hilbert spaces except the i^{th} one, and rename $A_i = A_S$. Then, using Eq. (5),

$$\mathrm{tr}\left[\left(A_S^\dagger A_S \otimes I\right) YY^\dagger\right] = \mathrm{tr}\left[A_i^\dagger A_i \rho_i\right] .$$

Hence we are left with the optimization problem:

$$\inf_{A_i \in \mathrm{SL}(n_i)} \mathrm{tr}\left[A_i A_i^\dagger \rho_i\right] . \tag{6}$$

We can see by the AM-GM inequality that the optimum of the program (6) is $n_i \det(\rho_i)^{1/n_i}$, which is achieved for $A_i = \det(\rho_i)^{1/2n_i} \rho_i^{-1/2}$ (if ρ_i is not invertible then we define the inverse on the support of ρ_i; the infimum will at any rate be zero).

We thus obtain the core of an alternating minimization algorithm for the capacity $\mathrm{cap}(X)$. At each step t, select an index $i \in [d]$, compute the quantum marginal ρ_i of the current tensor $X^{(t)}$, and update by performing the following *scaling step*:

$$X^{(t+1)} \leftarrow \det(\rho_i)^{1/2n_i} \rho_i^{-1/2} \cdot X^{(t)}. \tag{7}$$

Here and in the following we use the abbreviation

$$A_i \cdot Y := (I_{n_0} \otimes I_{n_1} \otimes \ldots \otimes I_{n_{i-1}} \otimes A_i \otimes I_{n_{i+1}} \otimes \ldots \otimes I_{n_d}) \cdot Y,$$

where we act nontrivially on the i-th tensor factor only.

We will analyze essentially this algorithm, augmented by an appropriate method for selecting the index i at each step and a stopping criterion (see Algorithm 1 in the next section).

3 Noncommutative duality and the tensor scaling algorithm

To obtain a scaling algorithm with rigorous guarantees, we will now derive a dual formulation of the optimization problem $\mathrm{cap}(X)$. This will achieved by a theorem from Geometric Invariant Theory, due to Kempf and Ness, which can be understood as part of a noncommutative duality theory.

We briefly return to the setting of a general action of a group G on a vector space V to explain this result. Fix a vector $v \in V$ and consider the function $f_v(g) := \|g \cdot v\|_2^2$. The *moment map* at v, $\mu(v)$, measures how much $f_v(g)$ changes when we perturb g around the identity. So $\mu(v)$ is just the derivative of the function $f_v(g)$ at the identity element (in a precise sense which we do not need to define here). Now suppose v is not in the null cone. Then there exists a *nonzero* vector $w \in \overline{G \cdot v}$ of minimal norm in the orbit closure. Since

the norm is minimal, perturbing w by the action of group elements g close to the identity element does not change the norm to first order. Hence $\mu(w) = 0$. To summarize, if v is not in the null cone, then there exists $0 \neq w \in \overline{G \cdot v}$ such that $\mu(w) = 0$. This condition can be understood as a vast generalization of the notion of "doubly stochastic", which one might be tempted to call *"G-stochastic"*.

A remarkable theorem due to Kempf and Ness [24] asserts that this is not only a necessary but in fact also sufficient condition, i.e., v is not in the null cone if and only if there exists $0 \neq w \in \overline{G \cdot v}$ such that $\mu(w) = 0$. This is a *duality theorem* and (along with the Hilbert-Mumford criterion discussed below) should be thought of as a non-commutative version of linear programming duality, which arises when the group G is commutative (see Section 3 below, [40], and [6]). For a detailed discussion of moment maps, we refer the reader to [45, 42].

We now return to our setting where $G = \mathrm{SL}(n_1) \times \cdots \times \mathrm{SL}(n_d)$ acts on tensors in $V = \mathrm{Ten}(n_0, n_1, \ldots, n_d)$. We first define the notion of a scaling and then instantiate the Kempf-Ness theorem in the situation at hand:

▶ **Definition 3** (Scaling). A tensor $Y \in \mathrm{Ten}(n_0, n_1, \ldots, n_d)$ is called a (*tensor*) *scaling* of $X \in \mathrm{Ten}(n_0, n_1 \ldots, n_d)$ if $Y \in G \cdot X$.

This notion of scaling generalizes the notions of operator scaling [14, 11] and matrix scaling [38]. It is immediate from the definition of the capacity that a tensor X is in the null cone if and only if any of its scalings Y lies in the null cone.

We now state the Kempf-Ness theorem for tensors:

▶ **Theorem 4** (Duality, special case of [24]). *A tensor $X \in \mathrm{Ten}(n_0, \ldots, n_d)$ is* not *in the null cone if and only if there exists $Y \in \overline{G \cdot X}$ such that the quantum marginals $\rho_1, \ldots, \rho_d$ of $\rho = YY^\dagger / Y^\dagger Y$ of the last d tensor factors are given by $\rho_i = I_{n_i}/n_i$.*

We note that the trace of $\rho = YY^\dagger / Y^\dagger Y$ is one. This normalization is very natural from the quantum viewpoint. In quantum theory, PSD matrices of unit trace describe quantum states of multi-partite systems and the quantum marginals describe the state of subsystems. The condition that ρ_i is proportional to the identity means that the state of the k^{th} subsystem is maximally mixed or, for ρ as above, that the k^{th} system is maximally entangled with the rest. We discuss applications of our result to quantum information theory in [6].

The condition that each ρ_i is proportional to the identity matrix generalizes the notion of "doubly stochastic", which arises in the study of the left-right action/operator scaling [14, 11]. We will refer to it as *"d-stochastic"*, which we define formally below.

▶ **Definition 5** (d-stochastic). A tensor $Y \in \mathrm{Ten}(n_0, \ldots, n_d)$ is called *d-stochastic* if the quantum marginals of $\rho = YY^\dagger / Y^\dagger Y$ of the last d tensor factors are given by $\rho_i = I_{n_i}/n_i$ for $i \in [d]$.

More explicitly, we can state the condition that $\rho_i = I_{n_i}/n_i$ as follows: We want that the slices of the tensor Y in the k^{th} direction are pairwise orthogonal and that their norm squares are equal to $1/n_i$. That is, for each $i \in [d]$ consider the order d tensors $Y^{(1)}, \ldots, Y^{(n_i)} \in \mathrm{Ten}(n_0, \ldots, n_{i-1}, n_{i+1}, \ldots, n_d)$ defined as

$$Y^{(j)}(j_0, \ldots, j_{i-1}, j_{i+1}, \ldots, j_d) := Y(j_0, \ldots, j_{i-1}, j, j_{i+1}, \ldots, j_d);$$

then the following should hold:

$$\langle Y^{(j)}, Y^{(j')} \rangle = \frac{1}{n_i} \delta_{j,j'}.$$

Here $\delta_{i,j}$ is the Kronecker delta function and the inner product is the Euclidean inner product of tensors.

Theorem 4 implies that X is *not* in the null cone if and only if we can find scalings Y whose last d quantum marginals are arbitrarily close to I_{n_i}/n_i. We can therefore rephrase the null-cone problem as *another, dual* optimization problem, where we seek to minimize the distance of the marginals to being maximally mixed. This is captured by the following definitions:

▶ **Definition 6** (Distance to d-stochasticity)**.** Let $Y \in \mathrm{Ten}(n_0, n_1, \ldots, n_d)$ be a tensor and $\rho = YY^\dagger / Y^\dagger Y$, with quantum marginals $\rho_1, \ldots, \rho_d$ on the last d systems. Then we define the *distance to d-stochasticity* $\mathrm{ds}(Y)$ as the (squared) distance between the marginals and d-stochastic marginals,

$$\mathrm{ds}(Y) := \sum_{i=1}^{d} \left\| \rho_i - \frac{I_{n_i}}{n_i} \right\|_F^2,$$

where $\|M\|_F := (\operatorname{tr} M^\dagger M)^{1/2}$ is the Frobenius norm. Following Eq. (4), we further define the *minimal distance to d-stochasticity* as

$$\mathrm{dds}(X) := \inf_{A \in G} \mathrm{ds}(A \cdot X).$$

Using this language, Theorem 4 states that X is in the null cone if and only if the minimal distance to d-stochasticity is nonzero. We summarize the duality between the two optimization problems:

$$X \text{ is in the null cone} \quad \Longleftrightarrow \quad \mathrm{cap}(X) = 0 \quad \Longleftrightarrow \quad \mathrm{dds}(X) > 0 \tag{8}$$

▶ Remark. According to Eq. (8), for any tensor X *exactly one* of the following two statements is true: X is in the null cone ($\mathrm{cap}(X) = 0$) or its orbit closure contains a d-stochastic tensor ($\mathrm{dds}(X) = 0$). Such dichotomies are well-known from the duality theory of linear programming (Farkas' lemma, Gordan's lemma, the duality between membership in a convex hull and the existence of separating hyperplanes, etc.). In the case of the (commutative) group of diagonal matrices, one recovers precisely these linear programming dualities from the general noncommutative framework.

3.1 The tensor scaling algorithm

If X is *not* in the null cone then there exist scalings Y such that $\mathrm{ds}(Y)$ is arbitrarily small. The main technical result of this paper is an algorithm that finds such scalings for any fixed $\varepsilon > 0$. It is a generalization of the results obtained for matrix and operator scaling in [30, 16, 14, 11]. Recall that we use n to denote the total number of coordinates of the tensor, $n = n_0 \cdots n_d$.

▶ **Theorem 7.** *Let $X \in \mathrm{Ten}(n_0, \ldots, n_d)$ be a (nonzero) tensor whose entries are integers of bitsize no more than b. Let $\ell = \min_{i \geq 1} n_i$ and suppose that $\varepsilon \leq d/(\max_{i \geq 1} n_i^2)$. Then Algorithm 1 with $T \geq \frac{18 \ln 2}{\ell \varepsilon} d\,(b + \log n)$ iterations either identifies that X is in the null cone or outputs a scaling $Y \in G \cdot X$ such that $\mathrm{ds}(Y) \leq \varepsilon$.*

To motivate Algorithm 1, note that, for every $i \in [d]$, we would like the quantum marginal of the i^{th} system to be proportional to the identity matrix. Suppose the quantum marginal on the i^{th} system is ρ_i. Then by acting on this system by any $A_i \in \mathrm{SL}(n_i)$ proportional to $\rho_i^{-1/2}$

Algorithm 1 Scaling algorithm for the null-cone problem.

Input: A tensor X in $\mathrm{Ten}(n_0, n_1, \ldots, n_d)$ with integer entries (specified as a list of entries, each encoded in binary, with bit size $\leq b$).

Output: Either the algorithm correctly identifies that X is in the null cone, or it outputs a scaling Y of X such that $\mathrm{ds}(Y) \leq \varepsilon$.

Algorithm:

1. If any of the quantum marginals of X is singular, then output that the tensor X is in the null cone and return. Otherwise set $Y^{(1)} = X$ and proceed to step 2.
2. For $t = 1, \ldots, T = \mathrm{poly}(n, b, 1/\varepsilon)$, repeat the following:
   - Compute the quantum marginals ρ_i of $Y^{(t)}Y^{(t),\dagger}/Y^{(t),\dagger}Y^{(t)}$ and find the index $i \in [d]$ for which $\left\|\rho_i - \frac{I_{n_i}}{n_i}\right\|_F^2$ is largest. If $\left\|\rho_i - \frac{I_{n_i}}{n_i}\right\|_F^2 < \varepsilon/d$, output $Y^{(t)}$ and return.
   - Otherwise, set $Y^{(t+1)} \leftarrow \det(\rho_i)^{1/2n_i}\rho_i^{-1/2} \cdot Y^{(t)}$.
3. Output that the tensor X is in the null cone.

we can precisely arrange for the quantum marginal on i^{th} system to be proportional to the identity matrix[15]. (But this might disturb the quantum marginals on the other systems!) An appropriate choice of A_i that ensures that it has determinant one is $A_i = \det(\rho_i)^{1/2n_i}\rho_i^{-1/2}$. Thus we find that the alternating minimization heuristics in Eq. (7) that we previously derived for $\mathrm{cap}(X)$ is equally natural from the perspective of the dual problem of minimizing $\mathrm{dds}(X)$! Remarkably, iterating this operation in an appropriate order of subsystems i does ensure that *all* the quantum marginals get arbitrarily close to the normalized identity matrices – provided that X is not in the null cone. (If X is in the null cone, then it will lead to a sequence of scalings of norm arbitrarily close to zero, certifying that X is in the null cone.)

▶ Remark (Rational Entries). Notice that we required our input tensor X to have integer entries. In general, an input tensor X could have rational entries. If this is the case, we can simply multiply X by the denominators to obtain an integral tensor X', on which we can then apply our algorithm above. Since multiplying a tensor by a nonzero number does not change the property of being in the null cone, our algorithm will still be correct. The following remarks discuss the changes in bit complexity of our algorithm. In this case, the size of X' is at most $b \cdot n$, and and therefore the bound for the number of iterations T is still $\mathrm{poly}(n, b, 1/\varepsilon)$, as we wanted.

When analyzing bit complexity, Algorithm 1 is not directly applicable, since it computes inverse square roots of matrices. Even if the entries of the square roots were always rational, the algorithm would also iteratively multiply rationals with very high bit complexity. We can handle these numerical issues by truncating the entries of the scalings $Y^{(t)}$ of X, as well as the quantum marginals ρ_k, to $\mathrm{poly}(n, b, \frac{1}{\varepsilon})$ many bits after the decimal point. Then, in a similar way as done in [14, 11], we can prove that there is essentially no change in the convergence required in the number of iterations T. Since each arithmetic operation will be done with numbers of $\mathrm{poly}(n, b, \frac{1}{\varepsilon})$ many bits, this truncation ensures that we run in polynomial time. As a consequence, we obtain our main theorem, which we had already announced in the introduction:

▶ **Theorem 1** (Main theorem). *There is a* $\mathrm{poly}(n, b, \frac{1}{\varepsilon})$ *time deterministic algorithm (Algorithm 1) that, given a tensor* $X \in V$ *with integer coordinates of bit size bounded by* b*, either identifies that* X *is in the null cone or outputs a "scaling"* $Y \in G \cdot X$ *such that* $\mathrm{ds}(Y) < \varepsilon$.

[15] It is not hard to see that if X is not in the null cone then ρ_i is invertible [6]

We remark that for $d = 2$, our algorithm in essence reduces to the algorithm of [16, 14, 15, 11]; for $n_0 = 1$, it refines the algorithm proposed previously in [41] without a complexity analysis. Indeed, Theorem 1 is a generalization of the main theorem of [11] on operator scaling (or the left-right action). There it was enough to take $\varepsilon = 1/n$ to be sure that the starting tensor is not in the null cone.

In our much more general setting it is still true that there exists some $\varepsilon = \varepsilon(n) > 0$ such that $\mathrm{ds}(Y) > \varepsilon$ for any Y in the null cone. Unfortunately, in our general setting, ε will be exponentially small (see Theorem 16 in the next section). Therefore, Theorem 1 with this choice of ε gives an exponential time algorithm for deciding the null-cone problem:

▶ **Theorem 8** (Null-cone problem). *There is a* $\mathrm{poly}(b)\exp(O(n\log n))$ *time deterministic algorithm (Algorithm 1 with* $\varepsilon = \exp(-O(n\log n))$*) that, given a tensor* $X \in \mathrm{Ten}(n_0, n_1, \ldots, n_d)$ *with integer coordinates of bit size at most* b*, decides whether* X *is in the null cone.*

Thus our algorithm does not solve the null-cone problem in polynomial time for general tensor actions, and this remains an excellent open question!

Nevertheless, Theorem 1 as such, with its promise that $\mathrm{ds}(Y) < \varepsilon$, is of interest beyond merely solving the null-cone problem. It has applications to quantitative notions of instability in geometric invariant theory, to a form of multipartite entanglement distillation in quantum information theory, and to questions about the slice rank of tensors, which underpinned recent breakthroughs in additive combinatorics. We discuss these in Section 4 and [6].

3.2 Analysis sketch

To analyze our algorithm and prove Theorem 7, we follow a three-step argument similar to the analysis of the algorithms for matrix scaling and operator scaling in [30, 14, 11] once we have identified the appropriate potential function and distance measure. In the following we sketch the main ideas and we refer to [6] for all technical details.

As potential function, we will use the *norm squared* of the tensor, denoted $f(Y) = \|Y\|_2^2$, and the distance measure is $\mathrm{ds}(Y)$ defined above in Theorem 6. Note that these the two functions are exactly dual to each other in our noncommutative optimization framework (see Eqs. (3), (4) and (8))!

The following two properties of the capacity will be crucial for the analysis:

A. $\mathrm{cap}(X) \geq 1/(n_1 \cdots n_d)^2$ if $X \in \mathrm{Ten}(n_0, \ldots, n_d)$ is a tensor with integral entries that is not in the null cone ([6]).

B. $\mathrm{cap}(X)$ is invariant under the action of the group $\mathrm{SL}(n_1) \times \cdots \times \mathrm{SL}(n_d)$.

Using the above properties, a three-step analysis follows the following outline:

1. An upper bound on $f(Y^{(1)}) = f(X)$ from the input size.

2. As long as $\mathrm{ds}(Y) \geq \varepsilon$, the norm squared decreases by a factor $2^{-\frac{\ell\varepsilon}{6d\ln 2}}$ in each iteration: $f(Y^{(t+1)}) \leq 2^{-\frac{\ell\varepsilon}{6d\ln 2}} f(Y^{(t)})$, where we recall that $\ell = \min_{i\geq 1} n_i$.

3. A lower bound on $f(Y^{(t)})$ for all t. This follows from properties A and B above.

The above three steps imply that we must achieve $\mathrm{ds}(Y^{(t)}) < \varepsilon$ for some $t \in [T]$.

Step 1 is immediate and step 2 follows from the G-invariance of the capacity (property B) and a quantative form the AM-GM inequality.

Step 3, or rather the lower bound on $\mathrm{cap}(X)$ (property A) is the technically most challenging part of the proof. It is achieved by quantifying the basic argument given on p. 9. The main tool required to carry out this analysis is the existence of a set of invariant polynomials with "small" integer coefficients that spans the space of invariant polynomials. We do so by giving an "explicit" construction of such a set via Schur-Weyl

duality [6]. Remarkably, we do not need to use the known exponential bounds on the degree of generating invariants in [10].

We give an alternative lower bound on $\mathrm{cap}(X)$ using Cayley's Omega-process in [6]. This proof is more general (it applies to arbitrary actions of product of SL's) but gives a weaker bound (although sufficient to yield a polynomial time analysis for the algorithm). Here we do need to rely on recent exponential bounds on the degree of generating invariants by Derksen.

The size of the integer coefficients of a spanning set of invariant polynomials is an interesting invariant theoretic quantity that appears to not have been studied in the invariant theory literature. It is crucial for our analysis, and we believe it could be of independent interest in computational invariant theory.

4 Hilbert-Mumford criterion and quantifying instability

In this section, we explain a special case of the Hilbert-Mumford criterion, which allows us to characterize tensors that lie in the null cone in terms of simple, one-dimensional subgroups of our group $\mathrm{SL}(n_1) \times \cdots \times \mathrm{SL}(n_d)$. Moreover, we define the instability (Theorem 11), which quantifies how fast a tensor in the null cone can be sent to the zero vector $0 \in V$ by the group action. The instability is an important quantity in invariant theory and we present an algorithm for the $(0, \varepsilon)$-gap problem (Theorem 15).

We again consider briefly the general setting of a group G acting on a vector space V. We know from Section 2 that a vector is in the null cone if and only if its orbit closure contains the zero vector, $0 \in \overline{G \cdot v}$. The *Hilbert-Mumford criterion* [19, 34] shows that it suffices to consider certain one-dimensional (and hence in particular commutative) subgroups of G. More precisely, recall that a *one-parameter subgroup* (*1-PSG*) of G is an algebraic group homomorphism $\lambda : \mathbb{C}^* \to G$, i.e., an algebraic map that satisfies $\lambda(zw) = \lambda(z)\lambda(w)$. (For example, any 1-PSG of $\mathbb{C}^*$ is of the form $z \mapsto z^k$ for some integer k.) Then the Hilbert-Mumford criterion states that a vector v is in the null cone if and only if there exists a 1-PSG λ of G such that $\lim_{z \to 0} \lambda(z) \cdot v = 0$.

A familiar example appears when the group $G = \mathrm{SL}(n)$ acts by left multiplication on a single matrix X. In this case, the null cone consists of the singular matrices and the Hilbert-Mumford criterion tells us that if X is singular, then there is a 1-PSG given by $\lambda(z) = B^{-1} \operatorname{diag}(z^{a_1}, \ldots, z^{a_n}) B$ which drives X to zero. It is easy to see that in this case such one-parameter subgroup can be found by taking B to be a matrix in $\mathrm{SL}(n)$ which makes X upper triangular (through row eliminations and permutations) with its last row being all zeros, and that the exponents a_i can be taken such that $a_1 = \cdots = a_{n-1} = 1$ and $a_n = 1 - n$.

We now return to our setup, where $G = \mathrm{SL}(n_1) \times \cdots \times \mathrm{SL}(n_d)$. Here any 1-PSG λ is of the following form:

$$\lambda(z) = \left(B_1^{-1} \operatorname{diag}(z^{a_{1,1}}, \ldots, z^{a_{1,n_1}}) B_1, \ldots, B_d^{-1} \operatorname{diag}(z^{a_{d,1}}, \ldots, z^{a_{d,n_d}}) B_d \right), \tag{9}$$

where $B_1, \ldots, B_d$ are invertible matrices (B_i of dimension $n_i \times n_i$) and $(a_{i,j})$ is a tuple of integers in

$$\Gamma := \left\{ (a_{i,j})_{i \in [d], j \in [n_i]} \;\middle|\; a_{i,j} \in \mathbb{Z}, \sum_{j=1}^{n_i} a_{i,j} = 0 \text{ for } i \in [d] \right\}. \tag{10}$$

Intuitively, the matrices B_i are a change of basis after which the action of G becomes reduced to an action of diagonal groups (similar to actions arising in the study of the classical matrix or tensor scalings).

We want to understand what it means for a 1-PSG λ and a tensor X that $\lim_{z\to 0} \lambda(z)\cdot X = 0$. For this, we write the tensor in the basis corresponding to $B = (B_1, \ldots, B_d)$, i.e., $Y = B\cdot X$. We define the *support* of Y as

$$\operatorname{supp}(Y) := \left\{(j_1, \ldots, j_d) \in [n_1] \times \cdots \times [n_d] \;\middle|\; \exists j_0 \text{ s.t. } Y_{j_0, j_1, \ldots, j_d} \neq 0\right\}.$$

Thus $(j_1, \ldots, j_d)$ is in the support of Y iff at least one of the slices $Y^{(i)}$ in the 0^{th} direction of the tensor Y has a nonzero entry at this position. Now note that

$$(B \cdot \lambda(z) \cdot X)_{j_0, j_1 \ldots, j_d} = z^{\sum_{i=1}^{d} a_{i,j_i}} Y_{j_0, j_1, \ldots, j_d}. \tag{11}$$

It follows that $\lim_{z\to 0} \lambda(z)\cdot X = 0$ is equivalent to the support $\operatorname{supp}(Y)$ being deficient in the following sense:

▶ **Definition 9** (Deficiency)**.** We call a subset $S \subseteq [n_1] \times \cdots \times [n_d]$ *deficient* if there exists $(a_{i,j}) \in \Gamma$ such that $\forall\, (j_1, \ldots, j_d) \in S \quad \sum_{i=1}^{d} a_{i,j_i} > 0$.

We note that in the case $d = 2$, a subset $S \subseteq [n] \times [n]$ is deficient if and only if the bipartite graph corresponding to S does not admit a perfect matching, as can be proved via Hall's matching theorem. For $d > 2$, we do not know of such a clean combinatorial characterization, although the characterization above is by a linear program, and therefore is efficiently testable.

In the Hilbert-Mumford criterion, one can do slightly better, namely restrict attention to the one-parameter subgroups compatible with a maximally compact subgroup of the group G. What this means in our case is that we will be able to take the matrices $B_1, \ldots, B_d$ to be unitary matrices (see Theorem 12)! We can thus summarize the statement of the Hilbert-Mumford criterion as follows:

▶ **Proposition 10** (Special case of [19, 34])**.** *A tensor $X \in \operatorname{Ten}(n_0, \ldots, n_d)$ is in the null cone of the natural action of the group $G = \operatorname{SL}(n_1) \times \cdots \times \operatorname{SL}(n_d)$ iff there exist unitary $n_i \times n_i$ matrices U_i, $i \in [d]$, such that the support $\operatorname{supp}(Y)$ of the tensor $Y = (U_1, \ldots, U_d)\cdot X$ is deficient.*

For completeness, we provide a proof of the criterion in [6].

▶ Remark. Deficiency can also be understood in terms of the null cone of the action of the subgroup $T \subseteq \operatorname{SL}(n_1) \times \cdots \times \operatorname{SL}(n_d)$ formed by tuples of diagonal matrices. Indeed, when we fixed B but varied the $a_{i,j}$, we were precisely varying over the 1-PSGs of T. Thus, a tensor $Y \in \operatorname{Ten}(n_1, \ldots, n_d)$ is in the null cone for the action of the diagonal subgroup if and only if $\operatorname{supp}(Y)$ is deficient. (This follows from the Hilbert-Mumford criterion for the commutative group T, or directly from Farkas' lemma.) Thus Proposition 10 is a special case of a general reduction principle in geometric invariant theory, from actions of non-commutative groups to commutative groups up to a basis change.

In geometric invariant theory [34], vectors in the null cone are also referred to as *unstable*. The Hilbert-Mumford criterion suggests a natural way of quantifying the instability of a vector: instead of merely asking whether there exists a 1-PSG that sends the vector to zero, we may measure the (suitably normalized) rate at which the vector is sent to zero in Eq. (11). This rate, also known as *Mumford's numerical function*, takes the following form for the tensor action that we are studying:

▶ **Definition 11** (Deficiency, instability)**.** Given a set $S \subseteq [n_1] \times \cdots \times [n_d]$, we define its *deficiency* as

$$\operatorname{def}(S) := \max_{(a_{i,j}) \in \Gamma} \frac{\min_{(j_1, \ldots, j_d) \in S} \left(\sum_{i=1}^{d} a_{i,j_i}\right)}{\sqrt{\sum_{i=1}^{d} \sum_{j=1}^{n_i} a_{i,j}^2}}.$$

where we recall that the set Γ was defined in Eq. (10). We then define the *instability* of a tensor $X \in \mathrm{Ten}(n_0, \dots, n_d)$ by

$$\mathrm{instability}(X) := \max_{\substack{U=(U_1,\dots,U_d) \text{ tuple} \\ \text{of unitary matrices}}} \mathrm{def}(\mathrm{supp}(U \cdot X)),$$

The instability can also be defined using general invertible matrices:

▶ **Lemma 12.** *For any $X \in \mathrm{Ten}(n_0, \dots, n_d)$, we have that*

$$\mathrm{instability}(X) = \max_{\substack{B=(B_1,\dots,B_d) \text{ tuple} \\ \text{of invertible matrices}}} \mathrm{def}(\mathrm{supp}(B \cdot X)).$$

In particular, $X \mapsto \mathrm{instability}(X)$ is a G-invariant function.

In our case, this follows from the QR decomposition, and we give a succinct proof in [6]. Clearly, a subset S is deficient if and only if its deficiency is positive, $\mathrm{def}(S) > 0$. And by the Hilbert-Mumford criterion, a tensor X is in the null cone if and only if its instability is positive, $\mathrm{instability}(X) > 0$. This suggests a natural relaxation of the null-cone problem:

▶ **Problem 13.** *For $\varepsilon > 0$, the problem* ε-instability *is defined as follows: Given a tensor $X \in \mathrm{Ten}(n_0, \dots, n_d)$ with integer coordinates such that either*
1. *X is not in the null cone (i.e., $\mathrm{instability}(X) \leq 0$), or*
2. *$\mathrm{instability}(X) \geq \varepsilon$,*

decide which is the case.

We will now show that the ε-instability problem can be solved as a consequence of our main Theorem 1. Importantly, the instability of a tensor is tightly connected to the distance from d-stochasticity, as defined in Theorem 6.

▶ **Lemma 14** (Special case of [35, Lemma 3.1, (iv)])**.** *For all tensors $X \in \mathrm{Ten}(n_0, \dots, n_d)$, $\mathrm{instability}(X) \leq \sqrt{\mathrm{dds}(X)}$.*

In fact, the inequality in Theorem 14 is tight for tensors in the null cone (e.g., [13, Corollary 11.2]), but we will not need this here (if X is not in the null cone then $\mathrm{instability}(X) \leq 0$ while $\mathrm{ds}(X) = 0$). We obtain the following corollary of Theorem 1.

▶ **Theorem 15.** *There is $\mathrm{poly}\left(n, b, \frac{1}{\varepsilon}\right)$ time deterministic algorithm (Algorithm 1) for the ε-instability problem (Problem 13) for tensors with entries of bit size bounded by b.*

The algorithm is obtained by running Algorithm 1 and outputting "X is not in the null cone" if Algorithm 1 produces a scaling Y with $\mathrm{ds}(Y) < \varepsilon$ and otherwise outputting "$\mathrm{instability}(X) \geq \varepsilon$".

Thus ε-instability can be decided efficiently if $\varepsilon = \Omega(1/\mathrm{poly}(n))$. Unfortunately, there may well exist tensors X which are in the null cone and whose instability is exponentially small. However, the instability cannot be worse than exponentially small, as we prove in [6]:

▶ **Lemma 16.** *Suppose a tensor $X \in \mathrm{Ten}(n_0, \dots, n_d)$ is in the null cone. Then $\sqrt{\mathrm{dds}(X)} \geq \mathrm{instability}(X) = \exp(-O(n \log n))$, where we recall that $n = n_0 \cdots n_d$.*

This begs the question: are there natural examples where we can apply the above algorithm in Theorem 15? One set of examples comes from the notion of slice rank of tensors – discovered and widely used in the recent breakthroughs in additive combinatorics regarding cap sets and other combinatorial objects (see, e.g., [3]). We discuss this in [6].

5 Conclusion and open problems

This paper continues the recent line of works studying the computational complexity of problems in invariant theory [31, 11, 21, 22, 20, 5, 32, 1]. There are many beautiful algorithms known in invariant theory [9, 39], but most of them come without a complexity analysis or will have at least exponential runtime. Designing efficient algorithms for problems in invariant theory is important for the GCT program [31, 32], and we believe that viewing invariant theory from the computational lens will provide significant new structural insights. Several open problems arise from this work. We mention the most interesting ones:

1. Is there a polynomial time algorithm for deciding the null-cone problem for the tensor actions that we study? It would suffice to give an analog of the algorithm in Theorem 1 with running time $\mathrm{poly}(n, b, \log(1/\varepsilon))$. One might wonder if known optimization techniques are sufficient to yield this runtime guarantee. While the optimization problem in Eq. (3) is not convex, it is nevertheless known to be *geodesically* convex [24, 35, 45] (roughly speaking one needs to move from the usual Euclidean geometry to the geometry of the group to witness convexity). The theory of geodesically convex optimization is just starting to emerge (see, e.g., [47] and the references therein) and it is an open problem whether there are computationally efficient analogs of the ellipsoid algorithm and interior point methods (algorithms that in Euclidean convex optimization guarantee $\mathrm{poly}(\log(1/\varepsilon))$ runtime).
2. Is there a polynomial time algorithm for the null-cone problem for more general group actions? As mentioned before, there is a natural notion of *"G-stochasticity"* using moment maps provided by the noncommutative duality theory. A first important goal is to design an algorithm that achieves the analog of our Theorem 1 (i.e., getting to a point in the orbit where the "distance" to satisfying G-stochasticity is at most ε). While it is not clear how far the alternating minimization approach can be taken, Kirwan's gradient flow [25] seems to be a promising alternative first proposed in [43, 42].
3. Is there a polynomial time algorithm for the one-body quantum marginal problems described in [6]? There is a natural scaling algorithm generalizing Algorithm 1 but it has so far evaded analysis. Even obtaining a polynomial time algorithm for a promise problem along the lines of Theorem 1 would be rather interesting.

Acknowledgements. We would like to thank Henry Cohn, Eric Naslund, Ion Nechita, Will Sawin, Suvrit Sra, Josué Tonelli, and John Wright for helpful discussions.

References

1 Velleda Baldoni, Michèle Vergne, and Michael Walter. Computation of dilated Kronecker coefficients. *Journal of Symbolic Computation*, 84:113–146, 2018.

2 Charles H Bennett, Sandu Popescu, Daniel Rohrlich, John A Smolin, and Ashish V Thapliyal. Exact and asymptotic measures of multipartite pure-state entanglement. *Physical Review A*, 63(1):012307, 2000.

3 Jonah Blasiak, Thomas Church, Henry Cohn, Joshua A. Grochow, Eric Naslund, William F Sawin, and Chris Umans. On cap sets and the group-theoretic approach to matrix multiplication. *Discrete Analysis*, 2017. doi:10.19086/da.1245.

4 Herm Brascamp and Elliot Lieb. Best constants in Young's inequality, its converse and its generalization to more than three functions. *Advances in Mathematics*, 20:151–172, 1976.

5 Peter Bürgisser, Matthias Christandl, Ketan D Mulmuley, and Michael Walter. Membership in moment polytopes is in NP and coNP. *SIAM Journal on Computing*, 46(3):972–991, 2017.

6 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. *CoRR*, abs/1711.08039, 2017. arXiv:1711.08039.

7 Arthur Cayley. On linear transformations. *Cambridge and Dublin Mathematical Journal*, 1:104–122, 1846.

8 Imre Csiszár and Gábor Tusnády. Information geometry and alternating minimization procedures. *Stat. Decis. Suppl.*, 1:205–237, 1984.

9 H. Derksen and G. Kemper. *Computational Invariant Theory*, volume 130. Springer-Verlag, Berlin, 2002.

10 Harm Derksen. Polynomial bounds for rings of invariants. *Proceedings of the American Mathematical Society*, 129(4):955–963, 2001.

11 Ankit Garg, Leonid Gurvits, Rafael Mendes de Oliveira, and Avi Wigderson. A deterministic polynomial time algorithm for non-commutative rational identity testing. *CoRR*, abs/1511.03730, 2015. arXiv:1511.03730.

12 Ankit Garg, Leonid Gurvits, Rafael Mendes de Oliveira, and Avi Wigderson. Algorithmic aspects of brascamp-lieb inequalities. *CoRR*, abs/1607.06711, 2016. arXiv:1607.06711.

13 Valentina Georgoulas, Joel W Robbin, and Dietmar A Salamon. The moment-weight inequality and the Hilbert-Mumford criterion. *math*, abs/1311.0410, 2013. arXiv:1311.0410.

14 Leonid Gurvits. Classical complexity and quantum entanglement. *Journal of Computer and System Sciences*, 69(3):448–484, 2004.

15 Leonid Gurvits. Hyperbolic polynomials approach to van der waerden/schrijver-valiant like conjectures: sharper bounds, simpler proofs and algorithmic applications. *STOC*, pages 417–426, 2006.

16 Leonid Gurvits and Peter N. Yianilos. The deflation-inflation method for certain semidefinite programming and maximum determinant completion problems. *Technical Report, NECI*, 1998.

17 Moritz Hardt. Understanding alternating minimization for matrix completion. *FOCS*, pages 651–660, 2014.

18 David Hilbert. Ueber die Theorie der algebraischen Formen. *Mathematische Annalen*, 36(4):473–534, 1890.

19 David Hilbert. Uber die vollen Invariantensysteme. *Math. Ann.*, 42:313–370, 1893.

20 Christian Ikenmeyer, Ketan D. Mulmuley, and Michael Walter. On vanishing of Kronecker coefficients. *Computational Complexity*, 2017.

21 Gabor Ivanyos, Youming Qiao, and K. V. Subrahmanyam. Non-commutative Edmonds' problem and matrix semi-invariants. *Computational Complexity*, 2016.

22 Gábor Ivanyos, Youming Qiao, and K. V. Subrahmanyam. Constructive noncommutative rank computation in deterministic polynomial time over fields of arbitrary characteristics. *ITCS*, 2017.

23 Prateek Jain, Praneeth Netrapalli, and Sujay Sanghavi. Low-rank matrix completion using alternating minimization. *STOC*, pages 665–674, 2013.

24 George Kempf and Linda Ness. The length of vectors in representation spaces. In *Algebraic Geometry*, volume 732 of *Lecture Notes in Math.*, pages 233–243. Springer, 1979. doi:10.1007/BFb0066647.

25 Frances Kirwan. *Cohomology of quotients in symplectic and algebraic geometry*, volume 31 of *Mathematical Notes*. Princeton University Press, 1984.

26 Alexander Klyachko. Coherent states, entanglement, and geometric invariant theory. *Quantum Physics*, abs/quant-ph/0206012, 2002. arXiv:quant-ph/0206012.

27 Alexander A. Klyachko. Quantum marginal problem and n-representability. *Journal of Physics: Conference Series*, 36(1):72, 2006. doi:10.1088/1742-6596/36/1/014.

28 Carlton Lemke and J. T. Howson. Equilibrium points of bimatrix games. *SIAM Journal on Applied Mathematics*, 12(2):413–423, 1964.

29 Elliot Lieb. Gaussian kernels have only Gaussian maximizers. *Inventiones Mathematicae*, 102:179–208, 1990.

30 Nati Linial, Alex Samorodnitsky, and Avi Wigderson. A deterministic strongly polynomial algorithm for matrix scaling and approximate permanents. *STOC*, pages 644–652, 1998.

31 Ketan Mulmuley. Geometric Complexity Theory V: Equivalence between Blackbox Derandomization of Polynomial Identity Testing and Derandomization of Noether's Normalization Lemma. *FOCS*, pages 629–638, 2012.

32 Ketan D. Mulmuley. Geometric Complexity Theory V: Efficient algorithms for Noether Normalization. *J. Amer. Math. Soc.*, 30(1):225–309, 2017. doi:10.1090/jams/864.

33 Ketan D. Mulmuley and Milind Sohoni. Geometric Complexity Theory I: An Approach to the P vs. NP and Related Problems. *SIAM J. Comput*, 31(2):496–526, 2001.

34 David Mumford. *Geometric invariant theory*. Springer-Verlag, Berlin-New York, 1965.

35 Linda Ness and David Mumford. A stratification of the null cone via the moment map. *American Journal of Mathematics*, 106(6):1281–1329, 1984. URL: http://www.jstor.org/stable/2374395.

36 John von Neumann. *Functional Operators, Vol. II: The Geometry of Orthogonal Spaces*, volume 22 of *Annals of Math. Studies*. Princeton University Press, 1950.

37 Meisam Razaviyayn, Mingyi Hong, and Zhi-Quan Luo. A unified convergence analysis of block successive minimization methods for nonsmooth optimization. *SIAM Journal on Optimization*, 23(2):1126–1153, 2013.

38 R. Sinkhorn. A relationship between arbitrary positive matrices and doubly stochastic matrices. *The Annals of Mathematical Statistics*, 35:876–879, 1964.

39 Bernd Sturmfels. *Algorithms in Invariant Theory*. Texts & Monographs in Symbolic Computation. Springer, 2nd edition, 2008.

40 B. Sury. An elementary proof of the Hilbert-Mumford criterion. *Electron. J. Linear Algebra*, 7:174–177, 2000. doi:10.13001/1081-3810.1053.

41 Frank Verstraete, Jeroen Dehaene, and De Moor Bart. Normal forms and entanglement measures for multipartite quantum states. *Physical Review A*, 68(1), 2003.

42 Michael Walter. *Multipartite Quantum States and their Marginals*. PhD thesis, ETH Zurich, 2014.

43 Michael Walter, Brent Doran, David Gross, and Matthias Christandl. Entanglement polytopes: multiparticle entanglement from single-particle information. *Science*, 340(6137):1205–1208, 2013.

44 Yu Wang, Wotao Yin, and Jinshan Zeng. Global convergence of ADMM in nonconvex nonsmooth optimization. *math*, abs/1511.06324, 2015. arXiv:1511.06324.

45 Chris Woodward. Moment maps and geometric invariant theory. *math*, abs/0912.1132, 2011. arXiv:0912.1132.

46 Yangyang Xu and Wotao Yin. A block coordinate descent method for regularized multiconvex optimization with applications to nonnegative tensor factorization and completion. *SIAM Journal on Imaging Sciences*, 6(3):1758–1789, 2013.

47 Hongyi Zhang and Suvrit Sra. First-order methods for geodesically convex optimization. *math*, abs/1602.06053, 2016. arXiv:1602.06053.

Further Limitations of the Known Approaches for Matrix Multiplication

Josh Alman[*1] **and Virginia Vassilevska Williams**[†2]

1 **MIT CSAIL and EECS, Cambridge, USA**
`jalman@mit.edu`
2 **MIT CSAIL and EECS, Cambridge, USA**
`virgi@mit.edu`

Abstract

We consider the techniques behind the current best algorithms for matrix multiplication. Our results are threefold.

(1) We provide a unifying framework, showing that all known matrix multiplication running times since 1986 can be achieved from a single very natural tensor - the structural tensor T_q of addition modulo an integer q.

(2) We show that if one applies a generalization of the known techniques (arbitrary zeroing out of tensor powers to obtain independent matrix products in order to use the asymptotic sum inequality of Schönhage) to an arbitrary monomial degeneration of T_q, then there is an explicit lower bound, depending on q, on the bound on the matrix multiplication exponent ω that one can achieve. We also show upper bounds on the value α that one can achieve, where α is such that $n \times n^\alpha \times n$ matrix multiplication can be computed in $n^{2+o(1)}$ time.

(3) We show that our lower bound on ω approaches 2 as q goes to infinity. This suggests a promising approach to improving the bound on ω: for variable q, find a monomial degeneration of T_q which, using the known techniques, produces an upper bound on ω as a function of q. Then, take q to infinity. It is not ruled out, and hence possible, that one can obtain $\omega = 2$ in this way.

1998 ACM Subject Classification F.2 Analysis of Algorithms and Problem Complexity

Keywords and phrases matrix multiplication, lower bound, monomial degeneration, structural tensor of addition mod p

Digital Object Identifier 10.4230/LIPIcs.ITCS.2018.25

1 Introduction

One of the most fundamental questions in computer science asks how quickly one can multiply two matrices. Since the surprising subcubic algorithm for $n \times n \times n$ matrix multiplication by Strassen in 1969 [26], there has been a long line of work on improving and refining the techniques and speeding up matrix multiplication algorithms (e.g. [19, 20, 2, 24, 8, 23, 25, 10, 11, 27, 16]). Progress on this problem is typically measured in terms of ω, the smallest constant such that, for any $\delta > 0$, one can design an algorithm for $n \times n \times n$ matrix multiplication running in time $O(n^{\omega+\delta})$. The biggest open question is whether one can achieve $\omega = 2$. The best bound we currently know, due to Le Gall [16], is $\omega \leq 2.3728639$.

* Partially supported by an NSF Graduate Research Fellowship.
† Partially supported by an NSF Career Award, a Sloan Fellowship, NSF Grants CCF-1417238, CCF-1528078 and CCF-1514339, and BSF Grant BSF:2012338.

A related line of work [10, 9, 15, 13] focuses on *rectangular* matrix multiplication instead of square matrix multiplication. Here, progress is measured in terms of α, the largest constant such that for any $\delta > 0$, one can design an algorithm for $n \times n^\alpha \times n$ matrix multiplication running in time $O(n^{2+\delta})$. Recent work [13] improved the best known bound to $\alpha > 0.31389$. The two values ω and α are very related, as $\omega = 2$ if and only if $\alpha = 1$.

All of the aforementioned bounds on ω and α follow a particular approach, which works as follows.[1] The key is to cleverly select a trilinear form (third-order tensor) $\mathbb{T}$ which needs to have two properties. First, there must be an efficient way to compute large tensor powers $\mathbb{T}^{\otimes n}$ of $\mathbb{T}$. This is done by finding a low *border rank expression* for $\mathbb{T}$, which implies (via Schönhage's asymptotic sum inequality) that for sufficiently large n, the power $\mathbb{T}^{\otimes n}$ has low rank. Second, $\mathbb{T}$ must be useful for actually performing matrix multiplication. Multiplying matrices corresponds in a precise way to evaluating a certain *matrix multiplication tensor*, and so to use $\mathbb{T}$ for this task, one needs to show that there is a 'degeneration' transforming $\mathbb{T}$ into a disjoint sum of matrix multiplication tensors. Combining these two properties of $\mathbb{T}$ yields an algorithm for matrix multiplication (see Lemma 4 below for the precise formula).

Of course, the resulting runtime depends on the choice of the tensor $\mathbb{T}$ as well as the bounds one can prove for the two desired properties. Strassen's original algorithm picked $\mathbb{T}$ to be the tensor for $2 \times 2 \times 2$ matrix multiplication itself. Later work used more and more elaborate tensors and corresponding border rank expressions, culminating with the most recent algorithms using the now-famous *Coppersmith-Winograd tensor*. All these tensors seem to come 'out of nowhere', and in particular, come up with seemingly 'magical' border rank identities to show that they have low border rank. We make some progress demystifying the tensors and their border rank expressions below.

1.1 The best known bounds on ω are actually from T_q

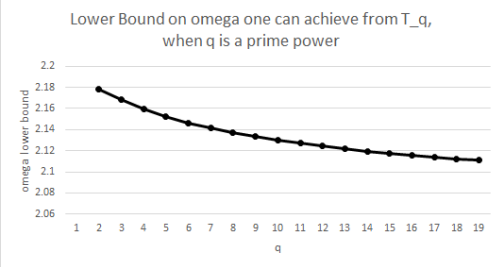
Our first result is a *unifying approach* to achieving all known bounds of ω ([24, 10, 11, 16]) since Strassen's 1986 proof that $\omega < 2.48$.

A simple remark first pointed out to us by Michalek [17] is that the so called Coppersmith-Winograd tensor used in the papers on matrix multiplication since 1990 [10, 11, 16], can be replaced with an equivalent tensor, rotating the original slightly in a certain way (see the Preliminaries), without changing any of the proofs, and thus yielding the same bounds on ω.

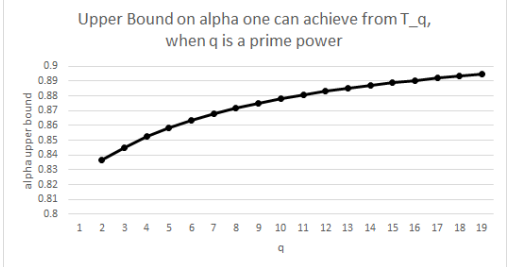
With this in mind, we consider a tensor T_q, the *structural tensor of* $\mathbb{Z}_q$, and give a very simple low rank expression for it based on roots of unity (this expression is natural and likely well-known). We then show that the tensor in [24] and the rotated Coppersmith-Winograd tensors that can be used in [10, 11, 16, 27], are all actually straightforward monomial degenerations of T_q. Since a monomial degeneration of a rank expression gives a border rank expression, this (for example) yields a straightforward border rank expression for the (rotated) Coppersmith-Winograd tensor, which is more intuitive than the border rank expressions from past work.

Another way to view this fact is that *all the bounds on ω since [10] can be viewed as using T_q (in fact for $q = 7$ or 8) as the underlying tensor* $\mathbb{T}$! This also suggests a potential way to improve the known bounds on ω: study other monomial degenerations of T_q.

[1] We give a very high level overview here. More precise definitions are given in Section 2. For a more gentle introduction, we recommend the notes by Markus Bläser [3].



(a) Lower bound on ω that one can achieve using T_q when q is a power of a prime. The bound approaches 2 as $q \to \infty$.



(b) Upper bound on α that one can achieve using T_q when q is a power of a prime. The bound approaches 1 as $q \to \infty$.

Figure 1 Bounds on ω and α that follow from Theorem 1 when q is a prime power

1.2 Limitations on monomial degenerations of T_p

Our second and main result is a lower bound on how fast a matrix multiplication algorithm designed in this way can be whenever $\mathbb{T}$ is a monomial degeneration of T_p:

▶ **Theorem 1** (Informal). *For every p, and for every $\varepsilon \in (0, 1]$, there is an explicit constant $\nu_{p,\varepsilon} > 1$ such that any algorithm for $n \times n^{\varepsilon} \times n$ matrix multiplication designed in the above way using T_p, or a monomial degeneration of T_p, runs in time $\Omega(n^{(1+\varepsilon)\nu_{p,\varepsilon}})$. (See Theorem 7 below for the precise statement).*

The constant $\nu_{p,\varepsilon}$ is defined as follows. Consider first when p is a fixed prime or power of a prime. Let z be the unique real number in $(0, 1)$ such that $3\sum_{j=1}^{p-1} z^j = (p-1)(1-2z^p)$; then

$$\nu_{p,\varepsilon} := (1+\varepsilon) \ln \left[\frac{1-z^p}{(1-z)z^{(p-1)/3}} \right].$$

There is also a variant of Theorem 1 that holds for T_p when p is not necessarily a prime power, but the constant $\nu_{p,\varepsilon} > 1$ is slightly different.

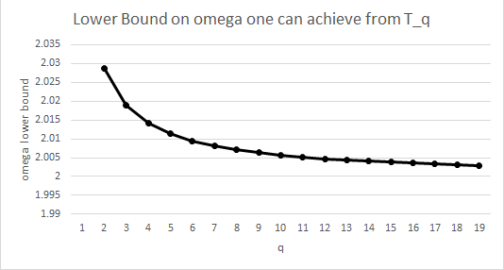
In particular, this shows that:

- This approach yields a square matrix multiplication algorithm with runtime at best $\Omega(n^{2\nu_{p,1}})$, with exponent $2\nu_{p,1} > 2$. Hence, this approach for a fixed p cannot yield $\omega = 2$.
- Let $\varepsilon_p \in (0, 1)$ be such that $(1+\varepsilon_p)\nu_{p,\varepsilon} = 2$. Then, this approach for a fixed p cannot yield a value of α bigger than ε_p.

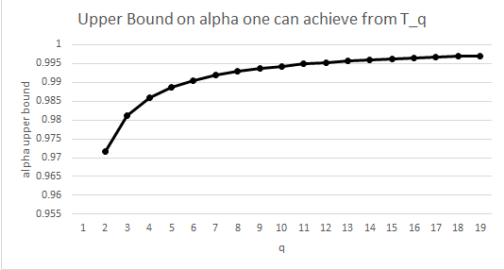
For modest values of p, the value $\nu_p := \nu_{p,1}$ is a fair bit larger than 1. For instance, $\nu_7 \approx 1.07065$. As we will show shortly, the best known algorithms for matrix multiplications use the approach above with a (rotated) Coppersmith-Winograd tensor which is a monomial degeneration of T_7. Our theorem implies among other things that using the approach with T_7 as the starting tensor cannot yield a bound on ω better than 2.14, no matter how one zeroes out the tensor powers of T_7 or its monomial degenerations. We plot the resulting bounds on ω and α for varying p, in Figures 1 and 2 (for technical reasons we discuss below, we get different bounds depending on whether q is a power of a prime).

1.3 A potential idea for improving ω

It should be noted that, despite our lower bounds, not all hope is lost for achieving $\omega = 2$ using T_q tensors. Indeed, in the limit as $q \to \infty$, our ω lower bound approaches 2, and our



(a) Lower bound on ω that one can achieve using T_q.
The bound approaches 2 as $q \to \infty$.



(b) Upper bound on α that one can achieve using T_q.
The bound approaches 1 as $q \to \infty$.

Figure 2 Bounds on ω and α that follow from Theorem 1 for any q

α upper bound approaches 1 (see Lemma 9 in Appendix A for a proof). Hence, our lower bound does not rule out achieving a runtime for $n \times n \times n$ matrix multiplication of $O(n^{2+\delta})$ for all $\delta > 0$ by using bigger and bigger values of q. We find this approach very exciting.

1.4 Tri-Colored Sum-Free Sets

A key component of our lower bound proof is a recent upper bound proved on the asymptotic size of a family of combinatorial objects called tri-colored sum-free sets. For an abelian group G, a *tri-colored sum-free set in* G^n is a set of triples $(a_i, b_i, c_i) \in (G^n)^3$ such that $a_i + b_j + c_k = 0$ if and only if $i = j = k$. In this paper we are especially interested in tri-colored sum-free sets over $\mathbb{Z}_q^n$.

Recent work [12, 14, 4, 18, 21] has proved upper bounds on how large tri-colored sum-free sets in $\mathbb{Z}_q^n$ can be. The bound is originally given in terms of the entropy of certain symmetric distributions, but we give a more explicit form written out by [18, 21] here.

For any integer $q \geq 2$ which is a power of a prime, let ρ be the unique number in $(0, 1)$ satisfying

$$\rho + \rho^2 + \cdots + \rho^{q-1} = \frac{q-1}{3}(1 + 2\rho^q).$$

Then, define $\gamma_q \in \mathbb{R}$ by $\gamma_q := \ln(1-\rho^q) - \ln(1-\rho) - \frac{q-1}{3}\ln(\rho)$.

▶ **Theorem 2** ([14]). *Let q be any prime or power of a prime. Then, any tri-colored sum-free set in $\mathbb{Z}_q^n$ has size at most $e^{\gamma_q n}$. Moreover, there exists a tri-colored sum-free set in $\mathbb{Z}_q^n$ with size $e^{\gamma_q n - o(n)}$.*

One can verify (see Lemma 9 in Appendix A) that $e^{\gamma_q} < q$, meaning in particular that there is no tri-colored sum-free set in $\mathbb{Z}_q^n$ of size $q^{n-o(n)}$. When q is not a prime power, one can also prove this, although the upper bound is not known to be as strong:

▶ **Theorem 3** ([4]). *Let $q \geq 2$ be any positive integer, and let $\kappa := \frac{1}{2}\log((2/3)2^{3/2}) \approx 0.02831$. Then, any tri-colored sum-free set in $\mathbb{Z}_q^n$ has size at most $q^{n(1-\kappa/q+o(1))}$.*

For notational simplicity in our main results in Section 6, define $\gamma_q := (1 - \kappa/q)\log(q)$ when $q \geq 2$ is not a power of a prime.

1.5 Proof Outline

In Section 2 we formally define all the notions related to tensors which are necessary for the rest of the paper, and in Section 3 we give our simple rank expression for T_q and straightforward monomial degenerations of T_{q+2} into CW_q as well as other tensors $\mathbb{T}$ from past work on matrix multiplication algorithms. The remainder of the paper gives the proof of Theorem 1, which proceeds in four main steps:

- In Section 3, we give a simple rank expression for T_q, and show that the rotated Coppersmith-Winograd tensor can be found as a simple monomial degeneration of T_q.
- In Section 4, we show that every matrix multiplication tensor has a zeroing out into a large number of independent triples. This generalizes a classical result that matrix multiplication tensors have monomial degenerations into a large number of independent triples.
- In Section 5, we show that if tensor A is a monomial degeneration of tensor B, and large powers of A can be zeroed out into many independent triples, then large powers of B can as well.
- Finally, in Section 6, we combine the above to show that if any tensor $\mathbb{T}$ is a monomial degeneration of T_q, and yields a fast matrix multiplication algorithm (meaning it can be zeroed out into many independent triples), then T_q can be zeroed out into many independent triples as well. By noticing that independent triples in T_q correspond to tri-colored sum-free sets, and combining with the upper bounds on the size of such a set, we get our lower bound.

1.6 Comparison with Past Work

There are two papers which have proved lower bounds on the value of ω that one can achieve using certain techniques.

The first is a work by Ambainis et al. [1]. They show a lower bound of $\Omega(n^{2.3078})$ for any algorithm for $n \times n \times n$ matrix multiplication one can design using the 'laser method with merging' using the Coppersmith-Winograd tensor and its relatives. The laser method is a technique proposed by Strassen [24] and used by all recent work [10, 11, 27, 16, 13] in order to show that the Coppersmith-Winograd tensor has a zeroing out into many big disjoint matrix multiplication tensors (the second property of the two properties of a tensor $\mathbb{T}$ we described earlier). While the bound that Ambainis et al. get is better than ours, our result is much more general: First, the Ambainis et al. bound is for algorithms which use the Coppersmith-Winograd tensor and some tensors like it, whereas ours applies to any tensor which is an arbitrary monomial degeneration of T_q. Second, their bound only applies when the laser method with merging is used to zero out the tensor into matrix multiplication tensors, whereas ours applies to any possible monomial degeneration into matrix multiplication tensors.

The second prior work is by Blasiak et al. [4]. Like us, the authors also use recent bounds on the size of certain tri-colored sum-free sets in order to prove lower bounds. However, rather than the tensor-based approach to matrix multiplication algorithms which we have been discussing, and which has been used in all of the improvements to ω and α to date, they instead focus on the 'group-theoretic approach' to matrix multiplication [7, 6]. This approach has been designed around formulating approaches that would imply $\omega = 2$ rather than on attempting any small improvement to the bounds on ω, and this paper refutes some earlier

conjectures along these lines. The work of Blasiak et al. implies that certain approaches to achieving $\omega = 2$ are impossible, similar to our work here.

In personal communication, Cohn [5] stated that the Coppersmith-Winograd tensor CW_q leads to a STPP (simultaneous triple product property) construction in $\mathbb{Z}_m^n$ with $m = q$ and n tending to infinity. Blasiak et al. present lower bounds on what can be proved about ω using the group theoretic approach using STPP constructions in $\mathbb{Z}_m^n$ for any fixed m, and hence their results imply that the *group-theoretic variant* of the Coppersmith-Winograd approach cannot yield $\omega = 2$ using a fixed q. It is not clear exactly what lower bounds this result implies for the original laser method approach, or for arbitrary monomial degenerations of T_q. Thus, we consider our results complementary to those of Blasiak et al. Furthermore, our results include limitations for rectangular matrix multiplication, which the prior work does not mention.

2 Preliminaries

In this section we introduce all the notions related to tensors which are used in the rest of the paper.

2.1 Tensor Definitions

Let $X = \{x_1, \ldots, x_n\}$, $Y = \{y_1, \ldots, y_m\}$, and $Z = \{z_1, \ldots, z_p\}$ be three sets of formal variables. A *tensor over* X, Y, Z is a trilinear form

$$T = \sum_{i=1}^{n} \sum_{j=1}^{m} \sum_{k=1}^{p} T_{ijk} x_i y_j z_k,$$

where the T_{ijk} terms are elements of a field $\mathbb{F}$. The *size* of a tensor A, denoted $|A|$, is the number of nonzero coefficients A_{ijk}. There are three particular tensors we will focus on in this paper. The *matrix multiplication tensor* $\langle n, m, p \rangle$ is given by

$$\langle n, m, p \rangle = \sum_{i=1}^{n} \sum_{j=1}^{m} \sum_{k=1}^{p} x_{ij} y_{jk} z_{ki}.$$

For a positive integer q, the *structural tensor of* $\mathbb{Z}_q$, denoted T_q, is given by

$$T_q = \sum_{i=0}^{q-1} \sum_{j=0}^{q-1} x_i y_j z_{-i-j \pmod q}.$$

For any positive integer q, the qth Coppersmith-Winograd tensor C_q [10] is given by $x_0 y_0 z_{q+1} + x_0 y_{q+1} z_0 + x_{q+1} y_0 z_0 + \sum_{i=1}^{q} (x_0 y_i z_i + x_i y_0 z_i + x_i y_i z_0)$. It is not hard to verify that using the Coppersmith-Winograd approach, one can obtain exactly the same values for ω from the following *rotated Coppersmith-Winograd tensor* CW_q, given by

$$CW_q = x_0 y_0 z_{q+1} + x_0 y_{q+1} z_0 + x_{q+1} y_0 z_0 + \sum_{k=1}^{q} \left(x_0 y_k z_{q+1-k} + x_k y_0 z_{q+1-k} + x_k y_{q+1-k} z_0 \right).$$

The main reason why CW_q works just as well as the original Coppersmith-Winograd tensor C_q is because they both have border rank $q + 2$ and because of the following other structural reason which is what is used in the prior work on fast matrix multiplication:

Let $X_0 = \{x_0\}$, $X_1 = \{x_1, \ldots, x_q\}$, $X_2 = \{x_{q+2}\}$. Similarly, let $Y_0 = \{y_0\}$, $Y_1 = \{y_1, \ldots, y_q\}$, $Y_2 = \{y_{q+2}\}$, and $Z_0 = \{z_0\}$, $Z_1 = \{z_1, \ldots, z_q\}$, $Z_2 = \{z_{q+2}\}$. When you restrict C_q and CW_q to $X_0 \times Y_2 \times Z_0$, or $X_2 \times Y_0 \times Z_0$, or $X_0 \times Y_0 \times Z_2$, both of them are isomorphic to $\langle 1, 1, 1\rangle$. When you restrict them to $X_0 \times Y_1 \times Z_1$, both are isomorphic to $\langle 1, 1, q\rangle$, when you restrict them to $X_1 \times Y_0 \times Z_1$, both are isomorphic to $\langle q, 1, 1\rangle$, and when you restrict them to $X_1 \times Y_1 \times Z_0$, both are isomorphic to $\langle 1, q, 1\rangle$. The Coppersmith-Winograd approach only looks at products of these blocks in higher tensor powers, which are hence isomorphic to the same matrix multiplication tensors and give the same bounds on ω.

2.2 Subsets and Degenerations

For two tensors A, B, we say that $A \subseteq B$ if A_{ijk} is always either B_{ijk} or 0. For instance, we can see that $CW_q \subseteq T_{q+2}$. We furthermore say that A is a *monomial degeneration* of B if $A \subseteq B$ and there are functions $a : X \to \mathbb{Z}$, $b : Y \to \mathbb{Z}$, and $c : Z \to \mathbb{Z}$ such that whenever $B_{ijk} \neq 0$,

- we have $a(x_i) + b(y_j) + c(z_k) \geq 0$, and
- furthermore, $a(x_i) + b(y_j) + c(z_k) = 0$ if and only if $A_{ijk} \neq 0$ as well.

We note that in prior work, degenerations are defined via polynomials in a variable ε, however when the degenerations are single monomials, the above definition is equivalent, where a, b, c give the corresponding exponents of ε.

Finally, we say that A is a *zeroing out* of B if A is a monomial degeneration of B such that $a(x) \geq 0$ for all $x \in X$, $b(y) \geq 0$ for all $y \in Y$, and $c(z) \geq 0$ for all $z \in Z$. One can think of this as substituting 0 for any variable which a, b, or c maps to a positive value.

2.3 Tensor Product

Let X, X', Y, Y', Z, Z' be sets of formal variables. If A is a tensor over X, Y, Z, and B is a tensor over X', Y', Z', then the *tensor product of A and B*, denoted $A \otimes B$, is a tensor over $X \times X', Y \times Y', Z \times Z'$ given by

$$A \otimes B = \sum_{\substack{(x_i, x'_{i'}) \in X \times X' \\ (y_j, y'_{j'}) \in Y \times Y' \\ (z_k, z'_{k'}) \in Z \times Z'}} A_{ijk} B_{i'j'k'} (x_i, x'_{i'})(y_j, y'_{j'})(z_k, z'_{k'}).$$

The nth tensor power of a tensor A, denoted $A^{\otimes n}$, is the result of tensoring n copies of A together. In other words, $A^{\otimes 1} = A$, and $A^{\otimes n} = A \otimes A^{\otimes n-1}$.

Tensor products preserve many key properties of tensors. For instance, if $A \subseteq C$ and $B \subseteq D$, then $A \otimes B \subseteq C \otimes D$, and this is also true if subset is replaced by monomial degeneration, or by zeroing out.

For a nonnegative integer k, if A is a tensor over X, Y, Z, and if $X_1, \ldots, X_k$ are k disjoint copies of X, and similar for Y and Z, then $k \odot A$ denotes the (disjoint) sum of k copies of A, one over X_i, Y_i, Z_i for each $1 \leq i \leq k$.

2.4 Independent Triples

Two triples $(x, y, z), (x', y', z') \in X \times Y \times Z$ are *independent* if $x \neq x'$, $y \neq y'$, and $z \neq z'$. A tensor A is independent if, whenever $A_{ijk} \neq 0$ and $A_{i'j'k'} \neq 0$, and $(i, j, k) \neq (i', j', k')$, then the triples (x_i, y_j, z_k) and $(x_{i'}, y_{j'}, z_{k'})$ are independent.

2.5 Tensor Rank

A tensor T over X, Y, Z is a *rank-one tensor* if there are coefficients a_x for each $x \in X$, b_y for each $y \in Y$, and c_z for each $z \in Z$ in the underlying field $\mathbb{F}$ such that

$$T = \left(\sum_{x\in X} a_x \cdot x\right)\left(\sum_{y\in Y} b_y \cdot y\right)\left(\sum_{z\in Z} c_z \cdot z\right) = \sum_{(x,y,z)\in X\times Y\times Z} a_x b_y c_z \cdot xyz.$$

More generally, T is a *rank-k tensor* if it can be written as the sum of k rank-one tensors. The rank of T, denoted $R(T)$, is the smallest k such that R is a rank-k tensor.

We can generalize this notion slightly to define the border rank of a tensor. We will now allow the a_x, b_y, and c_z coefficients to be elements of the polynomial ring $\mathbb{F}[\varepsilon]$ for a formal variable ε. We say that T is a *border rank-one tensor* if there are coefficients a_x, b_y, c_z in $\mathbb{F}[\varepsilon]$ and an integer $h \geq 0$ such that when

$$\left(\sum_{x\in X} a_x \cdot x\right)\left(\sum_{y\in Y} b_y \cdot y\right)\left(\sum_{z\in Z} c_z \cdot z\right) \tag{1}$$

is expanded as a polynomial in ε whose coefficients are tensors over X, Y, Z, then T is the coefficient of ε^h, and the coefficient of $\varepsilon^{h'}$ is 0 for all $0 \leq h' < h$. Similarly, the border rank $\underline{R}(T)$ of T is the smallest number of expressions of the form (1) whose sum, when written as a polynomial in ε, has T as its lowest order coefficient.

It is not hard to see that if A is a monomial degeneration of B, then $\underline{R}(B) \leq \underline{R}(A) \leq R(A)$.

2.6 Matrix Multiplication Tensor and Algorithms

Now that we have defined tensor rank, we can define ω as the infimum over all reals so that $R(\langle n, n, n\rangle) \leq O(n^{\omega+\varepsilon})$ for all $\varepsilon > 0$. Similarly, for any $\varepsilon \in (0, 1)$, define ω_ε to be the smallest real such that an $n \times n^\varepsilon$ matrix can be multiplied by an $n^\varepsilon \times n$ matrix in $n^{\omega_\varepsilon+o(1)}$ time.

We present a useful Lemma that follows from the work of Schönhage, which shows how the tensor rank notions we have been discussing can give bounds on ω_ε.

▶ **Lemma 4.** *If* $R(f \odot \langle n, n^\varepsilon, n\rangle) \leq g$, *then* $\omega_\varepsilon \leq \log_n(\lceil g/f\rceil)$.

Proof. By Schönhage [23] (see also [3, Lemma 7.7]), we have that $R(f \odot \langle n, n^\varepsilon, n\rangle) \leq g$ implies that for all integers $s \geq 1$, $R(f \odot \langle n^s, n^{s\varepsilon}, n^s\rangle) \leq f\lceil g/f\rceil^s$. Hence, multiplying an $n^s \times (n^s)^\varepsilon$ by an $(n/s)^\varepsilon \times n^s$ matrix can be done in $O(f\lceil g/f\rceil^s)$ time. Thus $\omega_\varepsilon \leq \lim_{s\to\infty} \log(f\lceil g/f\rceil^s)/\log(n^s) = \log_n(\lceil g/f\rceil)$. ◀

We can also define α as the largest real such that $R(\langle n, n^\alpha, n\rangle) \leq n^{2+o(1)}$. It is known that $\alpha \in [0.31, 1]$, and clearly $\alpha = 1$ if and only if $\omega = 2$.

3 The mod-p tensor and its degenerations

In this section, we give a rank expression for T_p, and then a monomial degeneration of T_{q+2} into CW_q.

3.1 The rank of T_p

Let us consider the tensor T_p of addition modulo p for any integer $p \geq 2$; recall that in trilinear notation, T_p is defined as

$$T_p = \sum_{\substack{i,j,k \in \{0,\ldots,p-1\} \\ i+j+k \equiv 0 \pmod p}} x_i y_j z_k.$$

The rank of T_p is p, as can be seen by the expression below. Let $w_1, \ldots, w_p \in \mathbb{C}$ be the pth roots of unity, meaning that $\sum_{i=1}^{p} w_i = 0$, and that for each i, $w_i^p = 1$. Then,

$$T_p = \frac{1}{p} \sum_{\ell=1}^{p} \left(\sum_{i=0}^{p-1} w_\ell^i x_i \right) \left(\sum_{j=0}^{p-1} w_\ell^j y_j \right) \left(\sum_{k=0}^{p-1} w_\ell^k z_k \right).$$

The above gives a rank expression for T_q over $\mathbb{C}$, which is sufficient for the approaches for matrix multiplication algorithms discussed above. That said, one can easily modify it to get an expression over some other fields as well. For instance, suppose $p+1$ is an odd prime. Then, we know that $\sum_{a=1}^{p} a \equiv 0 \pmod{p+1}$, and that $a^p \equiv 1 \pmod{p+1}$ for any $1 \leq a \leq p$, so we similarly get the following rank expression over $GF(p+1)$:

$$T_p = -\sum_{a=1}^{p} \left(\sum_{i=0}^{p-1} a^i x_i \right) \left(\sum_{j=0}^{p-1} a^j y_j \right) \left(\sum_{k=0}^{p-1} a^k z_k \right).$$

3.2 Monomial degeneration of T_{q+2} into CW_q

Here we will show that the rotated CW tensor CW_q for integer $q \geq 1$ is a degeneration of T_{q+2}. Recall that

$$\begin{aligned} CW_q = {} & x_0 y_0 z_{q+1} + x_0 y_{q+1} z_0 + x_{q+1} y_0 z_0 \\ & + \sum_{k=1}^{q} \left(x_0 y_k z_{q+1-k} + x_k y_0 z_{q+1-k} + x_k y_{q+1-k} z_0 \right). \end{aligned} \tag{2}$$

For ease of notation, we will change the indexing of the z variables in T_{q+2} (i.e. rename the variables) from our original definition[2] to write

$$T_{q+2} = \sum_{\substack{i,j,k \in \{0,\ldots,q+1\} \\ i+j+k \equiv q+1 \pmod{q+2}}} x_i y_j z_k. \tag{3}$$

In this form, one can see that CW_q is the subset of T_{q+2} consisting of all the terms containing at least one of x_0, y_0, or z_0. With this in mind, our degeneration of T_{q+2} is as follows. We will pick:

- $a(x_0) = 0$, $a(x_{q+1}) = 2$, and $a(x_i) = 1$ for $1 \leq i \leq q$, similarly,
- $b(y_0) = 0$, $b(y_{q+1}) = 2$, and $b(y_j) = 1$ for $1 \leq j \leq q$, and,
- $c(z_0) = -2$, $c(z_{q+1}) = 0$, and $c(z_k) = -1$ for $1 \leq k \leq q$.

We need to verify that for every term $x_i y_j z_k$ in (3) we have $a(x_i) + b(y_j) + c(z_k) \geq 0$, and moreover that for such $x_i y_j z_k$, $a(x_i) + b(y_j) + c(z_k) = 0$ if and only if $x_i y_j z_k$ also appears in (2). This is quite straightforward, but we do it here for completeness. Consider any term $x_i y_j z_k$ in (3). We consider three cases based on k:

[2] For every index $k \in \{0, 1, \ldots, q+1\}$, we will rename z_k to $z_{k-1 \pmod{q+2}}$.

- If $k = 0$, then our term is of the form $x_i y_{q+2-i} z_0$ for $0 \le i \le q+2$. This term always appears in (2) as well, and we can see that we always have $a(x_i) = 2 - b(y_{q+2-i})$, and so $a(x_i) + b(y_{q+2-i}) + c(z_0) = 0$.
- If $k = q+1$, then $c(z_{q+1}) = 0$, and we always have $a, b \ge 0$, so we definitely have that $a(x_i) + b(y_j) + c(z_k) \ge 0$. Moreover, we can only achieve 0 when $a = b = 0$, with the term $x_0 y_0 z_{q+1}$, which is the only term with z_{q+1} which appears in (2).
- If $1 \le k \le q$, then since $x_0 y_0 z_k$ is not a term in (3), we must have that $a(x_i) + b(y_j) \ge 1$, and so $a(x_i) + b(y_j) + c(z_k) \ge 0$. Moreover, we only achieve $a(x_i) + b(y_j) + c(z_k) = 0$ when $(a, b) = (0, 1)$ or $(1, 0)$, which correspond to the terms of the form $x_0 y_k z_{q+1-k}$ or $x_k y_0 z_{q+1-k}$ in (2).

3.3 Monomial degeneration of T_{q+1} into Strassen's 1986 tensor.

Strassen's 1986 tensor is defined for any integer $q \ge 1$ and is given by $S_q := \sum_{i=1}^{q} x_0 y_i z_{q+1-i} + x_i y_0 z_{q+1-i}$.

Similar to before, we will show that S_q is a degeneration of T_{q+1}, which we can write as

$$T_{q+1} = \sum_{\substack{i,j,k \in \{0,\dots,q\} \\ i+j+k \equiv q \pmod{q+1}}} x_i y_j z_k. \tag{4}$$

Our degeneration is as follows: $a(x_0) = b(x_0) = 0$, $a(x_i) = b(y_i) = 1$ for all $i \ge 1$, $c(z_q) = 0$ and $c(z_k) = -1$ for all $k \ge 1$. Simple casework shows again that the possible values for $a(x_i) + b(y_j) + c(z_k)$ are $0, 1, 2$, and that 0 is only achieved for the terms in S_q. Among other things, this degeneration gives a simple proof that the border rank of S_q is $q+1$.

Since a monomial degeneration of a rank expression gives a border rank expression, this shows in particular that the border rank of CW_q is $q+2$. Furthermore, it shows that the best known bounds for ω [10, 27, 16] can be obtained from T_7. Finally, since we only used monomial degenerations, we will be able to obtain lower bounds on what bounds on ω one can achieve via zeroing out powers of the CW_q tensor.

4 Independent Triples in Matrix Multiplication Tensors

In this section we show that there is a zeroing out of any matrix multiplication tensor into a fairly large independent tensor. This strengthens a classic result (see eg. [3, Lemma 8.6]) that any matrix multiplication tensor has a monomial degeneration into a fairly large independent tensor.

▶ **Lemma 5.** *For every positive integer q, and $\varepsilon \in (0, 1]$, there is a zeroing out of $\langle q, q^\varepsilon, q\rangle^{\otimes n}$ into $q^{(1+\varepsilon)n - o(n)}$ independent triples.*

Proof. Recall that $\langle q, q^\varepsilon, q\rangle = \sum_{i=1}^{q} \sum_{j=1}^{q^\varepsilon} \sum_{k=1}^{q} x_{ij} y_{jk} z_{ki}$. Hence,

$$\langle q, q^\varepsilon, q\rangle^{\otimes n} = \sum_{\vec{i},\vec{k} \in [q]^n,\ \vec{j} \in [q^\varepsilon]^n} x_{\vec{i}\vec{j}} y_{\vec{j}\vec{k}} z_{\vec{k}\vec{i}}.$$

We will zero out variables in three phases, and after the third phase we will have a sufficiently large independent tensor as desired.

4.1 Phase one

For vectors $\vec{i}, \vec{k} \in [q]^n$, and values $a, b \in [q]$, let $t_{ab}(\vec{ik})$ denote the number of $1 \leq \alpha \leq n$ such that $\vec{i}_\alpha = a$ and $\vec{k}_\alpha = b$. We say that $\vec{ik}$ is *balanced* if, for all $a, b, c, d \in [q]$, we have $t_{ab}(\vec{ik}) = t_{cd}(\vec{ik})$. We similarly say that $\vec{ij}$ is balanced if $t_{ab}(\vec{ij}) = t_{cd}(\vec{ij})$ for every $a, c \in [q]$ and $b, d \in [q^\varepsilon]$, and say that $\vec{jk}$ is balanced similarly. In the first phase, we zero out every variable $x_{\vec{ij}}$ such that $\vec{ij}$ is not balanced. We similarly zero out $y_{\vec{jk}}$ such that $\vec{jk}$ is not balanced, and $z_{\vec{ki}}$ such that $\vec{ki}$ is not balanced.

Note that if $\vec{ik}$ is balanced, then for each $a, b \in [q]$, we have $(\vec{i}_\alpha, \vec{k}_\alpha) = (a, b)$ for exactly n/q^2 choices of $\alpha \in [n]$. Hence, the number of choices of $\vec{i}, \vec{k} \in [q]^n$ such that $\vec{ik}$ is balanced is exactly $L_2 := \binom{n}{\frac{n}{q^2}, \frac{n}{q^2}, \ldots, \frac{n}{q^2}} = q^{2n-o(n)}$. If $\vec{ik}$ is balanced, then notice that the number K_ε of choices of $\vec{j} \in [q^\varepsilon]^n$ such that $\vec{ij}$ and $\vec{jk}$ are also balanced is independent of what $\vec{i}$ and $\vec{k}$ are, and satisfies $K_\varepsilon = q^{O(n)}$.

Similarly, the number of choices of $\vec{i} \in [q]^n$ and $\vec{j} \in [q^\varepsilon]^n$ such that $\vec{ij}$ is balanced is $L_{1+\varepsilon} := \binom{n}{\frac{n}{q^{1+\varepsilon}}, \frac{n}{q^{1+\varepsilon}}, \ldots, \frac{n}{q^{1+\varepsilon}}} = q^{(1+\varepsilon)n-o(n)}$. Moreover, when $\vec{ij}$ is balanced, the number K_1 of choices of $\vec{k}$ such that $\vec{ik}$ and $\vec{jk}$ are balanced satisfies $K_1 = q^{O(n)}$. Note that $L_2 K_\varepsilon = L_{1+\varepsilon} K_1$, since both count the number of triples remaining after phase one, and in particular, $K_1 \geq K_\varepsilon$.

4.2 Phase two

Let M be an odd prime number to be determined. Pick $w_0, w_1, \ldots, w_n \in [M]$ independently and uniformly at random, then define the hash functions $h_X : X \to [M]$, $h_Y : Y \to [M]$, and $h_Z : Z \to [M]$, by:

$$h_X(x_{\vec{ij}}) = 2 \sum_{\alpha=1}^{n} w_\alpha \cdot (\vec{i}_\alpha - \vec{j}_\alpha) \pmod{M},$$

$$h_Y(y_{\vec{jk}}) = 2w_0 + 2 \sum_{\alpha=1}^{n} w_\alpha \cdot (\vec{j}_\alpha - \vec{k}_\alpha) \pmod{M},$$

$$h_Z(z_{\vec{ki}}) = w_0 + \sum_{\alpha=1}^{n} w_\alpha \cdot (\vec{i}_\alpha - \vec{k}_\alpha) \pmod{M}.$$

Notice that, for every choice of $\vec{i}, \vec{j}, \vec{k} \in [q]^n$, we have that $h_X(x_{\vec{ij}}) + h_Y(y_{\vec{jk}}) = 2h_Z(z_{\vec{ki}}) \pmod{M}$. Now, let $H \subseteq [M]$ be a subset of size $|H| \geq M^{1-o(1)}$ which does not contain any nontrivial three-term arithmetic progressions mod M; in other words, if $a, b, c \in H$ such that $a + b = 2c \pmod{M}$, then $a = b = c$. Such a set is constructed by Salem and Spencer [22]. In the second phase, we zero out all $x_{\vec{ij}}$ such that $h_X(x_{\vec{ij}}) \notin H$, and similarly for the y and z variables. As a result, every term $x_{\vec{ij}} y_{\vec{jk}} z_{\vec{ki}}$ remaining in our tensor satisfies:

- $\vec{ij}$, $\vec{jk}$, and $\vec{ki}$ are balanced, and
- $h_X(x_{\vec{ij}}) = h_Y(y_{\vec{jk}}) = h_Z(z_{\vec{ki}})$.

4.3 Phase three

In the third phase we zero out some remaining variables to ensure that our resulting tensor is independent. First, however, we will compute some expected values.

For $h \in H$, let S_h be the set of terms $x_{\vec{ij}} y_{\vec{jk}} z_{\vec{ki}}$ remaining in our tensor after stage two such that $h_X(x_{\vec{ij}}) = h_Y(y_{\vec{jk}}) = h_Z(z_{\vec{ki}}) = h$. For a given term $x_{\vec{ij}} y_{\vec{jk}} z_{\vec{ki}}$ which was not

zeroed out in phase one, it will be in S_h whenever $h_X(x_{\vec{ij}}) = h$ and $h_Y(y_{\vec{jk}}) = h$, since in that case we must also have that $h_Z(z_{\vec{ki}}) = h$ as the three are in arithmetic progression. For a fixed choice of $\vec{i}, \vec{j}, \vec{k}$ such that $\vec{ij}$ and $\vec{jk}$ are balanced, we can see that $h_X(x_{\vec{ij}})$ and $h_Y(y_{\vec{jk}})$ are independent and uniformly random elements of $[M]$ (the randomness is over choosing the w_α values). Hence, this term will be in S_h with probability $1/M^2$, and so $\mathbb{E}[|S_h|] = L_{1+\varepsilon} \cdot K_1/M^2$.

Next, for $h \in H$, let P_h be the set of pairs of terms $(x_{\vec{ij}}y_{\vec{jk}}z_{\vec{ki}}, x_{\vec{i'j'}}y_{\vec{j'k'}}z_{\vec{k'i'}})$ such that both terms are in S_h, and $\vec{i} = \vec{i'}$ and $\vec{j} = \vec{j'}$, meaning they share the same x variable. Again, there are $L_{1+\varepsilon}$ choices for $\vec{i}$ and $\vec{j}$, then K_1 choices each for $\vec{k}$ and $\vec{k'}$, and similar to before, such a choice of $\vec{i}, \vec{j}, \vec{k}, \vec{k'}$ will be put in P_h with probability $1/M^3$. Hence, $\mathbb{E}[|P_h|] \leq L_{1+\varepsilon} \cdot K_1^2/M^3$. Similar calculations hold if we instead look at pairs Q_h which share a y variable, showing that $\mathbb{E}[|Q_h|] \leq L_{1+\varepsilon} \cdot K_1^2/M^3$, or pairs R_h which share a z variable, showing that $\mathbb{E}[|R_h|] \leq L_2 \cdot K_\varepsilon^2/M^3 \leq L_{1+\varepsilon} \cdot K_1^2/M^3$.

We now do our final zeroing out. If there are any distinct terms $x_{\vec{ij}}y_{\vec{jk}}z_{\vec{ki}}$ and $x_{\vec{i'j'}}y_{\vec{j'k'}}z_{\vec{k'i'}}$ remaining in our tensor such that $\vec{i} = \vec{i'}$ and $\vec{j} = \vec{j'}$, then we zero out $x_{\vec{ij}}$. We similarly zero out any variables $y_{\vec{jk}}$ or $z_{\vec{ki}}$ which appear in multiple terms. As a result, our final tensor is definitely independent.

It remains to show that it has enough terms remaining. Since each pair of terms left from phase two which share a variable is removed in phase three, we see that the number of terms remaining is at least

$$\sum_{h \in H} |S_h| - 2|P_h| - 2|Q_h| - 2|R_h|.$$

Let us pick M to be an odd prime number in the range $[12K_1, 24K_1]$. Hence, using our expected value calculations from before, we see that the expected number of remaining terms is at least

$$|H| \cdot \left(\frac{L_{1+\varepsilon}K_1}{M^2} - 6\frac{L_{1+\varepsilon}K_1^2}{M^3} \right) = \frac{|H|L_{1+\varepsilon}K_1}{M^2}\left(1 - 6\frac{K_1}{M}\right) \geq \frac{M^{1-o(1)}L_{1+\varepsilon}K_1}{M^2}\left(1 - 6\frac{1}{12}\right)$$
$$\geq \frac{L_{1+\varepsilon}}{K_1^{o(1)}} \geq q^{(1+\varepsilon)n - o(n)},$$

where the last step follows since $L_{1+\varepsilon} = q^{(1+\varepsilon)n-o(n)}$ and $K_1 = q^{O(n)}$. By the probabilistic method, there is a choice of hash functions which achieves this expected number of independent triples, as desired. ◀

5 Monomial Degenerations

▶ **Lemma 6.** *Suppose A and B are two tensors over X, Y, Z such that A is a monomial degeneration of B. Further suppose that $A^{\otimes n}$ has zeroing out into $f(n)$ independent triples. Then, $B^{\otimes n}$ has a zeroing out into $\Omega(f(n)/n^2)$ independent triples.*

Proof. Let $a : X \to \mathbb{Z}$, $b : Y \to \mathbb{Z}$, and $c : Z \to \mathbb{Z}$ be the functions for the monomial degeneration such that

- $a(x_i) + b(y_j) + c(z_k) \geq 0$ for all $x_i y_j z_k \in B$, and
- furthermore $a(x_i) + b(y_j) + c(z_k) = 0$ if and only if $x_i y_j z_k \in A$.

Let $a^- := \min_{x \in X} a(x)$ and $a^+ := \max_{x \in X} a(x)$, and define b^-, b^+, c^-, and c^+ similarly. Now, $B^{\otimes n}$ is a tensor over X^n, Y^n, Z^n. Define $a^n : X^n \to \mathbb{Z}$, by $a^n(x_{i_1}, \ldots, x_{i_n}) = \sum_{\alpha=1}^n a(x_{i_\alpha})$, and define $b^n : Y^n \to \mathbb{Z}$ and $c^n : Z^n \to \mathbb{Z}$ similarly. It follows that

- $a^n(x_{i_1}, \ldots, x_{i_n}) + b^n(y_{j_1}, \ldots, y_{j_n}) + c^n(z_{k_1}, \ldots, z_{k_n}) \geq 0$ for all $x_{i_1} \cdots x_{i_n} y_{j_1} \cdots y_{j_n} z_{k_1} \cdots z_{k_n} \in B^{\otimes n}$, and
- furthermore $a^n(x_{i_1}, \ldots, x_{i_n}) + b^n(y_{j_1}, \ldots, y_{j_n}) + c^n(z_{k_1}, \ldots, z_{k_n}) = 0$ if and only if $x_{i_1} \cdots x_{i_n} y_{j_1} \cdots y_{j_n} z_{k_1} \cdots z_{k_n} \in A^{\otimes n}$.

The range of a^n is integers in $[a^- n, a^+ n]$. For each integer p in that range, let X_p^n be the set of $x_{i_1} \cdots x_{i_n} \in X^n$ such that $a^n(x_{i_1} \cdots x_{i_n}) = p$. Define Y_q^n for integers $q \in [b^- n, b^+ n]$, and Z_r^n for integers $r \in [c^- n, c^+ n]$, similarly. Now, for $(p, q, r) \in [a^- n, a^+ n] \times [b^- n, b^+ n] \times [c^- n, c^+ n]$, let $B_{p,q,r}^{\otimes n}$ be the tensor one gets from $B^{\otimes n}$ by zeroing out all the X^n variables not in X_p^n, all the Y^n variables not in Y_q^n, and all the Z^n variables not in Z_r^n. Then, letting W be the set of triples of integers in $[a^- n, a^+ n] \times [b^- n, b^+ n] \times [c^- n, c^+ n]$, we see that

$$A^{\otimes n} = \sum_{(p,q,r) \in W | p+q+r=0} B_{p,q,r}^{\otimes n},$$

and each term of $A^{\otimes n}$ appears in exactly one of the summands. Now, let $A^{\otimes n'}$ be the zeroing out of $A^{\otimes n}$ into $f(n)$ independent triples. Let $B_{p,q,r}^{\otimes n'}$ be the zeroing out of $B_{p,q,r}^{\otimes n}$ in which we zero out those same variables. Hence,

$$A^{\otimes n'} = \sum_{(p,q,r) \in W | p+q+r=0} B_{p,q,r}^{\otimes n'},$$

where the sum is hence a disjoint sum of independent triples. The number of terms on the right is $O(n^2)$, and so at least one of the terms on the right must have size at least $|A^{\otimes n'}|/O(n^2) = \Omega(f(n)/n^2)$, as desired. ◀

6 Main Theorem

In this section, we will combine our results above with the bounds on the sizes of tri-colored sum-free sets from past work in order to prove our main theorem. Recall the definition of γ_p from Section 1.4, and define $c_p := e^{\gamma_p}$.

▶ **Theorem 7.** *Let $\varepsilon \in (0, 1]$. Let T be a tensor that is a monomial degeneration of T_p and suppose that $T^{\otimes N}$ can be zeroed out into $F \odot \langle G, G^\varepsilon, G \rangle$, giving a bound $\omega_\varepsilon \leq \omega'_\varepsilon$ where $G^{\omega'_\varepsilon} = \lceil p^N / F \rceil$. Then $\omega'_\varepsilon \geq (1 + \varepsilon) \log_{c_p} p$.*

Proof. Let $g = G^{1/N}$ so that $G = g^N$, and let $f = F^{1/N}$ so that $F = f^N$. Since $T^{\otimes N}$ can be zeroed out into $F \odot \langle G, G^\varepsilon, G \rangle$, via Lemma 5, $T^{\otimes N}$ can be zeroed out into $f^N \cdot g^{(1+\varepsilon)N - o(N)}$ independent triples. Due to Lemma 6 this means that $T_p^{\otimes N}$ can also be zeroed out into $D = f^N \cdot g^{(1+\varepsilon)N - o(N)}/N^2$ independent triples.

Now, let $S = \{(a_1, b_1, c_1), \ldots, (a_D, b_D, c_D)\}$ be the indices of the D independent triples obtained from $T_p^{\otimes N}$. Because they are obtained by zeroing out $T_p^{\otimes N}$, for every i, $a_i + b_i + c_i \equiv 0$ in Z_p^N. Now suppose that for some i, j, k, $a_i + b_j + c_k \equiv 0$ in Z_p^N. If i, j, k are not all the same, then (a_i, b_j, c_k) cannot be in S as the triples in S are independent. However, the only way for a triple of $T_p^{\times N}$ to be removed is if X_{a_i} or Y_{b_j} or Z_{c_k} is set to zero. Suppose that X_{a_i} is set to 0 (the other two cases are symmetric). Then there can be no triple in S sharing a_i as its first index. Thus in fact S forms a tri-colored sum-free set. Hence $D \leq c_p^N$.

From our earlier bound on D we get that $f^N \cdot g^{(1+\varepsilon)N - o(N)}/N^2 \leq c_p^N$, and taking the Nth root of both sides yields $f g^{1+\varepsilon-o(1)}/N^{2/N} \leq c_p$.

Recall that $G^{\omega'_\varepsilon} = \lceil p^N / F \rceil$, so that $g = (\lceil p/f \rceil)^{1/\omega'_\varepsilon}$. Plugging in above, we get that $f(\lceil p/f \rceil)^{(1+\varepsilon)/\omega'_\varepsilon - o(1)} \leq c_p$. Hence, $f^{1-(1+\varepsilon)/\omega'_\varepsilon + o(1)} p^{(1+\varepsilon)/\omega'_\varepsilon - o(1)} \leq c_p$. Since $\omega'_\varepsilon \geq (1 + \varepsilon)$, we have that $f^{1-(1+\varepsilon)/\omega'_\varepsilon + o(1)} \geq 1$. We obtain that $(1 + \varepsilon)/\omega'_\varepsilon \leq \log_p c_p + o(1)$ and

$$\omega'_\varepsilon \geq (1 + \varepsilon - o(1)) \log_{c_p} p.$$ ◀

As a corollary we obtain the following upper bound on what α can be achieved by zeroing out.

▶ **Corollary 8.** *Let T be a tensor that is a monomial degeneration of T_p. If one can prove $\alpha \leq \alpha'$ using the zeroing-out approach then, $\alpha' \leq \frac{2}{\log_{c_p} p} - 1$.*

References

1 Andris Ambainis, Yuval Filmus, and François Le Gall. Fast matrix multiplication: limitations of the coppersmith-winograd method. In *STOC*, pages 585–593, 2015.

2 D. Bini, M. Capovani, F. Romani, and G. Lotti. $O(n^{2.7799})$ complexity for $n \times n$ approximate matrix multiplication. *Inf. Process. Lett.*, 8(5):234–235, 1979.

3 Markus Bläser. Fast matrix multiplication. *Theory of Computing, Graduate Surveys*, 5:1–60, 2013.

4 Jonah Blasiak, Thomas Church, Henry Cohn, Joshua A Grochow, Eric Naslund, William F Sawin, and Chris Umans. On cap sets and the group-theoretic approach to matrix multiplication. *Discrete Analysis*, 2017(3):1–27, 2017.

5 Henry Cohn. personal communication, 2017.

6 Henry Cohn, Robert Kleinberg, Balazs Szegedy, and Christopher Umans. Group-theoretic algorithms for matrix multiplication. In *FOCS*, pages 379–388, 2005.

7 Henry Cohn and Christopher Umans. A group-theoretic approach to fast matrix multiplication. In *FOCS*, pages 438–449, 2003.

8 D. Coppersmith and S. Winograd. On the asymptotic complexity of matrix multiplication. In *SFCS*, pages 82–90, 1981.

9 Don Coppersmith. Rectangular matrix multiplication revisited. *Journal of Complexity*, 13(1):42–49, 1997.

10 Don Coppersmith and Shmuel Winograd. Matrix multiplication via arithmetic progressions. *Journal of symbolic computation*, 9(3):251–280, 1990.

11 A.M. Davie and A. J. Stothers. Improved bound for complexity of matrix multiplication. *Proceedings of the Royal Society of Edinburgh, Section: A Mathematics*, 143:351–369, 4 2013.

12 Jordan S Ellenberg and Dion Gijswijt. On large subsets of $\mathbb{F}_q^n$ with no three-term arithmetic progression. *Annals of Mathematics*, 185(1):339–343, 2017.

13 François Le Gall and Florent Urrutia. Improved rectangular matrix multiplication using powers of the coppersmith-winograd tensor. *CoRR*, abs/1708.05622, 2017. `arXiv:1708.05622`.

14 Robert Kleinberg, William F. Sawin, and David E. Speyer. The growth rate of tri-colored sum-free sets. *math*, abs/1607.00047, 2016. `arXiv:1607.00047`.

15 François Le Gall. Faster algorithms for rectangular matrix multiplication. In *FOCS*, pages 514–523, 2012.

16 François Le Gall. Powers of tensors and fast matrix multiplication. In *ISSAC*, pages 296–303, 2014.

17 Mateusz Michalek. personal communication, 2014.

18 Sergey Norin. A distribution on triples with maximum entropy marginal. *math*, abs/1608.00243, 2016. `arXiv:1608.00243`.

19 V. Y. Pan. Strassen's algorithm is not optimal. In *FOCS*, volume 19, pages 166–176, 1978.

20 V. Y. Pan. New fast algorithms for matrix operations. *SIAM J. Comput.*, 9(2):321–342, 1980.

21 Luke Pebody. Proof of a conjecture of kleinberg-sawin-speyer. *math*, abs/1608.05740, 2016. `arXiv:1608.05740`.

22 Raphaël Salem and Donald C Spencer. On sets of integers which contain no three terms in arithmetical progression. *Proceedings of the National Academy of Sciences*, 28(12):561–563, 1942.

23 A. Schönhage. Partial and total matrix multiplication. *SIAM J. Comput.*, 10(3):434–455, 1981.

24 V. Strassen. The asymptotic spectrum of tensors and the exponent of matrix multiplication. In *FOCS*, pages 49–54, 1986.

25 V. Strassen. Relative bilinear complexity and matrix multiplication. *J. reine angew. Math. (Crelles Journal)*, 375–376:406–443, 1987.

26 Volker Strassen. Gaussian elimination is not optimal. *Numerische Mathematik*, 13(4):354–356, 1969.

27 Virginia Vassilevska Williams. Multiplying matrices faster than coppersmith-winograd. In *STOC*, pages 887–898, 2012.

A Supporting Calculations

We recall some definitions from earlier in the paper. For any integer $q \geq 2$, let ρ be the unique number in $(0, 1)$ satisfying

$$\rho + \rho^2 + \cdots + \rho^{q-1} = \frac{q-1}{3}(1 + 2\rho^q).$$

Then, define $\gamma_q \in \mathbb{R}$ by $\gamma_q := \ln(1-\rho^q) - \ln(1-\rho) - \frac{q-1}{3}\ln(\rho)$. Then, the lower bound on ω we get from using T_q is $2\ln(q)/\gamma_q$. Here we show that this approaches 2 as $q \to \infty$:

▶ **Lemma 9.** $\lim_{q\to\infty} \frac{\gamma_q}{\ln(q)} = 1$.

Proof. Note that, since $\rho \in (0, 1)$, we have

$$\frac{1}{1-\rho} = 1 + \rho + \rho^2 + \cdots > \rho + \rho^2 + \cdots + \rho^{q-1} = \frac{q-1}{3}(1 + 2\rho^q) > \frac{q-1}{3}.$$

Rearranging, we see that $\rho > 1 - 3/(q-1)$. Hence,

$$\begin{aligned}\frac{\gamma_q}{\ln(q)} = \frac{\ln\left(\frac{1-\rho^q}{1-\rho}\right)}{\ln(q)} + \frac{(q-1)\ln(\rho)}{3\ln(q)} &> \frac{\ln\left(1+\rho+\cdots+\rho^{q-1}\right)}{\ln(q)} + \frac{(q-1)\ln(1-\frac{3}{q-1})}{3\ln(q)} \\ &> \frac{\ln\left((q-1)/3\right)}{\ln(q)} + \frac{(q-1)\ln(1-\frac{3}{q-1})}{3\ln(q)}.\end{aligned}$$

As $q \to \infty$, we have that $\ln_q((q-1)/3) \to 1$ and $(q-1)\ln_q(1-3/(q-1)) \to 0$, as desired. ◀

Local Decoding and Testing of Polynomials over Grids*†

Srikanth Srinivasan[1] and Madhu Sudan[2]

1 **Department of Mathematics, IIT Bombay, India**
srikanth@math.iitb.ac.in
2 **Harvard John A. Paulson School of Engineering and Applied Sciences, USA**
madhu@cs.harvard.edu

Abstract

The well-known DeMillo-Lipton-Schwartz-Zippel lemma says that n-variate polynomials of total degree at most d over grids, i.e. sets of the form $A_1 \times A_2 \times \cdots \times A_n$, form error-correcting codes (of distance at least 2^{-d} provided $\min_i\{|A_i|\} \geq 2$). In this work we explore their local decodability and local testability. While these aspects have been studied extensively when $A_1 = \cdots = A_n = \mathbb{F}_q$ are the same finite field, the setting when A_i's are not the full field does not seem to have been explored before.

In this work we focus on the case $A_i = \{0, 1\}$ for every i. We show that for every field (finite or otherwise) there is a test whose query complexity depends only on the degree (and not on the number of variables). In contrast we show that decodability is possible over fields of positive characteristic (with query complexity growing with the degree of the polynomial and the characteristic), but not over the reals, where the query complexity must grow with n. As a consequence we get a natural example of a code (one with a transitive group of symmetries) that is locally testable but not locally decodable.

Classical results on local decoding and testing of polynomials have relied on the 2-transitive symmetries of the space of low-degree polynomials (under affine transformations). Grids do not possess this symmetry: So we introduce some new techniques to overcome this handicap and in particular use the hypercontractivity of the (constant weight) noise operator on the Hamming cube.

1998 ACM Subject Classification F.1.2 Modes of Computation, Probabilistic computation

Keywords and phrases Property testing, Coding theory, Low-degree testing, Local decoding, Local testing

Digital Object Identifier 10.4230/LIPIcs.ITCS.2018.26

1 Introduction

Low-degree polynomials have played a central role in computational complexity. (See for instance [27, 8, 5, 21, 23, 19, 28, 3, 2] for some of the early applications.) One of the key properties of low-degree n-variate polynomials underlying many of the applications is the "DeMillo-Lipton-Schwartz-Zippel" distance lemma [10, 26, 30] which upper bounds the number of zeroes that a non-zero low-degree polynomial may have over "grids", i.e., over domains of the form $A_1 \times \cdots \times A_n$. This turns the space of polynomials into an

* This work was partially supported by a Simons Investigator Award and NSF Awards CCF 1565641 and CCF 1715187.

† A full version of the paper is available at [29], https://arxiv.org/abs/1709.06036

error-correcting code (first observed by Reed [24] and Muller [20]) and many applications are built around this class of codes. These applications have also motivated a rich collection of tools including polynomial time (global) decoding algorithms for these codes, and "local decoding" [4, 18, 9] and "local testing" [25, 1, 15] procedures for these codes.

Somewhat strikingly though, many of these tools associated with these codes don't work (at least not immediately) for all grid-like domains, but work only for the specific case of the domain being the vector space $\mathbb{F}^n$ where $\mathbb{F}$ is the field over which the polynomial is defined and $\mathbb{F}$ is finite. The simplest example of such a gap in knowledge was the case of "global decoding". Here, given a function $f : \prod_{i=1}^{n} A_i \to \mathbb{F}$ as a truth-table, the goal is to find a nearby polynomial (up to half the distance of the underlying code) in time polynomial in $|\prod_i A_i|$. When the domain equals $\mathbb{F}^n$ then such algorithms date back to the 1950s. However the case of general A_i remained open till 2016 when Kim and Kopparty [17] finally solved this problem.

In this paper we initiate the study of local decoding and testing algorithms for polynomials when the domain is not a vector space. As a first step towards this we consider the case of polynomials over hypercubes i.e., when $A_i = \{0, 1\} \subseteq \mathbb{F}$ for every i. (This setting easily extends to the case where $|A_i| = 2$ for all i — see more on this at the end of Section 1.1. The setting of $|A_i| > 2$ seems to offer new challenges that we don't explore in this paper.) We describe the problems formally next and then describe our results.

1.1 Distance, Local Decoding and Local Testing

We start with some brief notation. For finite sets $A_1, \ldots, A_n \subseteq \mathbb{F}$ and functions $f, g : A_1 \times \cdots A_n \to \mathbb{F}$, let the distance between f and g, denoted $\delta(f, g)$ be the quantity $\Pr_a[f(a) \neq g(a)]$ where a is drawn uniformly from $A_1 \times \cdots \times A_n$. We say f is δ-close to g if $\delta(f, g) \leq \delta$, and δ-far otherwise. For a family of functions $\mathcal{F} \subseteq \{h : A_1 \times \cdots \times A_n \to \mathbb{F}\}$, let $\delta(\mathcal{F}) = \min_{f \neq g \in \mathcal{F}}\{\delta(f, g)\}$.

To set the context for some of the results on local decoding and testing, we first recall the distance property of polynomials. If $|A_i| \geq 2$ for every i, the polynomial distance lemma asserts that the distance between any two distinct degree d polynomials[1] is at least 2^{-d}. Of particular interest is the fact that for fixed d this distance is bounded away from 0, independent of n or $|\mathbb{F}|$ or the structure of the sets A_i. In turn this behavior effectively has led to "local decoding" and "local testing" algorithms with complexity depending only on d — we define these notions and elaborate on this sentence next.

Given a family of functions $\mathcal{F}$ from the domain $A_1 \times \cdots \times A_n$ to $\mathbb{F}$, we say $\mathcal{F}$ is (δ, q)-*locally decodable* if there exists a probabilistic algorithm that, given $a \in A_1 \times \cdots \times A_n$ and oracle access to a function $f : A_1 \times \cdots \times A_n \to \mathbb{F}$ that is δ-close to some function $p \in \mathcal{F}$, makes at most q oracle queries to f and outputs $p(a)$ with probability at least $3/4$. (The existence of a (δ, q)-local decoder for $\mathcal{F}$ in particular implies that $\delta(\mathcal{F}) \geq 2\delta$.) We say that $\mathcal{F}$ is (δ, q)-*locally testable* if there exists a probabilistic algorithm that makes q queries to an oracle for $f : A_1 \times \cdots \times A_n \to \mathbb{F}$ and accepts with probability at least $3/4$ if $f \in \mathcal{F}$ and rejects with probability at least $3/4$ if f is δ-far from every function in $\mathcal{F}$.

When $A_1 = \cdots = A_n = \mathbb{F}$ (and so $\mathbb{F}$ is finite) it was shown by Kaufman and Ron [15] (with similar results in Jutla et al. [13]) that the family of n-variate degree d polynomials over $\mathbb{F}$ is (δ, q)-locally decodable and (δ, q)-locally testable for some $\delta = \exp(-d)$ and $q = \exp(d)$.

[1] Throughout this paper we only consider the functions represented by degree d polynomials. So, without loss of generality, these maybe viewed as polynomials of degree at most $|A_i| - 1$ in the ith variable, and specifically multilinear polynomials when $|A_i| = 2$.

In particular both q and $1/\delta$ are bounded for fixed d, independent of n and $\mathbb{F}$. Indeed in both cases δ is lower bounded by a constant factor of $\delta(\mathcal{F}(n,d))$ and q is upper bounded by a polynomial in the inverse of $\delta(\mathcal{F}(n,d))$ where $\mathcal{F}(n,d)$ denotes the family of functions corresponding to degree d n-variate polynomials over $\mathbb{F}$, seemingly suggesting that the testability and decodability may be consequences of the distance. If so does this phenomenon should extend to the case of other sets $A_i \neq \mathbb{F}$ - does it? We explore this question in this paper.

In what follows we say that the family of degree d n-variate polynomials is locally decodable (resp. testable) if there is bounded $q = q(d)$ and positive $\delta = \delta(d)$ such that $\mathcal{F}(n,d)$ is (δ, q)-locally decodable (resp. testable) for every n. The specific question we address below is when are the family of degree d n-variate polynomials locally decodable and testable when the domain is $\{0,1\}^n$. (We stress that the choice of $\{0,1\}^n$ as domain is for simplicity and is equivalent to the setting of $|A_i| = 2$ for all i. Working with domains of other (and varying) sizes would lead to qualitative changes and we do not consider that setting in this paper.)

1.2 Main Results

Our first result (Theorem 3.2) shows that even the space of degree 1 polynomials is *not locally decodable* over fields of zero characteristic or over fields of large characteristic. This statement already stresses the main difference between the vector space setting (domain being $\mathbb{F}^n$) and the "grid" setting (domain = $\{0,1\}^n$). One key reason underlying this difference is that the domain $\mathbb{F}^n$ has a rich group of symmetries that preserve the space of degree d polynomials, where the space of symmetries is much smaller when the domain is $\{0,1\}^n$. Specifically the space of degree d polynomials over $\mathbb{F}^n$ is "affine-invariant" (invariant under all affine maps from $\mathbb{F}^n$ to $\mathbb{F}^n$). The richness of this group of symmetries is well-known to lead to local decoding algorithms (see for instance [1]) and this explains the local decodability of $\mathcal{F}(n,d)$ over the domain $\mathbb{F}^n$. Of course the absence of this rich group of symmetries does not rule out local decodability — and so some work has to be done to establish Theorem 3.2. We give an overview of the proof in Section 1.3 and then give the proof in Section 5.

Our second result (Theorem 3.3) shows, in contrast, that the class of *degree d polynomials over fields of small characteristic are locally decodable*. Specifically, we show that there is a $q = q(d,p) < \infty$ and $\delta = \delta(d,p) > 0$ such that $\mathcal{F}(n,d)$ over the domain $\{0,1\}^n$ over a (possibly infinite) field $\mathbb{F}$ of characteristic p is (δ, q)-locally decodable. This is perhaps the first local-decodability result for polynomials over infinite fields. A key technical ingredient that leads to this result, which may be of independent interest, is that when $n = 2p^t$ (twice a power of the characteristic of $\mathbb{F}$) and g is a degree d polynomial for $d < n/2$ then $g(0)$ can be determined from the value of g on the ball on Hamming weight $n/2$ (see Lemma 6.1). Again, we give an overview of the proof in Section 1.3 and then give the actual proof in Section 6.

Our final, and main technical, result (Theorem 3.1) shows somewhat surprisingly that $\mathcal{F}(n,d)$ *is always (i.e., over all fields) locally testable*. This leads to perhaps the simplest natural example of a locally testable code that is not locally decodable. We remark there are of course many examples of such codes (see, for instance, the locally testable codes of Dinur [11]) but these are results of careful constructions and in particular not very symmetric. On the other hand $\mathcal{F}(n,d)$ over $\{0,1\}^n$ does possess moderate symmetry and in particular the automorphism group is transitive. We remark that for both our positive results (Theorems 3.3 and 3.1), the algorithms themselves are not obvious and the analysis leads to further interesting questions. We elaborate on these in the next section.

1.3 Overview of proofs

1.3.1 Impossibility of local decoding over fields of large characteristic

In Section 5 we show that even the family of affine functions over $\{0,1\}^n$ is not locally decodable. The main idea behind this construction and proof is to show that the value of a affine function $\ell : \{0,1\}^n \to \mathbb{F}$ at 1^n can not be determined from its values on any set S if $|S|$ is small (specifically $|S| = o(\log n / \log\log n)$) and S contains only "balanced" elements (i.e., $x \in S \Rightarrow |\sum_i x_i - (n/2)| = O(\sqrt{n})$. Since the space of affine functions from $\{0,1\}^n$ to $\mathbb{F}$ forms a vector space, this in turn translates to showing that no set of up to $|S|$ balanced vectors contain the vector 1^n in their affine span (over $\mathbb{F}$) and we prove this in Lemma 5.2.

Going from the above statement to Theorem 3.2 is relatively standard in the case of finite fields. We show that if one picks a random linear function and simply erase its values on imbalanced inputs, this leads to only a small fraction of error, but its value at 1^n is not decodable with $o(\log n / \log\log n)$ queries. (Indeed many of the ingredients go back to the work of [6], who show that a canonical non-adaptive algorithm is effectively optimal for linear codes, though their results are stated in terms of local testing rather than local decoding.) In the case of infinite fields one has to be careful since one can not simply work with functions that are chosen uniformly at random. Instead we work with random linear functions with bounded coefficients. The bound on the coefficients leads to mild complications due to border effects that need care. In the proof of Theorem 5.3, we show how to overcome these complications using a counting (or encoding) argument.

The technical heart of this part is thus the proof of Lemma 5.2 and we give some idea of this proof next. Suppose $S = \{x^1, \ldots, x^t\}$ contained $x^0 = 1^n$ in its affine span and suppose $|\sum_{j=1}^n x_j^i - (n/2)| \leq n/s$ for all i. Let $a_1, \ldots, a_t \in \mathbb{F}$ be coefficients such that $x^0 = \sum_i a_i x^i$ with $\sum_i a_i = 1$. Our proof involves reasoning about the size of the coefficients $a_1, \ldots, a_t$. To get some intuition why this may help, note that

$$\frac{n}{2} = \left|\sum_{j=1}^n x_j^0 - \frac{n}{2}\right| = \left|\sum_{i=1}^t a_i \cdot \left(\sum_{j=1}^n x_j^i - \frac{n}{2}\right)\right| \leq \sum_{i=1}^t |a_i| \cdot \left|\sum_{j=1}^n x_j^i - \frac{n}{2}\right| \leq \frac{n}{s} \cdot \sum_j |a_j|.$$

So in particular if the a_j's are small, specifically if $|a_j| \leq 1$ then we conclude $t = \Omega(s)$. But what happens if large a_j's are used? To understand this, we first show that the coefficients need not be too large (as a function of t) - see Lemma 5.1, and then use this to prove Lemma 5.2. The details are in Section 5.1.

1.3.2 Local decodability over fields of small characteristic

The classical method to obtain a q-query local decoder is to find, given a target point $x^0 \in \mathbb{F}^n$, a distribution on queries $x^1, \ldots, x^q \in \mathbb{F}^n$ such that (1) $P(x^0)$ is determined by $P(x^1), \ldots, P(x^q)$ for every degree d polynomial P, and (2) the query x^i is independent of x^0 (so that an oracle f that usually equals P will satisfy $P(x^i) = f(x^i)$ for all i, with probability at least $3/4$). Classical reductions used the "2-transitivity" of the underlying space of automorphisms to guarantee that x^i is independent of x^j for every pair $i \neq j \in \{0, \ldots, q\}$ — a stronger property than required! Unfortunately, our automorphism space is not "2-transitive" but it turns out we can still find a distribution that satisfies the minimal needs.

Specifically, in our reduction we identify a parameter $k = k(p,d)$ and map each variable x_ℓ to either y_j or $1 - y_j$ for some $j = j(\ell) \in [k]$. This reduces the n-variate decoding task with oracle access to $f(x_1, \ldots, x_k)$ to a k-variate decoding task with access to the function $g(y_1, \ldots, y_k)$. Since there are only 2^k distinct inputs to g, decoding can solved with at most

2^k queries (if it can be solved at all). The choice of whether x_ℓ is mapped to y_j or $1 - y_j$ is determined by x^0_j so that $f(x^0) = g(0^k)$. Thus given x^0, the only randomness is in the choice of $j(\ell)$. We choose $j(\ell)$ uniformly and independently from $[k]$ for each ℓ. For $y \in \{0,1\}^k$, x^y denote the corresponding query in $\{0,1\}^n$ (i.e., $g(y) = f(x^y)$). Given our choices, x^y is not independent of x^0 for every choice of y. Indeed if y has Hamming weight 1, then x^y is very likely to have Hamming distance $\approx n/k$ from x^0 which is far from independent. However if $y \in \{0,1\}^k$ is a balanced vector with exactly $k/2$ 1s (so in particular we will need k to be even), then it turns out x^y is indeed independent of x^0. So we query only those x^y for which y is balanced. But this leads to a new challenge: can $P(0^k)$ be determined from the values of $P(y)$ for balanced ys? It turns out that for a careful choice of k (and this is where the small characteristic plays a role) the value of a degree d polynomial at 0 is indeed determined by its values on balanced inputs (see Lemma 6.1) and this turns out to be sufficient to build a decoding algorithm over fields of small characteristic. Details may be found in Section 6.

1.3.3 Local testability over all fields

We now turn to the main technical result of the paper, namely the local testability of polynomials over grids. All previous analyses of local testability of polynomials with query complexity independent of the number of variables have relied on symmetry either implicitly or explicitly. (See for example [16] for further elaboration.) Furthermore many also depend on the local decodability explicitly; and in our setting we seem to have insufficient symmetry and definitely no local decodability. This forces us to choose the test and analysis quite carefully.

It turns out that among existing approaches to analyses of local tests, the one due to Bhattacharyya et al [7] (henceforth BKSSZ) seems to make the least use of local decodability and our hope is to be able to simulate this analysis in our case — but the question remains: "which tester should we use?". This is a non-trivial question since the BKSSZ test is a natural one in a setting with sufficient symmetry; but their analysis relies crucially on the ability to view their test as a sequence of restrictions: Given a function $f : \mathbb{F}^n \to \mathbb{F}$ they produce a sequence of functions $f = f_n, f_{n-1}, \ldots, f_k$, where the function f_r is an r-variate function obtained by restricting f_{r+1} to a codimension one affine subspace. Their test finally checks to see if f_k is a degree d polynomial. To emulate this analysis, we design a somewhat artificial test: We also produce a sequence of functions $f_n, f_{n-1}, \ldots, f_k$ with f_r being an r-variate function. Since we do not have the luxury to restrict to arbitrary subspaces, we instead derive f_r from $f_{r+1}(z_1, \ldots, z_{r+1})$ by setting $z_i = z_j$ or $z_i = 1 - z_j$ for some random pair i, j (since these are the only simple affine restrictions that preserve the domain). We stop when the number of variables k is small enough (and hopefully a number depending on d alone and not on n or $\mathbb{F}$). We then test that the final function has degree d.

The analysis of this test is not straightforward even given previous works, but we are able to adapt the analyses to our setting. Two new ingredients that appear in our analyses are the hypercontractivity of hypercube with the constant weight noise operator (analyzed by Polyanskiy [22]) and the intriguing stochastics of a random set-union problem. We explain our analysis and where the above appear next.

We start with the part which is more immediate from the BKSSZ analysis. This corresponds to a key step in the BKSSZ analysis where it is shown that if f_{r+1} is far from degree d polynomials then, with high probability, so also is f_r. This step is argued via contradiction. If f_r is close to the space of degree d polynomials for many restrictions, then from the many polynomials that agree with f_r (for many of the restrictions) one can glue together an $r+1$-variate polynomial that is close to f_{r+1}. This step is mostly algebraic and works out in our case also; though the actual algebra is different and involves more cases.

The new part in our analysis is in the case where f_n is moderately close to some low-degree polynomial P. In this case we would still like to show that the test rejects f_n with positive probability. In both BKSSZ and in our analysis this is shown by showing the the 2^k queries into f_n (that given the entire truth table of the function f_k) satisfy the property that exactly f_n is not equal to P on exactly one of the queried points. Note that the value of $f_k(y)$ is obtained by querying f at some point, which we denote x^y. In the BKSSZ analysis x^a and x^b are completely independent given $a \neq b \in \{0,1\}^k$. (Note that the mapping from y to x^y is randomized and depends on the random choices of the tester.) In our setting the behavior of x^a and x^b is more complex and depends on both the set of coordinates j such that where $a_j \neq b_j$ and on the number of indices $i \in [n]$ such that the variable x_i is mapped to variable y_j. Our analysis ends up depending on two new ingredients: (1) The number of variables x_i that map to any particular variable y_j is $\Omega(n/k)$ with probability at least $2^{-O(k)}$. This part involves the analysis of a random set-union process elaborated on below. (2) Once the exact number of indices i such that x_i maps to y_j is fixed for every $j \in [k]$ and none of the sets is too small, the distribution of x^a and x^b is sufficiently independent to ensure that the events $f(x^a) = P(x^a)$ and $f(x^b) = P(x^b)$ co-occur with probability much smaller than the individual probabilities of these events. This part uses the hypercontractivity of the hypercube but under an unusual noise operator corresponding to the "constant weight operator", fortunately analyzed by Polyanskiy [22]. Invoking his theorem we are able to conclude the proof of this section.

We now briefly expand on the "random set-union" process alluded to above. Recall that our process starts with n variables, and at each stage a pair of remaining variables is identified and given the same name. (We may ignore the complications due to the complementation of the form $z_i = 1 - z_j$ for this part.) Equivalently we start with n sets $X_1, \ldots, X_n$ with $X_i = \{i\}$ initially. We then pick two random sets and merge them. We stop when there are k sets left and our goal is to understand the likelihood that one of the sets turn out to be too tiny. (The expected size of a set is n/k and too tiny corresponds to being smaller than $n/(4k)$.) It turns out that the distribution of set sizes produced by this process has a particularly clean description as follows: Randomly arrange the elements 1 to n on a cycle and consider the partition into k sets generated by the set of elements that start with a special element and end before the next special element as we go clockwise around the cycle, where the elements in $\{1, \ldots, k\}$ are the special ones. The sizes of these partitions are distributed identically to the sizes of the sets S_j! For example, when $k = 2$ the two sets have sizes distributed uniformly from 1 to $n - 1$. In particular the sets size are not strongly concentrated around n/k - but nevertheless the probability that no set is tiny is not too small and this suffices for our analysis.

Details of this analysis may be found in Section 4.

Organization

In Section 2 we start with some preliminaries including the main definitions and some of the tools we will need later. In Section 3 we give a formal statement of our results. In Section 4 we present the local tester over all fields. In Section 5 we sketch our proof that over fields of large (or zero) characteristic, local decoding is not possible. Finally in Section 6 we give a local decoder over fields of small characteristic. Most analysis is omitted from this version and included in the full version of this paper [29].

2 Preliminaries

2.1 Basic notation

Fix a field $\mathbb{F}$ and an $n \in \mathbb{N}$. We consider functions $f : \{0,1\}^n \to \mathbb{F}$ that can be written as *multilinear* polynomials of total degree at most d. We denote this space by $\mathcal{F}(n, d; \mathbb{F})$. The space of all functions from $\{0,1\}^n$ to $\mathbb{F}$ will be denoted simply as $\mathcal{F}(n; \mathbb{F})$. (We will simplify these to $\mathcal{F}(n, d)$ and $\mathcal{F}(n)$ respectively, if the field $\mathbb{F}$ is clear from context.)

Given $f, g \in \mathcal{F}(n)$, we use $\delta(f, g)$ to denote the fractional Hamming distance between f and g. I.e.,

$$\delta(f, g) := \Pr_{x \in \{0,1\}^n} [f(x) \neq g(x)]$$

For a family $\mathcal{F}' \subseteq \mathcal{F}(n)$, we use $\delta(f, \mathcal{F}')$ to denote $\min_{g \in \mathcal{F}'}\{\delta(f, g)\}$. Given an $f \in \mathcal{F}(n)$ and $d \geq 0$, we use $\delta_d(f)$ to denote $\delta(f, \mathcal{F}(n, d))$.

2.2 Local Testers and Decoders

Let $\mathbb{F}$ be any field. We define the notion of a local tester and local decoder for subspaces of $\mathcal{F}(n)$. These notions go back at least to the works of Goldreich and Sudan [12] and Katz and Trevisan [14], though the exact definitions and paramaters may differ here.

▶ **Definition 2.1** (Local tester)**.** Fix $q \in \mathbb{N}$ and $\delta \in (0, 1)$. Let $\mathcal{F}'$ be any subspace of $\mathcal{F}(n)$.

We say that a randomized algorithm T is a (δ, q)-*local tester for* $\mathcal{F}'$ if on an input $f \in \mathcal{F}(n)$, the algorithm does the following.

- T makes at most q queries to f and either accepts or rejects.
- (Completeness) If $f \in \mathcal{F}'$, then T accepts with probability at least $3/4$.
- (Soundness) If $\delta(f, \mathcal{F}') \geq \delta$, then T rejects with probability at least $3/4$.

We say that a tester is *adaptive* if the queries it makes to the input f depend on the answers to its earlier queries. Otherwise, we say that the tester is *non-adaptive.*

▶ **Definition 2.2** (Local decoder)**.** Fix $q \in \mathbb{N}$ and $\delta \in (0, 1)$. Let $\mathcal{F}'$ be any subspace of $\mathcal{F}(n)$.

We say that a randomized algorithm T is a (δ, q)-*local decoder for* $\mathcal{F}'$ if on an input $f \in \mathcal{F}(n)$ and $x \in \{0,1\}^n$, the algorithm does the following.

- T makes at most q queries to f and outputs $b \in \mathbb{F}$.
- If $\delta(f, \mathcal{F}') \leq \delta$, then the output $b = f(x)$ with probability at least $3/4$.

We say that a decoder is *adaptive* if the queries it makes to the input f depend on the answers to its earlier queries. Otherwise, we say that the tester is *non-adaptive.*

2.3 Some basic facts about binomial coefficients

▶ **Fact 2.3.** *For integer parameters $0 \leq b \leq a$, let $\binom{a}{\leq b}$ denote the size of a Hamming ball of radius b in $\{0,1\}^a$; equivalently, $\binom{a}{\leq b} = \sum_{j \leq b} \binom{a}{j}$. Then, we have*

$$\binom{a}{\leq b} \leq 2^{aH(b/a)}$$

where $H(\cdot)$ is the binary entropy function.

2.4 Hypercontractivity theorem for spherical averages.

In this section, let $\mathbb{R}$ be the underlying field. Let $\eta \in (0,1)$ be arbitrary. We define a smoothing operator T_η, which maps $\mathcal{F}(r) = \{f : \{0,1\}^r \to \mathbb{R}\}$ to itself. For $F \in \mathcal{F}(r)$, we define $T_\eta F$ as follows

$$T_\eta F(x) = \mathop{\mathbf{E}}_{J \in \binom{[r]}{\eta r}} [F(x \oplus J)]$$

where $x \oplus J$ is the point $y \in \{0,1\}^r$ obtained by flipping x at exactly the coordinates in J.

Recall that for any $F \in \mathcal{F}(r)$ and any $p \geq 1$, $\|F\|_p$ denotes $\mathbf{E}_{x \in \{0,1\}^r} \left[|F(x)|^p\right]^{1/p}$.

We will use the following hypercontractivity theorem of Polanskiy [22].

▶ **Theorem 2.4** (Follows from Theorem 1 in [22]). *Assume that $\eta \in [1/20, 19/20]$ and $\eta_0 = 1/20$. For any $F \in \mathcal{F}(r)$, we have*

$$\|T_\eta F\|_2 \leq C \cdot \|F\|_p$$

for $p = 1 + (1 - 2\eta_0)^2$ and C is an absolute constant.

▶ **Corollary 2.5.** *Assume that η_0, η are as in the statement of Theorem 2.4 and let $\delta \in (0,1)$ be arbitrary. Say $E \subseteq \{0,1\}^r$ s.t. $|E| \leq \delta \cdot 2^r$. Assume that $(x', x'') \in \{0,1\}^r$ are chosen as follows: $x' \in \{0,1\}^r$ and $I' \in \binom{[r]}{\eta r}$ are chosen i.u.a.r., and we set $x'' = x' \oplus I'$. Then we have*

$$\Pr_{x', I'} [x' \in E \wedge x'' \in E] \leq C \cdot \delta^{1+(1/40)}$$

where C is the constant from Theorem 2.4.

Proof. Let $F : \{0,1\}^n \to \{0,1\} \subseteq \mathbb{R}$ be the indicator function of the set E. Note that we have

$$\Pr_{x', I'} [x' \in E \wedge x'' \in E] = \mathop{\mathbf{E}}_{x', I'} [F(x')F(x' \oplus I')] = \mathop{\mathbf{E}}_{x'} [F(x') T_\eta F(x')] .$$

By the Cauchy-Schwarz inequality and Theorem 2.4 we get

$$\mathop{\mathbf{E}}_{x'} [F(x') T_\eta F(x')] \leq \|F\|_2 \cdot C \cdot \|F\|_p \tag{1}$$

for $p = 1 + (1 - 2\eta_0)^2$. Note that we have

$$\begin{aligned} \|F\|_p &\leq \delta^{1/p} = \delta^{\frac{1}{1+(1-2\eta_0)^2}} \\ &= \delta^{\frac{1}{2(1-2\eta_0(1-\eta_0))}} \leq (\sqrt{\delta})^{1+\min\{\eta_0, 1-\eta_0\}} = \sqrt{\delta}^{1+(1/20)} \end{aligned}$$

where for the last inequality we have used the fact that for $\eta_0 \in [0,1]$ we have

$$\frac{1}{1 - 2\eta_0(1-\eta_0)} \geq 1 + 2\eta_0(1-\eta_0) \geq 1 + \min\{\eta_0, 1-\eta_0\}.$$

Putting the upper bound on $\|F\|_p$ together with the fact that $\|F\|_2 \leq \sqrt{\delta}$ and (1), we get the claim. ◀

3 Results

We show upper and lower bounds for testing and decoding polynomial codes over grids. All our upper bounds hold in the non-adaptive setting, while our lower bounds hold in the stronger adaptive setting.

Our first result is that for any choice of the field $\mathbb{F}$ (possibly even infinite), the space of functions $\mathcal{F}(n,d)$ is locally testable. More precisely, we show the following.

▶ **Theorem 3.1** ($\mathcal{F}(n,d)$ has a local tester for any field)**.** *There exists a constant $c < \infty$ and polynomial $p_0(x)$ such that the following holds for every field $\mathbb{F}$, every non-negative integer d, every positive integer n and every real number $\varepsilon > 0$: The space $\mathcal{F}(n,d;\mathbb{F})$ has a non-adaptive (ε, q)-local tester for $q \leq 2^{c\cdot d} \cdot p_0(1/\varepsilon)$.*

In contrast, we show that the space $\mathcal{F}(n,d)$ is *not* locally *decodable* over fields of large characteristic, even for $d = 1$.

▶ **Theorem 3.2** ($\mathcal{F}(n,d)$ does not have a local decoder for large characteristic)**.** *For every $\varepsilon > 0$ there exists $c_\varepsilon > 0$ such that the following holds: Let $n \in \mathbb{N}$ and let $\mathbb{F}$ be a field such that either $\mathrm{char}(\mathbb{F}) = 0$ or $\mathrm{char}(\mathbb{F}) \geq n^2$. Then any adaptive (ε, q)-local decoder for $\mathcal{F}(n,1;\mathbb{F})$ must satisfy $q \geq c_\varepsilon \cdot \log n / \log\log n$.*

Complementing the above result, we can show that if $\mathrm{char}(\mathbb{F})$ is a constant, then in fact the space $\mathcal{F}(n,d)$ does have a local decoding procedure.

▶ **Theorem 3.3** ($\mathcal{F}(n,d)$ has a local decoder for constant characteristic)**.** *There exists a constant $c < \infty$ such that for every field $\mathbb{F}$ of characteristic p, every non-negative integer d and every positive integer n, the space $\mathcal{F}(n,d;\mathbb{F})$ has a non-adaptive $(2^{-c\cdot p\cdot d}, 4^{p\cdot d})$-local decoder.*

4 A local tester for $\mathcal{F}(n,d)$ over any field

We now present our local tester and its analysis. The reader may find the overview from Section 1.3 helpful while reading the below.

We start by introducing some notation for this section. Throughout, fix any field $\mathbb{F}$. We consider functions $f : \{0,1\}^I \to \mathbb{F}$ where I is a finite set of positive integers and indexes into the set of variables $\{X_i \mid i \in I\}$. We denote this space as $\mathcal{F}(I)$. Similarly, $\mathcal{F}(I,d)$ is defined to be the space of functions of degree at most d over the variables indexed by I.

The following is the test we use to check if a given function $f : \{0,1\}^I \to \mathbb{F}$ is close to $\mathcal{F}(I,d)$.

Test $T_{k,I}(f_I)$

Notation. Given two variables X and Y and $a \in \{0,1\}$, "replacing X by $a \oplus Y$" refers to substituting X by Y if $a = 0$ and by $1 - Y$ if $a = 1$.

- If $|I| > k$, then
  - Choose a random $a \in \{0,1\}$ and distinct $i_0, j_0 \in I$ at random and replace X_{j_0} by $a \oplus X_{i_0}$. Let f'_I denote the resulting restriction of f_I.
  - Run $T_{k,I\setminus\{j_0\}}(f'_I)$ and output what it outputs.
- If $|I| = k$ then
  - Choose a uniformly random bijection $\sigma : I \to [k]$.

- Choose an $a \in \{0,1\}^k$ uniformly at random.
- Replace each X_i $(i \in I)$ with $Y_{\sigma(i)} \oplus a_i$.
- Check if the restricted function $g(Y_1, \ldots, Y_k) \in \mathcal{F}(k,d)$ by querying g on all its inputs. Accept if so and reject otherwise.

▶ Remark. It is not strictly necessary to choose a *random* bijection σ in the test $T_{k,I}$ and a fixed bijection $\sigma : I \to [k]$ would do just as well. However, the above leads to a cleaner reformulation of the test.

▶ **Observation 4.1.** *Test $T_{k,I}$ has query complexity 2^k.*

▶ **Observation 4.2.** *If $f_I \in \mathcal{F}(I,d)$, then $T_{k,I}$ accepts with probability 1.*

The following theorem is the main result of this section and implies Theorem 3.1 from Section 3.

▶ **Theorem 4.3.** *For each positive integer d, there is a $k = O(d)$ and $\varepsilon_0 = 1/2^{O(d)}$ such that for any I of size at least $k+1$ and any $f_I \in \mathcal{F}(I)$,*

$$\Pr\left[\textit{Test } T_{k,I} \textit{ rejects } f_I\right] \geq \frac{1}{2^{O(d)}} \cdot \min\{\delta_d(f_I), \varepsilon_0\}.$$

Theorem 3.1 immediately follows from Theorem 4.3 since to get an $(\varepsilon, 2^{O(d)})$-tester, we repeat the test $T_{k,[n]}$ $t = 2^{O(d)} \cdot \mathrm{poly}(1/\varepsilon)$ many times and accept if and only if each iteration of the test accepts. If the input function $f \in \mathcal{F}(n)$ is of degree at most d, this test accepts with probability 1. Otherwise, this test rejects with probability at least $3/4$ for suitably chosen t as above. The number of queries made by the test is $2^k \cdot t = 2^{O(d)} \cdot \mathrm{poly}(1/\varepsilon)$.

5 Impossibility of local decoding when $\mathrm{char}(\mathbb{F})$ is large

In this section, we prove Theorem 5.3 which is a more detailed version of Theorem 3.2. Again we remind the reader that an overview may be found in Section 1.3.

Let n be a growing parameter and $\mathbb{F}$ a field of characteristic 0 or positive characteristic greater than n^2. For the results in this section, it will be easier to deal with the domain $\{-1,1\}^n$ rather than $\{0,1\}^n$. Since there a natural invertible linear map that maps $\{0,1\}$ to $\{-1,1\}$ (i.e. $a \mapsto 1-2a$), this change of input space is without loss of generality.

5.1 Local linear spans of balanced vectors

Let $u \in \mathbb{F}^n$ and $U \subseteq \mathbb{F}^n$. For any integer $t \in \mathbb{N}$, we say that u is in the t-span of U if it can be written as a linear combination of at most t elements of U. For $x \in \{-1,1\}^n$, we use $|x|$ to denote the sum of the entries of x over $\mathbb{Z}$. In this section, we wish to show that if the vector 1^n is in the t-span of balanced vectors, i.e., vectors x with $|x| \leq n/s$ then t is must be growing as a function of s.

As explained earlier we first establish a bound on the size of the solutions of linear equations in systems over $\mathbb{Q}$ with few variables or few constraints. This fact is well-known, but we prove it here for completeness.

▶ **Lemma 5.1.** *Let $r, s \in \mathbb{N}$ and let $t = \min\{r,s\}$. Let $Mx = u$ be a system of linear equations with $M \in \{-1,0,1\}^{r \times s}$ and $u \in \{-1,0,1\}^r$.*

- *If $\mathbb{F}$ is a field of characteristic zero and the system has a solution in $\mathbb{F}^s$, then there exist integers $a_1, \ldots, a_s, b \in \mathbb{Z}$ with $|a_i|, |b| \leq t!$ such that $x_i = a_i/b$ is a solution to $Mx = u$. In particular, there is a solution in $\mathbb{Q}^s$.*

- *If $\mathbb{F}$ is a field of characteristic p and if the system has a solution in $\mathbb{F}^s$, then there exist integers $a_1, \ldots, a_s, b \in \mathbb{Z}$ with $|a_i|, |b| \leq t!$ such that $x_i = a_i/b \pmod p$ is a solution to $Mx = u$. In particular, there is a solution in $\mathbb{F}_p^s$.*

Proof omitted from this version.

We now turn to the main technical lemma of this section showing that 1^n is not in linear span of a small number of nearly balanced elements of $\{-1, 1\}^n$.

▶ **Lemma 5.2.** *Let $n, s = s(n) \in \mathbb{N}$ with $s(n) \leq n$. Let $S = \{x \in \{-1, 1\}^n \mid |x| \leq n/s)\}$. Then $x^0 = 1^n$ is not in the t-span of S unless $t \geq \log s / \log\log s$ provided $\mathbb{F}$ is field of zero characteristic or of characteristic $p \geq 2n^2$.*

Proof. We first consider the case when $\mathbb{F}$ is of zero characteristic. Note that in this case $\mathbb{Q} \subseteq \mathbb{F}$. Suppose $x^0 \in \text{Span}\{x^1, \ldots, x^t\}$ with $x^0 = \sum_{i=1}^n c_i x^i$. Note that the c_i's are expressible as the solution to a linear system whose $Mz = u$ where M and u have entries in $\{-1, 0, 1\}$ and M is a $n \times t$ matrix. By Lemma 5.1 we have that $c_i \in \mathbb{Q}$ with $|c_i| \leq t!$ (more specifically we have $c_i = a_i/b$ with $|a_i| \leq t!$ and this implies $|c_i| \leq t!$). We thus have

$$n = \left|\sum_{j=1}^n x_j^0\right| = \left|\sum_{i=1}^t c_i \sum_{j=1}^n x_j^i\right| \leq \sum_{i=1}^t |c_i| \cdot \left|\sum_{j=1}^n x_j^i\right| \leq \sum_{i=1}^t (t!) \cdot (n/s) \leq (t+1)! \cdot (n/s).$$

We thus conclude that $(t+1)! \geq s$ and thus $t \geq \log s / \log\log s$.

In the case of finite field $\mathbb{F}$, we proceed as above and let $x^0 = \sum_{i=1}^t c_i x^i$. By Lemma 5.1 we have that there are integers a_i, b with $|a_i|, |b| \leq t!$ such that $c_i = a_i/b \pmod p$ is a solution to $x^0 = \sum_{i=1}^t c_i x^i$. Now consider $b \cdot n$ and we get $b \cdot n = \sum_{i=1}^t a_i \sum_{j=1}^n x_j^i \pmod p$. We now show that this implies $(t+1)! \geq \min\{p/(2n), s\} = s$ (where the equality follows from $p \geq 2n^2$ and $s \leq n$). Assume $(t+1)! \leq p/(2n)$. Then we have $n \leq |b \cdot n| \leq t! \cdot n < p/2$ over the integers, and $\left|\sum_{i=1}^t a_i \sum_{j=1}^n x_j^i\right| \leq (t+1)!(n/s) < p/2$ also over the integers. We again conclude that $n \leq (t+1)!(n/s)$ and so $(t+1)! \geq s$ as claimed. The lemma follows. ◀

We now state the main result of this section which immediately implies Theorem 3.2.

▶ **Theorem 5.3.** *Let $n \in \mathbb{N}$ be a growing parameter and $\varepsilon \in (0, 1)$ such that $\varepsilon \geq 2\exp(-n/2s^2)$ for some $s \in \mathbb{N}$ with $100 \leq s \leq \sqrt{n}/100$. Let $\mathbb{F}$ be any field such that either $\text{char}(\mathbb{F}) = 0$ or $\text{char}(\mathbb{F}) \geq n^2$. Then any adaptive (ε, q)-local decoder for $\mathcal{F}(n, 1)$ that corrects an ε fraction of errors must satisfy $q = \Omega(\log s / \log\log s)$.*

Proof omitted from this version.

6 Local decoding when $\text{char}(\mathbb{F})$ is small

In this section, we give a local decoder over fields of small characteristic. An overview of this construction may be found in Section 1.3.

Let p be a prime of constant size and let $\mathbb{F}$ be any (possibly infinite) field of characteristic p. Let d be the degree parameter and k be the smallest power of p that is strictly greater than d. Note that $k \leq pd$. We show that the space $\mathcal{F}(n, d)$ has a $(1/(4 \cdot \binom{2k}{k}), \binom{2k}{k})$-local decoder, hence proving Theorem 3.3.

The main technical tool we use is a suitable linear relation on the space $\mathcal{F}(2k, d)$, which we describe now. We say that a set $S \subseteq \{0, 1\}^{2k}$ is *useful* if for every polynomial $G \in \mathcal{F}(2k, d)$, $G(0^{2k})$ is determined by the restriction of the function G to the inputs in S. Let $B \subseteq \{0, 1\}^{2k}$ denote the set of all balanced inputs (i.e. inputs of Hamming weight exactly k).

▶ **Lemma 6.1.** *Fix d, k as above. Then the set $B \subseteq \{0,1\}^{2k}$ of balanced inputs is useful.*

The proof of the above lemma will use Lucas' theorem, which we recall below.

▶ **Theorem 6.2** (Lucas' theorem)**.** *Let p be any prime and $a, b \in \mathbb{N}$. Let $a_1, \ldots, a_\ell \in \{0, \ldots, p-1\}$ and $b_1, \ldots, b_\ell \in \{0, \ldots, p-1\}$ be the digits in the p-ary expansion of a and b, i.e., $a = \sum_{j \in [\ell]} a_j p^{j-1}$ and $b = \sum_{j \in [\ell]} b_j p^{j-1}$. Then, we have*

$$\binom{a}{b} \equiv \prod_{i \leq \ell} \binom{a_i}{b_i} \pmod{p}$$

where $\binom{a_i}{b_i}$ is defined to be 0 if $a_i < b_i$.

▶ **Corollary 6.3.** *For $i \in \{0, \ldots, d\}$, we have $\binom{d+k-i}{k-i} \not\equiv 0 \pmod{p}$ if and only if $i = 0$.*

Proof omitted from this version.

Proof of Lemma 6.1. Fix any $G \in \mathcal{F}(2k, d)$. Assume that

$$G(Y_1, \ldots, Y_{2k}) = \sum_{I \subseteq [2k] : |I| \leq d} \alpha_I Y^I$$

where Y^I denotes $\prod_{i \in I} Y_i$.

Let B' denote all those inputs in B where the last $k-d$ bits are set to 0. We will compute the sum of G on inputs from B'. But let us first consider a monomial Y^I and see what its sum over $y \in B'$ looks like. The monomial evaluates to 1 on $y \in B'$ if $y_i = 1$ for every $i \in I$, and evaluates to 0 otherwise. There are exactly $\binom{d+k-|I|}{k-|I|}$ choices of $y \in B'$ that satisfy $y_i = 1$ for every $i \in I$. Thus summing over $y \in B'$ we get $\sum_{y \in B'} y^I = \binom{d+k-|I|}{k-|I|}$. Summing over all monomials we get:

$$\begin{aligned} \sum_{y \in B'} G(y) &= \sum_{I \subseteq [2k] : |I| \leq d} \alpha_I \cdot \sum_{y \in B'} Y^I \\ &= \sum_{I \subseteq [2k] : |I| \leq d} \alpha_I \cdot \binom{d+k-|I|}{k-|I|} \end{aligned} \tag{2}$$

By Corollary 6.3, it follows that for $i \in \{0, \ldots, d\}$, we have

$$\binom{d+k-i}{k-i} \not\equiv 0 \pmod{p}$$

if and only if $i = 0$ and so $\sum_{y \in B'} G(y) = \binom{d+k}{k} \cdot \alpha_\emptyset$. Let $c = \binom{d+k}{k} \pmod{p}$. We have $c \in \mathbb{F}_p^* \subseteq \mathbb{F}^*$ and in particular c is invertible in $\mathbb{F}$, and $\sum_{y \in B'} G(y) = c \cdot \alpha_\emptyset = c \cdot G(0^{2k})$. Hence, we get $G(0^{2k}) = c^{-1} \cdot \sum_{y \in B'} G(y)$. Therefore, $G(0^{2k})$ is determined by the restriction of G to B' and hence also by its restriction to B. ◀

We now show that $\mathcal{F}(n, d)$ has a $(1/(4 \cdot \binom{2k}{k})), \binom{2k}{k})$-local decoder.

The decoder. We now give the formal description of the decoder. Let the decoder be given oracle access to f with the promise that f is $1/(4 \cdot \binom{2k}{k})$-close to some $F \in \mathcal{F}(n, d)$. Let the input to the decoder be $x \in \{0,1\}^n$. The problem is to find $F(x)$.

We describe the decoder below:

Decoder $D_k^f(x)$.

- Partition $[n]$ into $2k$ parts by choosing a *uniformly* random map $h : [n] \to [2k]$. I.e. each $h(j)$ is chosen i.u.a.r. from $[2k]$.
- For $i \in [2k]$ and $j \in [n]$ such that $h(j) = i$, identify X_j with $Y_i \oplus x_j$.
- Let $g(Y_1, \ldots, Y_{2k})$ and $G(Y_1, \ldots, Y_{2k})$ be the restrictions of f and F respectively. Assuming $g|_B = G|_B$, query g at all inputs in B and decode $G(0^{2k})$ from $G|_B$. Output the value decoded.

The main theorem of this section is the following. Note that this implies Theorem 3.3.

▶ **Theorem 6.4.** *Let $\mathbb{F}$ be a field of characteristic p. For integer $d \geq 0$, let k be the smallest power of p greater than d. Then the decoder D_k is a $(1/(4 \cdot \binom{2k}{k}), \binom{2k}{k})$-local decoder for $\mathcal{F}(n, d; \mathbb{F})$.*

Proof omitted from this version.

References

1. Noga Alon, Tali Kaufman, Michael Krivelevich, Simon Litsyn, and Dana Ron. Testing reed-muller codes. *IEEE Trans. Information Theory*, 51(11):4032–4039, 2005. doi:10.1109/TIT.2005.856958.
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. *J. ACM*, 45(1):70–122, 1998. doi:10.1145/273865.273901.
4. Donald Beaver and Joan Feigenbaum. Hiding instances in multioracle queries. In C. Choffrut and T. Lengauer, editors, *Proceedings of the 7th Annual Symposium on Theoretical Aspects of Computer Science*, pages 37–48, Rouen, France, 22–24 February 1990. Springer. Lecture Notes in Computer Science, Volume 415.
5. Michael Ben-Or, Shafi Goldwasser, and Avi Wigderson. Completeness theorems for non-cryptographic fault-tolerant distributed computation (extended abstract). In Janos Simon, editor, *Proceedings of the 20th Annual ACM Symposium on Theory of Computing, May 2-4, 1988, Chicago, Illinois, USA*, pages 1–10. ACM, 1988. doi:10.1145/62212.62213.
6. Eli Ben-Sasson, Prahladh Harsha, and Sofya Raskhodnikova. Some 3cnf properties are hard to test. *SIAM J. Comput.*, 35(1):1–21, 2005. doi:10.1137/S0097539704445445.
7. 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. IEEE Computer Society, 2010. doi:10.1109/FOCS.2010.54.
8. Manuel Blum, Ashok K. Chandra, and Mark N. Wegman. Equivalence of free boolean graphs can be decided probabilistically in polynomial time. *Inf. Process. Lett.*, 10(2):80–82, 1980. doi:10.1016/S0020-0190(80)90078-2.
9. 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.
10. Richard A. DeMillo and Richard J. Lipton. A probabilistic remark on algebraic program testing. *Inf. Process. Lett.*, 7(4):193–195, 1978. doi:10.1016/0020-0190(78)90067-4.
11. Irit Dinur. The PCP theorem by gap amplification. *J. ACM*, 54(3):12, 2007. doi:10.1145/1236457.1236459.
12. 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.

13 Charanjit S. Jutla, Anindya C. Patthak, Atri Rudra, and David Zuckerman. Testing low-degree polynomials over prime fields. *Random Struct. Algorithms*, 35(2):163–193, 2009. doi:10.1002/rsa.20262.

14 Jonathan Katz and Luca Trevisan. On the efficiency of local decoding procedures for error-correcting codes. 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 80–86. ACM, 2000. doi:10.1145/335305.335315.

15 Tali Kaufman and Dana Ron. Testing polynomials over general fields. *SIAM J. Comput.*, 36(3):779–802, 2006. doi:10.1137/S0097539704445615.

16 Tali Kaufman and Madhu Sudan. Algebraic property testing: the role of invariance. In Cynthia Dwork, editor, *Proceedings of the 40th Annual ACM Symposium on Theory of Computing, Victoria, British Columbia, Canada, May 17-20, 2008*, pages 403–412. ACM, 2008. doi:10.1145/1374376.1374434.

17 John Y. Kim and Swastik Kopparty. Decoding reed-muller codes over product sets. In Ran Raz, editor, *31st Conference on Computational Complexity, CCC 2016, May 29 to June 1, 2016, Tokyo, Japan*, volume 50 of *LIPIcs*, pages 11:1–11:28. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2016. doi:10.4230/LIPIcs.CCC.2016.11.

18 Richard Lipton. New directions in testing. In *Distributed Computing and Cryptography*, volume 2 of *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, pages 191–202. AMS, 1991.

19 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.

20 D. E. Muller. Application of Boolean algebra to switching circuit design and to error detection. *IEEE Transactions on Computers*, 3:6–12, 1954.

21 Ketan Mulmuley, Umesh V. Vazirani, and Vijay V. Vazirani. Matching is as easy as matrix inversion. *Combinatorica*, 7(1):105–113, 1987. doi:10.1007/BF02579206.

22 Yury Polyanskiy. Hypercontractivity of spherical averages in Hamming space. *CoRR*, abs/1309.3014, 2013.

23 Alexander A. Razborov. On the method of approximations. In David S. Johnson, editor, *Proceedings of the 21st Annual ACM Symposium on Theory of Computing, May 14-17, 1989, Seattle, Washigton, USA*, pages 167–176. ACM, 1989. doi:10.1145/73007.73023.

24 Irving S. Reed. A class of multiple-error-correcting codes and the decoding scheme. *IEEE Transactions on Information Theory*, 4:38–49, 1954.

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

26 Jacob T. Schwartz. Fast probabilistic algorithms for verification of polynomial identities. *J. ACM*, 27(4):701–717, 1980. doi:10.1145/322217.322225.

27 Adi Shamir. How to share a secret. *Commun. ACM*, 22(11):612–613, 1979. doi:10.1145/359168.359176.

28 Adi Shamir. Ip=pspace. In *31st Annual Symposium on Foundations of Computer Science, St. Louis, Missouri, USA, October 22-24, 1990, Volume I*, pages 11–15. IEEE Computer Society, 1990. doi:10.1109/FSCS.1990.89519.

29 Srikanth Srinivasan and Madhu Sudan. Local decoding and testing of polynomials over grids. *CoRR*, abs/1709.06036, 2017. arXiv:1709.06036.

30 Richard Zippel. Probabilistic algorithms for sparse polynomials. In Edward W. Ng, editor, *Symbolic and Algebraic Computation, EUROSAM '79, An International Symposiumon Symbolic and Algebraic Computation, Marseille, France, June 1979, Proceedings*, volume 72 of *Lecture Notes in Computer Science*, pages 216–226. Springer, 1979. doi:10.1007/3-540-09519-5_73.

Relaxed Locally Correctable Codes*

Tom Gur[†1], Govind Ramnarayan[‡2], and Ron D. Rothblum[§3]

1 **UC Berkeley, CA, USA**
`tom.gur@berkeley.edu`
2 **MIT, MA, USA**
`govind@mit.edu`
3 **MIT and Northeastern University, MA, USA**
`ronr@csail.mit.edu`

Abstract

Locally decodable codes (LDCs) and locally correctable codes (LCCs) are error-correcting codes in which individual bits of the message and codeword, respectively, can be recovered by querying only few bits from a noisy codeword. These codes have found numerous applications both in theory and in practice.

A natural relaxation of LDCs, introduced by Ben-Sasson *et al.* (SICOMP, 2006), allows the decoder to reject (i.e., refuse to answer) in case it detects that the codeword is corrupt. They call such a decoder a *relaxed decoder* and construct a constant-query relaxed LDC with almost-linear blocklength, which is sub-exponentially better than what is known for (full-fledged) LDCs in the constant-query regime.

We consider an analogous relaxation for local *correction*. Thus, a *relaxed local corrector* reads only few bits from a (possibly) corrupt codeword and either recovers the desired bit of the codeword, or rejects in case it detects a corruption.

We give two constructions of relaxed LCCs in two regimes, where the first optimizes the query complexity and the second optimizes the rate:

1. **Constant Query Complexity:** A relaxed LCC with polynomial blocklength whose corrector only reads a constant number of bits of the codeword. This is a sub-exponential improvement over the best *constant query* (full-fledged) LCCs that are known.
2. **Constant Rate:** A relaxed LCC with *constant rate* (i.e., linear blocklength) with quasi-polylogarithmic query complexity (i.e., $(\log n)^{O(\log \log n)}$). This is a nearly sub-exponential improvement over the query complexity of a recent (full-fledged) constant-rate LCC of Kopparty *et al.* (STOC, 2016).

1998 ACM Subject Classification F.1.3 Computation by Abstract Devices, Complexity Measures and Classes

Keywords and phrases Coding Theory, Locally Correctable Codes, Probabilistically Checkable Proofs

Digital Object Identifier 10.4230/LIPIcs.ITCS.2018.27

* A full version of the paper is available at [15], `https://eccc.weizmann.ac.il/report/2017/143/`

† Tom Gur is partially supported by the UC Berkeley Center for Long-Term Cybersecurity, the ISF grant number 671/13, and Irit Dinur's ERC grant number 239985.
‡ Govind Ramnarayan is supported by National Science Foundation grant number 1218547.
§ Ron D. Rothblum is partially supported by NSF MACS - CNS-1413920, SIMONS Investigator award Agreement Dated 6-5-12 and the Cybersecurity and Privacy Institute at Northeastern University.

1 Introduction

Dating back to the seminal works of Shannon [24] and Hamming [17], error correcting codes are used to reliably transmit data over noisy channels and store data. Roughly speaking, error correcting codes are injective functions that take a message and output a codeword, in which the message is encoded with extra redundancy, with the property that even if some of the symbols in the codeword are corrupted, the message is still recoverable.

Locally decodable codes (LDCs) and *locally correctable codes* (LCCs) are error correcting codes that admit highly efficient procedures for recovering small amounts of data. More specifically, in an LDC, a single symbol of the message can be recovered by only reading a few bits from a noisy codeword. An LCC has the same property but with respect to recovering bits of the *codeword* itself (rather than the message).

Locally decodable codes and locally correctable codes have had a profound impact various areas of theoretical computer science including cryptography, PCPs, hardness of approximation, interactive proofs, private information retrieval, program checking, and databases (see [26] and the more recent [20] for a survey on local decodable and correctable codes). While these codes have found numerous uses in theory and practice, one significant downside is that current constructions require adding a large amount of redundancy. Specifically, to decode or correct with a *constant* number of queries, the current state of the art LDCs have super polynomial blocklength [25, 10] (by blocklength we refer to the length of the codeword as a function of the message length) and the current best LCC, which has *sub-exponential* blocklength.[1]

Motivated by this, Ben-Sasson *et al.* [5] defined a natural relaxation of locally decoding, for which they could achieve a dramatically better blocklength. Roughly speaking, their relaxation allows the decoder to abort in case of failure, while still requiring the decoder to successfully decode non-corrupted codewords (in particular, this prevents the decoder from always aborting). Moreover, in the constant query regime, such codes can be transformed to codes, with similar parameters, that are guaranteed to successfully decode on the majority of message bits.

Thus, a *relaxed local decoder* for a code C gets oracle access to a string w that is relatively close to some codeword $c = C(x)$ and an index $i \in [|x|]$. The decoder should make only few queries to w and satisfy the following:

1. If the string $w = c$ (i.e., w is an uncorrupted codeword), the relaxed decoder must always output x_i.
2. Otherwise, with high probability, the decoder should either output x_i or a special "abort" symbol $\perp$ (indicating the decoder detected an error and is unable to decode).[2]

The additional freedom introduced by allowing the decoder to abort turns out to be extremely useful. Using the notion of PCPs of proximity (PCPP), which they also introduce[3] and construct, Ben-Sasson *et al.* obtain relaxed locally decodable codes (RLDCs) with constant query complexity and almost-linear blocklength.

In this work we extend the relaxation of Ben-Sasson *et al.* to LCCs and define the analogous notion of relaxed LCCs as follows: We say that a code $C : \Sigma^k \to \Sigma^n$ is a *relaxed*

[1] These are Reed-Muller codes over a constant-size alphabet and with constant degree (but large dimension).

[2] The actual definition in [5] also requires that for a constant fraction of the coordinates, the decoder decodes correctly (i.e., does not output $\perp$) with high probability. However, they later show that this additional condition follows from Conditions (1) and (2) above. See further discussion in the full version.

[3] The equivalent notion of *assignment tester* was introduced independently by Dinur and Reingold [9].

LCC with query complexity q, if there exists a corrector that has oracle access to a string $w \in \Sigma^n$, which is close to some codeword $c \in C$, and also gets as explicit input an index $i \in [n]$. The algorithm makes at most q queries to the string w, and satisfies the following:

1. If $w = c$ (i.e., w was not corrupted), then the corrector always outputs c_i.
2. Otherwise, with high probability, the corrector either outputs c_i, or a special "abort" symbol $\perp$.

The remarkable savings achieved by Ben-Sasson *et al.* begs the question: can relaxed locally *correctable* codes achieve similar savings in blocklength over current constructions of locally correctable codes? We answer this question in the affirmative by constructing relaxed LCCs with significantly better parameters than that of the state-of-the-art (full-fledged) LCCs.

2 Our Results

In this work, we construct relaxed locally correctable codes in two different parameter regimes: the first, which we view as our main technical contribution, focuses on the constant *query complexity* regime, whereas the second, which is easier to prove given previous work, focuses on constant *rate*.

Constant Query RLCC

Our first result is a relaxed LCC which requires only $O(1)$ queries and has a polynomial blocklength.

▶ **Theorem 1** (Constant Query Relaxed LCC, Informally Stated). *There exists a relaxed* LCC $C : \{0,1\}^k \to \{0,1\}^n$ *with constant relative distance, constant query complexity, and blocklength* $n = \text{poly}(k)$. *Furthermore,* C *is a linear code.*

Theorem 1 yields a sub-exponential improvement compared to the best known (full-fledged) LCCs with constant query complexity, which have sub-exponential blocklength. This result heavily relies on a certain type of PCPs of proximity (PCPPs) that we construct. We elaborate on our PCPP constructions in Section 2.1 below.

We remark that the specific blocklength in Theorem 1 is roughly *quartic* (i.e., fourth power) in the message length. Constructing a constant-query RLCC with a shorter blocklength (let alone an (almost) linear one, as known for relaxed LDCs) is an interesting open problem.

Constant Rate RLCC

Our second main result is a construction of a relaxed LCC with *constant rate*[4] and almost polylogarithmic query complexity.

▶ **Theorem 2** (Constant Rate Relaxed LCC, Informally Stated). *There exists a relaxed* LCC $C : \{0,1\}^k \to \{0,1\}^n$ *with constant relative distance, query complexity* $(\log n)^{O(\log \log n)}$, *and constant rate (i.e., blocklength* $n = O(k)$*). Furthermore,* C *is a linear code and has distance-rate tradeoff approaching the Zyablov bound [27].*

[4] Recall that the rate of a code $C : \Sigma^k \to \Sigma^n$ is defined as k/n. We use the terms "constant rate" and "linear blocklength" interchangeably.

This is a nearly sub-exponential improvement in query complexity over the best constant rate (full-fledged) LCCs, due to Kopparty *et al.* [19], which requires $2^{\tilde{O}(\sqrt{\log n})}$ queries for correction. As a matter of fact, our construction is essentially identical to one of the constructions of [19].[5] Our main insight in proving Theorem 2 is that their code allows for *relaxed* local correction with much better parameters.[6] As a secondary contribution, we also provide a modular presentation for the *distance amplification* step, which is the main step in [19] (and is originally due to Alon, Edmonds and Luby [1]).

▶ Remark. As mentioned in Footnote 2, the original definition of RLDC [5] includes a third condition, which requires that the decoder must successfully decode a constant fraction of the coordinates. More precisely, for every $w \in \Sigma^n$ that is close to some codeword $c = C(x)$, there exists a set $I_w \subseteq [k]$ of size $\Omega(k)$ such that for every $i \in I_w$ with high probability the decoder D outputs x_i (rather than outputting $\perp$).

Ben-Sasson *et al.* showed that every RLDC with constant query complexity that satisfies the first two conditions, can be transformed into an RLDC with similar parameters that satisfies the third condition as well. We remark that this transformation also applies to RLCCs with constant query complexity. However, for super-constant query complexity (as in Theorem 2) the same transformation only guarantees successful decoding of a constant fraction of coordinates, if the codeword is corrupted on a sub-constant fraction of its coordinates (i.e., the fraction roughly corresponds to the reciprocal of the query complexity).

▶ Remark. Both our constant-query and constant-rate RLCCs are systematic[7]. Hence they are automatically also relaxed locally *decodable* codes (i.e., RLDCs). In particular, the code from Theorem 2 is also the first construction of a relaxed locally *decodable* code in the constant-rate regime, with query complexity $(\log n)^{O(\log \log n)}$.

2.1 PCP Constructions

PCPs of proximity (PCPP), first studied by Ben-Sasson *et al.* [5] and by Dinur and Reingold [9] were originally introduced to facilitate PCP composition. Beyond their usefulness in PCP constructions, of PCPPs have proved to be extremely useful in coding theory as well. Indeed, PCPPs lie at the heart of several constructions of LTCs [14], relaxed LDCs [5, 13], universal LTCs [11, 12], as well as in our construction of relaxed LCCs (specifically in Theorem 1).

Loosely speaking, a PCPP is a proof system that allows for probabilistic verification of approximate decision problems by querying only a small number of locations in both the statement and the proof. (In contrast, a standard PCP verifier reads the *entire* statement, and probabilistically verifies an *exact* decision problem, by querying only a small number of locations in the proof.) Similarly to the scenario in property testing, the soundness guarantee provided by PCPPs is that the PCPP verifier is only required to reject statements that are "far" (in Hamming distance) from being correct.

In this work, we provide new constructions of PCPPs that play a crucial role in our constant-query relaxed LCC construction. The PCPPs that we construct are for verifying membership in affine subspaces (rather than general languages in P or NP), since this is all that we need for our RLCC constructions. More specifically, we shall construct PCPPs that

[5] Interestingly, our construction is inspired by the [19] construction of a locally *testable* code, rather than their locally *correctable* code.

[6] We were informed that a similar observation about the [19] code has been made recently and independently in an unpublished work of Hemenway, Ron-Zewi, and Wootters [18].

[7] Recall that a code is systematic if the first part of every codeword is the original message.

are: *linear*, *robust*, *self-correctable*, and admit *strong canonical soundness*. We discuss these properties in more detail next (see full version for precise definitions). We remark that our PCPP construction is inspired by the construction of linear-inner proof-systems (LIPS) by Goldreich and Sudan [14].

Linearity

We call a PCPP proof-system *linear* if it satisfies two conditions. First, the prescribed proof π for any statement x must be a linear function of the statement. Second, to decide whether to accept, the PCPP verifier only checks that the values that it reads from the input and PCPP proof lie in an affine subspace. Put differently, the verifier's decision predicate is computable by a linear circuit. We remark that in the literature [6, 22], the term "linear PCPP" sometimes refers only to the latter of the two requirements but here we also insist that the proof be a linear function of the statement.

We use linearity both to assure that our resulting codes are linear codes, as well as to facilitate composition with other PCPPs. We note that standard PCPs are typically inherently non-linear (since they are designed for general languages in P or in NP). However, in our context we are only trying to verify membership in affine spaces and so it is reasonable to expect to have linear PCPPs.

Robustness

The notion of *robust* PCPPs, introduced by Ben Sasson *et al.* [5], refers to PCPP systems whose verifier, roughly speaking, is not only required to reject statements that are far from valid but also that the local view of the verifier (i.e., the answers to the queries made by verifier) is far from any local view that would have caused the verifier to accept. Robustness plays a key role in enabling PCP composition. While this condition holds trivially for verifiers with constant query complexity, in our construction we will also consider verifiers with super-constant query complexity, for which achieving robustness is non-trivial.

Self-Correctability

In a *self-correctable* PCPP system, the proof oracle admits a local correction procedure that allows for local recovery of individual bits of a moderately corrupted PCPP proof. The self-correctability of the PCPP oracles allows us to include them as part of an RLCC's codeword.

Strong Canonical Soundness

The notion of PCPPs with *strong canonical soundness*, introduced by Goldreich and Sudan [14], requires that correct inputs (i.e., that reside in the target language) have a canonical proof and the PCPP verifier is required to reject "wrong" (i.e., non-canonical) proofs, *even for correct statements*. In more detail, these PCPPs satisfy two additional requirements: (1) *canonicity*: for every true statement there exists a unique canonical proof that the verifier is required to always accept, and (2) *strong canonical soundness*: the verifier is required to reject any pair (x, π) of statement and proof with probability that is roughly proportional to its distance from a true statement and its corresponding canonical proof.

We are now ready to state our results on PCPs of proximity with the aforementioned properties. The first construction has exponential length and constant query complexity,

whereas the second construction, whose proof is significantly more involved, has polynomial length and poly-logarithmic query complexity.

Our first result is a variant of the Hadamard PCPP, with exponential length but constant query complexity.

▶ **Theorem 3** (Informally stated, see full version). *There exists a linear, self-correctable, strong canonical* PCPP *for membership in affine subspaces, with query complexity* $O(1)$ *and exponential length (in the size of the statement).*

Our second result is a variant of the [4] PCP, which has poly-logarithmic query complexity and polynomial length.

▶ **Theorem 4** (Informally stated, see full version). *There exists a linear, self-correctable,* $\Omega(1)$*-robust, strong canonical* PCPP *for membership in affine subspaces, with query complexity* $\text{polylog}(n)$ *and* $\text{poly}(n)$ *length, for statements of length* n*.*

3 Technical Overview

The techniques used for our two constructions are quite different. We first outline the constant-query result, which is more complex, in Section 3.1 and then outline the constant-rate result in Section 3.2.

3.1 Constant-Query Relaxed LCC

The starting point for our construction is the [5] construction of relaxed locally *decodable* codes (RLDC), which we review next.[8] In their construction, each codeword has two parts: the first part provides the distance, and the second enables relaxed local decodability. More specifically, they construct an RLDC C' whose codewords consist of the following two equal-length parts: (1) repetitions of a codeword $C(x)$, where $C : \{0,1\}^k \to \{0,1\}^n$ is some systematic code with constant distance and rate, (2) for every message bit in $C(x)$, they add a PCPP, which is a proof that x_i is indeed the i^{th} bit of $C(x)$.[9]

We remark that the repetitions in the first part of the code are simply meant to ensure distance (as the PCPP proof strings are not necessarily a code with good distance). To decode, the relaxed decoder for C' invokes the PCPP verifier to check that the i'th bit of the first part is indeed $C(x)_i$ and outputs it, unless the verifier rejects, in which case the relaxed decoder may return $\perp$.

Ben-Sasson *et al.* show that this code is indeed a relaxed LDC. However, in general, it will not necessarily be a relaxed LCC. Specifically, it is unclear how to correct bits that are part of the PCPP proof strings. Simply appending even more PCPPs to deal with the original ones will not do since we will also need to correct those. Moreover, it is worth pointing out that even if the PCPP proof strings had some internal self-correction mechanism, this would still not suffice since each PCPP proof string by itself is very short (as compared to the entire codeword) and could therefore be entirely corrupted.

Before proceeding to cope with this difficulty, we first suggest a different perspective on the [5] construction, which is inspired by the highly influential and useful notion of PCP

[8] We describe the simpler variant of the [5] construction, which achieves nearly *quadratic* blocklength. We remark that [5] also present a more involved construction that achieves nearly *linear* blocklength.

[9] Actually, our presentation differs slightly from that of [5]. Their construction contains an additional part that consists of repetitions of the original *message*. However, when using a systematic code C, this addition is not necessary.

composition [3]. Specifically, we think of the [5] construction as a *composition* of the code C, which is trivially locally decodable with n queries, with a constant-query PCPP. This composition yields a *relaxed* LDC with query complexity $O(1)$, at a moderate increase in blocklength (which comes from appending all of the PCPP proof strings).

We shall adopt the composition perspective, and use it to construct a relaxed locally *correctable* code, by introducing a technique for composing a (possibly relaxed) LCC C with a special type of PCP of proximity (PCPP). The result of the composition is a *relaxed* LCC C' which basically inherits the query complexity of the PCPP (and with a moderate overhead in blocklength).

Similarly to the relaxed LDCs of [5], the codewords of C' are constructed by taking repetitions of a codeword of a (possibly relaxed) robust RLCC C and concatenating it with many PCPP proof strings. Specifically, for each set of queries that the relaxed local corrector for C would like to make, we write down a PCPP proof that this set of queries would be answered correctly. We shall refer to the first part of C', which contains repetitions of C, as the *core* of C', and refer to the second as the PCPP part.

Observe that the foregoing approach allows us to locally correct bits of the *core* of C'. The relaxed corrector for the composed RLCC takes the queries made by the old corrector as input, and uses the PCPP verifier to test if the old corrector would have accepted.[10] However, we shall need a more sophisticated machinery to correct the PCPP part of C' (indeed this is exactly the challenge that we faced when trying to follow the [5] approach). This will be achieved by ensuring that the PCPPs that we use have strong properties.

In particular, we shall employ the foregoing composition strategy while using the PCPPs of Section 2.1, which admit canonical proofs, strong canonical soundness, and self-correctability. Recall that a PCPP is said to have strong canonical soundness if every valid input has a *canonical* proof that it accepts, and any pair of statement and proof are rejected with probability proportional to their distance from a true statement and its corresponding canonical proof. In addition, recall that a canonical PCPP is said to be *self-correctable* if the canonical proof strings form a locally correctable code (i.e., if it is possible to locally recover individual bits of a noisy PCPP oracle).

Suppose that we want to correct a bit that lies in the PCPP part of a purported codeword of C'. If this bit is in a PCPP oracle that is not too corrupted, we can simply use the PCPP's self-corrector to recover the bit. However, as pointed out before, this naive attempt to self-correct fails when the *entire* proof string is corrupted. This can easily happen when the proof strings, each of which refers to a single possible query set of the original corrector, are short relative to the size of the entire codeword.

Thus, our main challenge is to detect whether the given PCPP proof string was (possibly entirely) corrupted. We observe that if on the one hand, the PCPP oracle we wish to correct is heavily corrupted, while on the other hand, the statement to which the PCPP refers (i.e., the queries that the corrector for C makes) is *not* heavily corrupted, then the proof is far from the prescribed canonical proof. The strong canonical soundness guarantees that in such case the PCPP verifier will detect the corruption and reject. Thus, we are left with the case that *both* the PCPP oracle and the statement that it refers to are heavily corrupted.

To detect this deviation, we choose a random point in the foregoing statement and read it directly. Since that point is in the core of the code, and we have already described the

[10] Even for this to work, we need to ensure that the original RLCC is *robust*, in the sense that with high probability the corrector's view (i.e., the answers to its queries) are *far* from answers that would make it output an incorrect value. Otherwise, we have no guarantee that the PCPP verifier will see the error.

procedure for correcting in the core, we can also correct this point and compare the corrected value with the symbol that we read directly. Since we have assumed that the statement was heavily corrupted, the value that we read directly will likely be different than what the corrector returns, in which case we can reject.

Equipped with this composition theorem, we can now construct our code. By applying the composition theorem to the low-degree extension code, of suitable parameters, and the PCPP given in Theorem 3, we can already construct a constant-query RLCC with quasi-polynomial blocklength. We note that this is already a significant improvement over the current best (full-fledged) LCCs. However, to obtain polynomial blocklength, we shall perform two compositions with different PCPPs (in direct analogy to the first proof of the PCP theorem [2]).

As in the quasi-polynomial result mentioned above, our starting point is the low-degree extension code. Under a suitable parameterization, this code is known to be a robust (full-fledged) LCC with almost linear blocklength and polylogarithmic query complexity. We shall first compose it with the polynomial length, polylogarithmic query, strong canonical, self-correctable and robust PCPP of Theorem 4. Since the foregoing PCPP is robust, this composition yields a robust RLCC with polynomial blocklength and slightly sub-logarithmic query complexity. Finally, we compose yet again with the exponential length, constant query, strong canonical, self-correctable PCPP from Theorem 3, which yields an RLCC with constant query complexity.

Each of our two composition steps introduces at least a quadratic overhead to the blocklength. This is because our composition of an RLCC with a PCPP appends a PCPP proof-string for every pair (i, ρ) of coordinate i to be corrected from the base code and random string ρ of the underlying corrector with respect to the point i.[11] Since we apply two such composition steps, we get a code with roughly $n \approx k^4$ blocklength.

3.2 Constant-Rate RLCC

For the constant rate construction, we build on the recent breakthrough construction of locally testable[12] and correctable codes of Kopparty *et al.* [19]. Interestingly, we will actually focus on the [19] construction of locally *testable* codes (rather than correctable ones), even though our own goal is to construct (relaxed) locally *correctable* codes.[13]

Kopparty *et al.* construct locally testable codes with query complexity $(\log n)^{O(\log\log(n))}$ by taking an iterative approach, similar to that of Meir [21]. They start off with a code of dimension $\operatorname{poly}\log(n)$ (which is trivially locally testable, by reading the entire codeword) and gradually increase its blocklength, while (almost) preserving the local testability and maintaining the rate of the code close to 1. This amplification step is achieved by combining two transformations on codes:

1. *Code tensoring*: this transformation squares the block-length (which is good since we want to obtain blocklength n) and rate (which is not too bad since our rate is close to 1). The main negative affect is that this transformation also squares the distance.

[11] In contrast, in standard PCP composition, one only appends an inner PCP proof-string for every random string ρ of the outer PCP. Thus, as long as the randomness complexity of the outer PCP is minimal, it is possible to achieve close to constant multiplicative overhead when composing.

[12] Recall that a locally testable code [14] is a code for which one can test, using a sub-linear number of queries, whether a given string is a codeword or far from such.

[13] This may not be surprising, since the notions of relaxed correctability and testability are closely related. In particular, as observed in [11], every RLDC (analogously, RLCC) is roughly equivalent to a code C such that for every coordinate i and value b, the subcode obtained by fixing the i'th bit to b (i.e., $\{C(x) : C(x)_i = b\}$) is locally testable.

2. *Distance amplification*: remarkably, this transformation fixed the loss in distance caused by the tensoring step, without harming the rate or local testability too much.

As noted above, in their work, Kopparty *et al.* also construct a locally correctable code, albeit only with query complexity $2^{\tilde{O}(\sqrt{\log(n)})}$. The reason why their LCC construction does not match the parameters of their LTC construction is that the tensoring step, used in their construction of locally *testable* codes, is not known to preserve local correctability.[14]

Our key observation is that tensoring does preserve *relaxed* local correctability. Recall that the tensor of a code $C : \mathbb{F}^k \to \mathbb{F}^n$ is the code $C^2 : \mathbb{F}^{k^2} \to \mathbb{F}^{n^2}$ that consists of all strings $c \in \mathbb{F}^{n^2}$, viewed as $n \times n$ matrices, that consist of rows and columns that are each codewords of c.

Suppose that C is a (relaxed) LCC with query complexity q. We want to show that C^2 is also a (relaxed) LCC with query complexity roughly q. Let $w \in \mathbb{F}^{n^2}$ be a (possibly) corrupt codeword of C^2. Thus, w which we also view as an $n \times n$ matrix, is close to some codeword $c \in \mathbb{F}^{n^2}$. Given an index $(i, j) \in [n] \times [n]$ to correct, a natural approach is apply the (relaxed) local corrector of C on the i^{th} row of w, with respect to the index j.

If it were the case that the i^{th} row of w were close to the i^{th} row of c, we would be done. However, the i^{th} row of w only constitutes a $1/n$ fraction of w and so it could potentially be entirely corrupt. Let us assume that it is indeed the case that i^{th} rows of c and w (almost) entirely disagree.

To detect that this is the case, our corrector chooses at random a few columns $J \subset [n]$. On the one hand, since the i^{th} rows of w and c disagree almost everywhere, with high probability for some $j' \in J$ it will be the case that $w_{i,j'} \neq c_{i,j'}$. On the other hand, since j' is just a random column, with high probability the j'^{th} columns of w and c are close.

Given this, a natural approach is to have our corrector read the (i, j')-th entries of w for every $j' \in J$, by applying the (relaxed) local corrector of C. In the likely case that it chooses a j' such that the j'^{th} column of w and c are close, with high probability the corrector will either return $c_{i,j'}$ or $\perp$. If our corrector sees $\perp$ it can immediately reject (since this would never happen for an exact codeword). Otherwise, if our corrector sees the value $c_{i,j'}$, it can compare this value with $w_{i,j'}$ (by explicitly reading the (i, j')'th entry of w). By the above analysis, these values will be different (with high probability), in which case our corrector can also reject.

To calculate the overall query complexity of the resulting code, we need to account for the overhead introduced by both the tensoring and distance amplification steps. Assuming that C is (relaxed) locally correctable up to distance δ_R, the tensoring step only increases the query complexity by $O(1/\delta_R)$. Each distance amplification increases the query complexity by roughly a polylog(n) factor. Thus, since we need roughly $\log \log(n)$ iterations to reach blocklength n, the overall query complexity is $(\log n)^{O(\log \log n)}$.

4 Related Works

A similar notion to RLDCs called *Locally Decode/Reject Codes (*LDRC*s)* arose in the beautiful work of Moshkovitz and Raz [23] on constructing two-query PCPs with sub-constant error. These are similar to RLDCs in that they are codes with a decoder that is permitted to reject if it sees errors. However, it is important to note that the two notions differ in a few significant ways and are overall incomparable. First, LDRCs decode a k-tuple of coordinates jointly,

[14] It can be shown that tensoring at most *squares* the query complexity for local correcting. However, the [19] iterative approach cannot afford such an overhead in each iteration.

rather than a single coordinate. Second, LDRCs have a "list-decoding" guarantee – namely, that the decoding agrees with one message in a small list of messages – as opposed to RLDCs, which provide unique decoding (but up to a smaller radius). Finally, LDRCs only need to work with high probability over the choice of k-tuple, while RLDCs must decode or reject with high probability for *every* coordinate. See [23, Section 2] for the formal definition of LDRCs and a comparison to RLDCs.

Another related notion is that of *decodable* PCP*s* (dPCP), first introduced by Dinur and Harsha [8] to the end of obtaining a modular and simpler proof of the the [23] result. A dPCP is a PCP oracle, encoding an NP-witness, which allows for list decoding of individual bits of the NP witness it encodes. Dinur and Harsha provided constructions of such dPCPs as well as a composition theorem for dPCPs.

Additionally, in a recent work, Goldreich and Gur [11] introduced the notion of *universal locally testable codes* (universal-LTC), which can be thought of as generalizing the notion of relaxed LDCs. A universal-LTC $C : \{0,1\}^k \to \{0,1\}^n$ for a family of functions $\mathcal{F} = \left\{f_i : \{0,1\}^k \to \{0,1\}\right\}_{i\in[M]}$ is a code such that for every $i \in [M]$ and $b \in \{0,1\}$, membership in the subcode $\{C(x) : f_i(x) = b\}$ can be locally tested. As was shown in [11], universal-LTCs with respect to the family of dictators functions (i.e., of the form $f(x) = x_i$) are roughly equivalent to RLDCs. We remark that their formulation can be naturally generalized to also capture the notion of RLCC.

Finally, we remark that the relaxed LDCs have been used in the context of interactive proofs of proximity [16] and property testing [7].

Acknowledgements. We thank Oded Goldreich for initiating a discussion that led to this work, for insightful conversions and for his encouragement. The second author would like to also thank Dana Moshkovitz for useful discussions surrounding this work.

References

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

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

3 Sanjeev Arora and Shmuel Safra. Probabilistic checking of proofs: A new characterization of NP. *J. ACM*, 45(1):70–122, 1998. doi:10.1145/273865.273901.

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 Eli Ben-Sasson, Oded Goldreich, Prahladh Harsha, Madhu Sudan, and Salil P. Vadhan. Robust pcps of proximity, shorter pcps, and applications to coding. *SIAM J. Comput.*, 36(4):889–974, 2006. doi:10.1137/S0097539705446810.

6 Eli Ben-Sasson, Prahladh Harsha, Oded Lachish, and Arie Matsliah. Sound 3-query pcpps are long. *TOCT*, 1(2):7:1–7:49, 2009. doi:10.1145/1595391.1595394.

7 Clément L. Canonne and Tom Gur. An adaptivity hierarchy theorem for property testing. *Computational Complexity Conference (CCC)*, 2017.

8 Irit Dinur and Prahladh Harsha. Composition of low-error 2-query pcps using decodable pcps. In *50th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2009,*

October 25-27, 2009, Atlanta, Georgia, USA, pages 472–481. IEEE Computer Society, 2009. doi:10.1109/FOCS.2009.8.

9 Irit Dinur and Omer Reingold. Assignment testers: Towards a combinatorial proof of the PCP theorem. *SIAM J. Comput.*, 36(4):975–1024, 2006. doi:10.1137/S0097539705446962.

10 Klim Efremenko. 3-query locally decodable codes of subexponential length. *SIAM J. Comput.*, 41(6):1694–1703, 2012. doi:10.1137/090772721.

11 Oded Goldreich and Tom Gur. Universal locally testable codes. *Electronic Colloquium on Computational Complexity (ECCC)*, 23:42, 2016. URL: http://eccc.hpi-web.de/report/2016/042.

12 Oded Goldreich and Tom Gur. Universal locally verifiable codes and 3-round interactive proofs of proximity for CSP. *Electronic Colloquium on Computational Complexity (ECCC)*, 23:192, 2016. URL: http://eccc.hpi-web.de/report/2016/192.

13 Oded Goldreich, Tom Gur, and Ilan Komargodski. Strong locally testable codes with relaxed local decoders. In David Zuckerman, editor, *30th Conference on Computational Complexity, CCC 2015, June 17-19, 2015, Portland, Oregon, USA*, volume 33 of *LIPIcs*, pages 1–41. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2015. doi:10.4230/LIPIcs.CCC.2015.1.

14 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.

15 Tom Gur, Govind Ramnarayan, and Ron Rothblum. Relaxed locally correctable codes. *Electronic Colloquium on Computational Complexity (ECCC)*, 24:143, 2017. URL: https://eccc.weizmann.ac.il/report/2017/143.

16 Tom Gur and Ron D. Rothblum. Non-interactive proofs of proximity. *Computational Complexity*, pages 1–109, 2016. doi:10.1007/s00037-016-0136-9.

17 Richard W Hamming. Error detecting and error correcting codes. *Bell Labs Technical Journal*, 29(2):147–160, 1950.

18 Brett Hemenway, Noga Ron-Zewi, and Mary Wootters, 2017. Personal communication.

19 Swastik Kopparty, Or Meir, Noga Ron-Zewi, and Shubhangi Saraf. High-rate locally-correctable and locally-testable codes with sub-polynomial query complexity. 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 202–215. ACM, 2016. doi:10.1145/2897518.2897523.

20 Swastik Kopparty and Shubhangi Saraf. Local testing and decoding of high-rate error-correcting codes. *Electronic Colloquium on Computational Complexity (ECCC)*, 24:126, 2017. URL: https://eccc.weizmann.ac.il/report/2017/126.

21 Or Meir. Combinatorial construction of locally testable codes. *SIAM J. Comput.*, 39(2):491–544, 2009. doi:10.1137/080729967.

22 Or Meir. Combinatorial pcps with short proofs. *Computational Complexity*, 25(1):1–102, 2016. doi:10.1007/s00037-015-0111-x.

23 Dana Moshkovitz and Ran Raz. Two-query pcp with subconstant error. *Journal of the ACM (JACM)*, 57(5):29, 2010.

24 Claude Elwood Shannon. Communication in the presence of noise. *Proceedings of the IRE*, 37(1):10–21, 1949.

25 Sergey Yekhanin. Towards 3-query locally decodable codes of subexponential length. *J. ACM*, 55(1):1:1–1:16, 2008. doi:10.1145/1326554.1326555.

26 Sergey Yekhanin. Locally decodable codes. *Foundations and Trends in Theoretical Computer Science*, 6(3):139–255, 2012. doi:10.1561/0400000030.

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

Entropy Samplers and Strong Generic Lower Bounds For Space Bounded Learning*†

Dana Moshkovitz‡[1] **and Michal Moshkovitz**§[2]

1 **Department of Computer Science, UT Austin, USA**
danama@cs.utexas.edu

2 **The Edmond and Lily Safra Center for Brain Sciences, Hebrew University, Israel**
michal.moshkovitz@mail.huji.ac.il

Abstract

With any hypothesis class one can associate a bipartite graph whose vertices are the hypotheses $\mathcal{H}$ on one side and all possible labeled examples $\mathcal{X}$ on the other side, and an hypothesis is connected to all the labeled examples that are consistent with it. We call this graph the *hypotheses graph*. We prove that any hypothesis class whose hypotheses graph is mixing cannot be learned using less than $\Omega(\log^2|\mathcal{H}|)$ memory bits unless the learner uses at least a large number $|\mathcal{H}|^{\Omega(1)}$ labeled examples. Our work builds on a combinatorial framework that we suggested in a previous work for proving lower bounds on space bounded learning. The strong lower bound is obtained by defining a new notion of pseudorandomness, the entropy sampler. Raz obtained a similar result using different ideas.

1998 ACM Subject Classification I.2.6 Learning, F.1.3 Complexity Measures and Classes

Keywords and phrases learning, space bound, mixing, certainty, entropy sampler

Digital Object Identifier 10.4230/LIPIcs.ITCS.2018.28

1 Introduction

Let $\mathcal{H}$ be a family of Boolean hypotheses. One can learn an hypothesis from $\mathcal{H}$ after seeing $O(\log|\mathcal{H}|)$ random labeled examples. Intuitively, this is true since a typical labeled example cuts the number of possible hypotheses by a factor of two. However, learning with so few examples requires enough memory to store $\Theta(\log|\mathcal{H}|)$ examples in memory. If $\mathcal{X}$ is the family of possible labeled examples, then such a learner uses $\Theta(\log|\mathcal{X}|\cdot\log|\mathcal{H}|)$ memory bits. It is also possible to learn $\mathcal{H}$ using many fewer memory bits: enumerate the hypotheses one by one, moving to the next hypothesis only after encountering a new labeled example that is inconsistent with the current hypothesis. Such a brute force learner uses only $\log|\mathcal{H}|$ memory bits but requires an extravagant number $\Theta(|\mathcal{H}|\log|\mathcal{H}|)$ of labeled examples. A natural question is whether one can learn with *both* $\ll\Theta(\log|\mathcal{X}|\cdot\log|\mathcal{H}|)$ memory bits and $\ll|\mathcal{H}|$ labeled examples.

* A preliminary version of this work appeared as ECCC TR17-116.
† A full version of the paper is available at https://eccc.weizmann.ac.il/report/2017/116/.
‡ This material is based upon work supported by the National Science Foundation under grants number 1218547 and 1648712.
§ This work is partially supported by the Gatsby Charitable Foundation, The Israel Science Foundation, and Intel ICRI-CI center. M.M. is grateful to the Harry and Sylvia Hoffman Leadership and Responsibility Program.

Perhaps surprisingly, Raz [8] showed that parities ($\mathcal{X} = \{0,1\}^n \times \{0,1\}$ and $\mathcal{H} = \{\oplus_{i\in I} x_i | I \subseteq \{1,\ldots,n\}\}$) cannot be learned unless the learner uses either $\Omega(\log|\mathcal{X}|\cdot\log|\mathcal{H}|) = \Omega(n^2)$ memory bits or $|\mathcal{H}|^{\Omega(1)} = 2^{\Omega(n)}$ labeled examples. Until recently, parities gave the only hypothesis classes known with strong lower bounds on space-bounded learning[1].

In this work we show that strong lower bounds hold for any hypothesis class that satisfies a natural combinatorial condition about the mixing of a graph associated with the class. This subsumes the result on parities and shows similar results for random classes and classes that correspond to error correcting codes [6]. Many other applications follow using the large body of research on combinatorial mixing (see, e.g., [2]). More details will appear in the full version of this paper.

An hypothesis class can be described by a bipartite graph whose vertices are the hypotheses $\mathcal{H}$ and the labeled examples $\mathcal{X}$, and whose edges connect every hypothesis $h \in \mathcal{H}$ to the labeled examples $(x,y) \in \mathcal{X}$ that are consistent with it, i.e., $h(x) = y$. We say that the hypothesis class is d-mixing if for any set of hypotheses $A \subseteq \mathcal{H}$ and any set $B \subseteq \mathcal{X}$ of labeled examples it holds that $||E(A,B)| - |A||B|/2| \leq \mathrm{d}\sqrt{|A||B|}$, where $E(A,B)$ is the set of edges between A and B in the hypotheses graph. For instance, for parities, $d = \Theta(\sqrt{|\mathcal{X}|})$ (see, e.g., [6]). We prove that mixing hypothesis classes admit strong lower bounds on space-bounded learning.

▶ **Theorem 1** (main theorem). *If the hypotheses graph is* d*-mixing,* $r := \frac{|\mathcal{H}||\mathcal{X}|}{\mathrm{d}^2}$ *and* $|\mathcal{H}|$ *are at least some constants, then any learning algorithm that outputs the underlying hypothesis with probability at least* $r^{-\Theta(1)}$ *must use at least* $\Omega(\log^2 r)$ *memory bits or* $r^{\Omega(1)}$ *labeled examples.*

A similar theorem holds if the learner only *approximately* learns the underlying hypothesis [6].

1.1 Related Work

In this work we rely on a combinatorial framework – henceforth referred to as the *low certainty framework* – that we introduced in a previous work for analyzing space-bounded learning [6]. In [6] the bound on the number of memory states was only $\approx |\mathcal{H}|^{1.25}$ (the bound on the number of labeled examples was the optimal $|\mathcal{H}|^{\Omega(1)}$). In between those two works (the current work and [6]) Raz [9] showed a lower bound of $\Omega(\log^2|\mathcal{H}|)$ on the number of memory bits (as in the current paper), relying on a spectral mixing condition instead of a combinatorial mixing condition. In a subsequent work, Garg, Raz and Tal [3], and, independently, Beame, Gharan and Yang [1], improved the lower bound to the optimal $|\mathcal{X}|^{\Omega(\log|\mathcal{H}|)}$.

1.2 Entropy Sampler

The key idea in the current work is a new notion of pseudorandomness, which we call the entropy sampler. Fix a probability distribution p over a space $\mathcal{M}$. For every element $m \in \mathcal{M}$ let its "entropy level" be $k_m = \log(1/p(m))$. The min-entropy of p is $\min_m k_m$. A sampler with multiplicative error is defined as follows:

---

[1] Kol, Raz and Tal [4] generalized Raz's work to parities on l variables out of n, showing that either $\Omega(n)$ memory bits or $2^{\Omega(l)}$ examples are needed, and for $l \leq n^{0.9}$, either $\Omega(nl^{0.99})$ memory bits or $l^{\Omega(l)}$ examples are needed. Note: (1) For small l there are learners with both $\ll |\mathcal{X}|^{\Omega(\log|\mathcal{H}|)} = n^{\Omega(nl)}$ memory states and $\ll |\mathcal{H}|^{\Omega(1)} = n^{\Omega(l)}$ examples [4]. (2) The work [4] implies lower bounds for classes that contain parities on l out of n variables. To get a result for interesting classes, like DNFs or decision trees, one can pick $l \approx \log n$, but then the lower bounds are weak.

▶ **Definition 2** (Sampler)**.** A bipartite graph $(\mathcal{M}, \mathcal{H}, E)$ is a sampler with multiplicative factor L, min-entropy k and error ε, if for every distribution p over $\mathcal{M}$ of min-entropy at least k, for every $H \subseteq \mathcal{H}$, $|H| \geq \varepsilon|\mathcal{H}|$,

$$\sum_{m \in \mathcal{M}} p(m) \cdot \frac{|E(m,H)|}{|E(m,\mathcal{H})|} \leq L \cdot \frac{|H|}{|\mathcal{H}|},$$

where $E(\cdot,\cdot)$ denotes the set of edges between given memories and hypotheses in the knowledge graph.

The parameters of the sampler L, ε, are typically related to the min-entropy k. The higher the min-entropy k is, the lower the sampling parameters are. However it's possible, e.g., that all elements are at high entropy levels, except for one, for the min-entropy to be low and for the sampling parameters to be high. An entropy samplerbenefits from elements of all entropy levels starting k; higher entropy levels contribute to better sampling. Formally:

▶ **Definition 3** (entropy sampler)**.** A bipartite graph $(\mathcal{M}, \mathcal{H}, E)$ is an entropy samplerwith multiplicative factor L, min-entropy k, error ε and benefit α if for every distribution p of min-entropy k, for every $H \subseteq \mathcal{H}$, $|H| \geq \varepsilon|\mathcal{H}|$,

$$\sum_{m \in \mathcal{M}} p(m) \cdot \frac{|E(m,H)|}{|E(m,\mathcal{H})|} \cdot 2^{\alpha \cdot k_m} \leq L \cdot \frac{|H|}{|\mathcal{H}|}.$$

Typically, pseudorandom objects can only be defined with respect to min-entropy, and therefore the notion of an entropy sampleris unusual and may have other applications.

1.3 Proof Outline

The proof of Theorem 1 builds on the low certainty framework of [6]. A key object in the framework is the *knowledge graph* of the algorithm at various time steps. The knowledge graph at a certain time step is a bipartite graph, where one side corresponds to memory states of the learning algorithm and the other side corresponds to the possible hypotheses in $\mathcal{H}$. There is an edge (m, h) between a memory state m and an hypothesis h for every sequence of labeled examples that is consistent with h and leads to m at the relevant time step. Note that when an hypothesis is picked uniformly at random, the neighborhood of a memory state corresponds to the probability distribution over the possible hypotheses conditioned on landing in the memory state at the relevant time step. In this respect, the knowledge graph captures exactly the knowledge of the algorithm about the underlying hypothesis at the time step.

In order to prove lower bounds, the work [6] shows that when the hypotheses graph is mixing and the space is sufficiently bounded, the knowledge graph remains "pseudorandom" throughout the execution of the algorithm. Unfortunately, the pseudorandomness property of [6] (analogous to an extractor property) no longer holds when we wish to rule out learners that can store whole labeled examples in memory. The main idea of the current work is to prove that the knowledge graph is instead an entropy sampler(or, rather, a version of Definition 3 that is suitable for the knowledge graph). We show this by induction on the time t in the execution of the algorithm. Every probability distribution over memories at time $t+1$ corresponds to a probability distribution over memories at time t. This distribution depends on the likelihood of transitions to the time $t+1$ memories. Roughly speaking, less likely transitions from time t to time $t+1$ may give a lot of information about the underlying hypothesis. The notion of an entropy samplerguarantees that even after taking the new information into account sampling still holds (the actual analysis is quite involved, partly because it takes irregularity into account).

2 Preliminaries

$\log(\cdot)$ always means $\log_2(\cdot)$. The following claims were proven in [6]:

▶ **Claim 4.** *Let p be a probability distribution over a set A with $\sum_{i\in A} p(i)^2 \leq r$. Then, for every $A' \subseteq A$ it holds that $\sum_{i\in A'} p(i) \leq \sqrt{|A'|r}$.*

▶ **Claim 5** (generalized law of total probability)**.** *For any events A, B and a partition of the sample space $C_1, \ldots, C_n$,*

$$\Pr(A|B) = \sum_i \Pr(A|B, C_i)\Pr(C_i|B).$$

▶ **Claim 6** (generalized Bayes' theorem)**.** *For any three events A, B, C,*

$$\Pr(A|B, C) = \Pr(B|A, C)\frac{\Pr(A|C)}{\Pr(B|C)}$$

▶ **Claim 7.** *Suppose $B_1, \ldots, B_n$ are some disjoint events. Then,*

$$\Pr(A|B_1 \cup \ldots \cup B_n) = \sum_{i=1}^{n} \Pr(A|B_i)\frac{\Pr(B_i)}{\Pr(B_1 \cup \ldots \cup B_n)}.$$

2.1 Mixing

For a bipartite graph (A, B, E), A are the left vertices and B are the right vertices. For sets $S \subseteq A, T \subseteq B$ let

$$E(S, T) = \{(a, b) \in E | a \in S, b \in T\}.$$

For $a \in A$ (and similarly for $b \in B$) the neighborhood of a is $\Gamma(a) = \{b \in B | (a, b) \in E\}$, and the degree of a is $d_a = |\Gamma(a)|$. If all d_a are equal, we say that the graph is d_a-left regular or just left regular. We similarly define right regularity.

▶ **Definition 8** (mixing)**.** We say that a bipartite graph (A, B, E) with average left degree $\bar{d}_A$ is d-mixing if for any $S \subseteq A, T \subseteq B$ it holds that

$$\left||E(S, T)| - \frac{|S||T|}{|B|/\bar{d}_A}\right| \leq \mathrm{d}\sqrt{|S||T|}$$

▶ **Definition 9** (sampler)**.** A bipartite graph (A, B, E) is an (ϵ, ϵ')-*sampler* if for every $T \subseteq B$ it holds that

$$\Pr_{a\in A}\left(\left|\frac{|\Gamma(a) \cap T|}{d_a} - \frac{|T|}{|B|}\right| > \epsilon\right) < \epsilon',$$

where a is sampled uniformly.

We say that a vertex $a \in A$ *samples* T correctly if $\left|\frac{|\Gamma(a)\cap T|}{d_a} - \frac{|T|}{|B|}\right| \leq \epsilon$. The sampler property implies that there are only a few vertices $S \subseteq A$ that do not sample T correctly.

▶ **Claim 10** (Mixing implies sampler)**.** *If a bipartite graph (A, B, E) is* d*-mixing and d_A-left regular then it is also an $(\epsilon, \frac{2\mathrm{d}^2|B|}{d_A^2\epsilon^2|A|})$-sampler for any $\epsilon > 0$. Specifically, if $d_A = |B|/2$ then the graph is an $(\epsilon, \frac{8\mathrm{d}^2}{|B||A|\epsilon^2})$-sampler for any $\epsilon > 0$.*

Proof. See Claim 13 in [6]. ◀

3 The Low Certainty Framework

In this section we will summarize the main components of the combinatorial framework presented in our earlier work [6].

3.1 Hypotheses Graph

The hypotheses graph associated with an hypothesis class $\mathcal{H}$ and labeled examples $\mathcal{X}$ is a bipartite graph whose vertices are hypotheses in $\mathcal{H}$ and labeled examples in $\mathcal{X}$, and whose edges connect every hypothesis $h \in \mathcal{H}$ to the labeled examples $(x, y) \in \mathcal{X}$ that are consistent with h, i.e., $h(x) = y$.

Let us explore a few examples of hypothesis classes with mixing property.

parity. The hypotheses in $PARITY(n)$ are all the vectors in $\{0,1\}^n$, and the labeled examples are $\{0,1\}^n \times \{0,1\}$ (i.e., $|\mathcal{H}| = 2^n$ and $|\mathcal{X}| = 2 \cdot 2^n$).

▶ **Lemma 11** (Lindsey's Lemma). *Let H be a $n \times n$ matrix whose entries are 1 or -1 and every two rows are orthogonal. Then, for any $S, T \subseteq [n]$,*

$$\left| \sum_{i \in S, j \in T} H_{i,j} \right| \leq \sqrt{|S||T|n}.$$

Lindsey's Lemma and Claim 11 from [6] imply that the hypotheses graph of $PARITY(n)$ is $O(\sqrt{|\mathcal{X}|})$-mixing.

random class. For each hypothesis h and an example x, we have $h(x) = 1$ with probability $1/2$. The hypotheses graph is a random bipartite graph. It is well known that this graph is mixing (see [5]).

We can rephrase Claim 10 for the hypotheses graph and get

▶ **Proposition 12.** *If a graph $(\mathcal{H}, \mathcal{X}, E)$ is d-mixing then it is also $(\epsilon, \frac{8\mathrm{d}^2}{|\mathcal{H}||\mathcal{X}|\epsilon^2})$-sampler for any $\epsilon > 0$.*

3.2 H-expander

The main notion of expansion we will use for the hypotheses graph is H-expander, as we define next (H stands for Hypotheses graph). This notion follows from mixing (Definition 8).

▶ **Definition 13** (H-expander). A left regular bipartite graph (A, B, E) with left degree d_A is an $(\alpha, \beta, \epsilon)$-*H-expander* if for every $T \subseteq B, S \subseteq A$, with $|S| \geq \alpha|A|, |T| \geq \beta|B|$ it holds that

$$\left| |E(S,T)| - \frac{|S||T|}{|B|/d_A} \right| \leq \epsilon|S||T|.$$

For example, the hypotheses graph $(\mathcal{H}, \mathcal{X}, E)$ is left regular with left degree $|\mathcal{X}|/2$, so in this case the denominator $|B|/d_A$ will be equal to 2.

Note the following simple observation that relates mixing and H-expander.

▶ **Proposition 14.** *If a graph $(\mathcal{H}, \mathcal{X}, E)$ is d-mixing then it is also $(\alpha, \beta, \frac{2\mathrm{d}}{\sqrt{\alpha|\mathcal{H}|\beta|\mathcal{X}|}})$ – H-expander, for any $\alpha, \beta \in (0,1)$.*

3.3 Knowledge Graph

▶ **Definition 15** (knowledge graph)**.** The *knowledge graph at time* t of a learning algorithm with memory states $\mathcal{M}$ for an hypothesis class $\mathcal{H}$ is a bipartite *multigraph* $G_t = (\mathcal{H}, \mathcal{M}, E_t)$ where an edge $(h, m) \in E_t$ corresponds to a series of t labeled examples $(x_1, y_1), \ldots, (x_t, y_t)$ with $h(x_i) = y_i$ for every $1 \leq i \leq t$ and the algorithm ends up in memory state m after receiving these t examples.

At each step we will remove a tiny fraction of the edges from the knowledge graph and we focus only on the memories M_t — denote this graph by G'_t. We can read off from this graph the probability $q_t(h, m)$ which indicates the probability that the algorithm reached memory m after t steps and all examples are labeled by h. The probability $q_t(h, m)$ is proportional to the number of edges $E'_t(m, h)$ between a memory m and an hypothesis h in the graph G'_t. We can also observe the conditional probability $q_t(m|h)$ which is the probability that the algorithm reached memory state m given that all the examples observed after t steps are consistent with hypothesis h. We can deduce the probability of a memory m: $q_t(m) = \sum_t q_t(m|h)q_t(h)$. We can also find the probability of a set of memories $M \subseteq \mathcal{M}$, $q_t(M) = \sum_{m \in M} q_t(m)$. If the algorithm, after t steps, is in memory state m, we can deduce the probability that the true hypothesis is h, $q_t(h|m) = \frac{q_t(m|h)q_t(h)}{q_t(m)}$.

3.4 Certainty

Throughout the analysis we will maintain a substantial set of memories $M_t \subseteq \mathcal{M}$ and a set of hypotheses $H_t \subseteq \mathcal{H}$. At time t we pick the underlying hypothesis uniformly from H_t and only consider memories in M_t. Initially, before any labeled example is received, $H_0 = \mathcal{H}$ and M_0 contains all the memories. At later times, H_t and M_t will exclude certain bad hypotheses and memories.

Certainty is a progress measure for the learning algorithm defined as follows:

▶ **Definition 16** (certainty)**.** The *certainty* of a memory m at time t is defined as

$$\sum_h q_t(h|m)^2.$$

The *average certainty* of a set of memories M at time t is defined as

$$cer^t(M) := \sum_{m \in M} q_t(m) \sum_h q_t(h|m)^2.$$

To simplify the notation we write $cer^t(m)$ when we mean $cer^t(\{m\}) = q_t(m) \sum_h q_t(h|m)^2$, i.e., the average certainty with the set $\{m\}$ of memories. We also define a weighted certainty using a weight vector w of length $|\mathcal{M}|$ and each coordinate in w is some value in $[0, 1]$ by

$$cer^t_w(M) = \sum_{m \in M} q_t(m) w_m \cdot q_t^2(h|m).$$

Note that if w is the all 1 vector then $cer^t_w(M) = cer^t(M)$.

At each time t we will focus only on memories that are not too certain, i.e., whose certainty is not much more then the average certainty. Using Markov's inequality we will prove that with high probability the algorithm only reaches these not-too-certain memories. Let us define this set more formally,

$$Bad^c_M = \left\{ m \in M \;\middle|\; \sum_h q_t^2(h|m) > c \cdot cer^t(M_t) \right\},$$

for some $c > 0$, that is of the order $|\mathcal{H}|^{\epsilon}$, for some small constant ϵ. Oftentimes, we will omit c when it is clear from the context. For all $t \geq 1$ we will make sure that M_t will not include Bad_M^c (and additional memories, as will be defined in later sections). The following claims are proved in [6].

▶ **Claim 17.** *For any $c > 0$ and time t, $q_t(Bad_M^c) \leq 1/c$*

There is an equivalent definition of certainty in terms of the certainty of the hypothesis, rather than the memory.

▶ **Claim 18.** *For each memory m, hypothesis h and time t*

$$q_t(m)q_t(h|m)^2 = q_t(h)q_t(h|m)q_t(m|h)$$

In particular we can prove

▶ **Claim 19.** *The average certainty is also equal to*

$$cer^t(M) = \sum_{h\in\mathcal{H}} q_t(h) \sum_{m\in M} q_t(h|m)q_t(m|h).$$

We can therefore define the certainty of an hypothesis h, when focusing on a set of memories M as

$$\sum_{m\in M} q_t(h|m)q_t(m|h)$$

Given the last claim in mind we define

$$Bad_H^c = \{h \in \mathcal{H} \mid \sum_{m\in M_t} q_t(m|h)q_t(h|m) > c \cdot cer^t(M_t)\}.$$

Oftentimes, we will omit c when it is clear from the context.

Define $H_1 = \mathcal{H}$ and for $t > 1$, $H_{t+1} = H_t \setminus Bad_H$. We will define the distribution over the hypotheses at time t by $q_t(h) = \frac{1}{|H_t|}$ if $h \in H_t$, else $q_t(h) = 0$. The next claim proves that H_t is large.

▶ **Claim 20.** *For any $c > 0$, $|H_{t+1}| \geq (1 - 1/c)|H_t|$.*

In the rest of the paper we will prove that the average certainty of M_t, even for a large $t \sim \log c$, will be at most $\frac{c}{|\mathcal{H}|}$, and later we choose $c \sim \log \frac{|\mathcal{H}||\mathcal{X}|}{\mathrm{d}^2}$.

▶ **Claim 21.** *Suppose that the learning algorithm ends after t steps, $|H_t| \geq 3$ and at most γ fraction of the edges were removed from the knowledge graph. Then, there is an hypothesis h such that the probability to correctly return it is at most*

$$3\sqrt{c \cdot cer^t(M_t)} + 3(1 - q_t(M_t)) + \gamma$$

3.5 Representative Labeled Examples

For each memory m at time t, a representative labeled example x is one with $q_t(x|m)$ equal roughly to $\frac{1}{|\mathcal{X}|}$. In particular, given m and the unlabeled example, the probability to guess the label is roughly $1/2$.

▶ **Definition 22.** Let m be a memory state at time t, and let $\epsilon^{rep} > 0$. We say that a labeled example x is ϵ^{rep}-representative at m if

$$\frac{1-\epsilon^{rep}}{|\mathcal{X}|} \leq q_{t+1}(x|m) \leq \frac{1+\epsilon^{rep}}{|\mathcal{X}|}$$

We denote the set of labeled examples that are not ϵ^{rep}-representative at m by $NRep(m, \epsilon^{rep})$.

In [6] a weaker notion of $NRep$ with some specific constant ϵ^{rep} was used.

▶ **Claim 23.** *Let m be a memory in the knowledge graph at time t with certainty bounded by r, i.e., $\sum_h q_t(h|m)^2 \leq r$, assuming the hypotheses graph is an $(\alpha, \beta, \epsilon) - H$-expander, $|NRep(m, 4\sqrt{\alpha|\mathcal{H}|r} + 4\epsilon)| \leq 2\beta$.*

We prove this claim in Section 3.5.1.

3.5.1 Auxiliary Claims

The next claim will imply an equivalent definition for $NRep$.

▶ **Claim 24.** *For any set of labeled examples $S \subseteq \mathcal{X}$ and a memory m it holds that*

$$q_{t+1}(S|m) = \sum_h \Pr(S|h) q_t(h|m).$$

Proof. Using Claim 5 we know that

$$\begin{aligned} q_{t+1}(S|m) &= \sum_h q_{t+1}(S|m,h) q_t(h|m) \\ &= \sum_h \Pr(S|h) q_t(h|m) \end{aligned}$$

◀

Using Claim 24, we know that the not-representative set $NRep(m, \epsilon^{rep})$ is also equal to

$$\left\{ x \in \mathcal{X} | \sum_{h \in \mathcal{H}} \Pr(x|h) q_t(h|m) < \frac{1-\epsilon^{rep}}{|\mathcal{X}|} \right\} \cup \left\{ x \in \mathcal{X} | \sum_{h \in \mathcal{H}} \Pr(x|h) q_t(h|m) > \frac{1+\epsilon^{rep}}{|\mathcal{X}|} \right\}.$$

We would like to prove that $NRep(m, \epsilon^{rep})$ is small for any memory with small certainty. Note that

$$q_t(h|m,x) \propto q_t(h|m) I_{(x,h)\in E},$$

where $I_{(x,h)\in E}$ means that x and h are connected in the hypotheses graph (this follows from Claim 6 with $A = \{h\}, B = \{x\}, C = \{m\}$ and $q_t(x|h,m) = q_t(x|h) = \frac{2}{|\mathcal{X}|} I_{(x,h)\in E}$). This probability distribution can be imagined as if it were constructed by taking the hypotheses graph and adding weight $q_t(h|m)$ to every hypothesis h. Keeping this observation in mind we need some new notation.

Suppose there is a weight w_i for each hypothesis in the hypotheses graph $(\mathcal{H}, \mathcal{X}, E)$. Then, define the weights between sets $S \subseteq \mathcal{H}$ and $T \subseteq \mathcal{X}$ by $w(S,T) := \sum_{s\in S, t\in T} w(s) I_{(s,t)\in E}$ and $w(S) := \sum_{s\in S} w(s)$. We would like to prove that even if there are weights on the hypotheses the hypotheses graph is still pseudo-random. More formally, we will use the following definition.

▶ **Definition 25.** A left regular bipartite graph (A, B, E) is (β, ϵ) – weighted-expander with weights $w_1, \ldots, w_{|A|}$, $\sum_i w_i = 1$, $\forall i, w_i \geq 0$, and left degree d_A if for every $S \subseteq A$ and $T \subseteq B, |T| \geq \beta|B|$ it holds that

$$\left| w(S,T) - \frac{w(S)}{|B|/d_A}|T| \right| \leq \epsilon|T|$$

The next claim proves that any H-expander is a also a weighted-expander assuming low ℓ_2^2 weights.

▶ **Claim 26.** *[see [6]] If the hypotheses graph $(\mathcal{H}, \mathcal{X}, E)$ is an $(\alpha, \beta, \epsilon)$ – H-expander and $\sum_{i=1}^{|\mathcal{H}|} w_i^2 \leq r$ then the hypotheses graph is a $(\beta, 2\epsilon + 2\sqrt{\alpha|\mathcal{H}|r})$ – weighted-expander with weights $w_1, \ldots, w_{|\mathcal{H}|}$.*

Next we will prove our main claim in this section.

Proof of Claim 23. Denote $\epsilon^* = 4\sqrt{\alpha|\mathcal{H}|r} + 4\epsilon$. Define $T_1 = \{x | \sum_{h\in\mathcal{H}} \Pr(x|h)q_t(h|m) < \frac{1-\epsilon^*}{|\mathcal{X}|}\}$ and define weights to hypotheses $w(h) = q_t(h|m)$. From the definition of T_1 we know that

$$\sum_{h\in\mathcal{H}, x\in T_1} \Pr(x|h)q_t(h|m) < \frac{|T_1|(1-\epsilon^*)}{|\mathcal{X}|}.$$

The left term is equal to

$$\sum_{h\in\mathcal{H}, x\in T_1} \frac{2}{|\mathcal{X}|} I_{(x,h)\in E} q_t(h|m) = w(\mathcal{H}, T_1)\frac{2}{|\mathcal{X}|}$$

Assume by a way of contradiction that $|T_1| \geq \beta|\mathcal{X}|$, then Claim 26 implies that

$$\begin{aligned} w(\mathcal{H}, T_1)\frac{2}{|\mathcal{X}|} &\geq \left(\frac{w(\mathcal{H})}{2}|T_1| - 2(\sqrt{\alpha|\mathcal{H}|r} + \epsilon)|T_1|\right)\frac{2}{|\mathcal{X}|} \\ &= \frac{|T_1|}{|\mathcal{X}|} - 2\sqrt{\alpha|\mathcal{H}|r}\frac{2|T_1|}{|\mathcal{X}|} - 2\epsilon\frac{2|T_1|}{|\mathcal{X}|}, \end{aligned}$$

where the equality follows from the fact that $w(\mathcal{H}) = 1$.

Thus

$$\frac{|T_1|(1-\epsilon^*)}{|\mathcal{X}|} > \frac{|T_1|}{|\mathcal{X}|} - 2\sqrt{\alpha|\mathcal{H}|r}\frac{2|T_1|}{|\mathcal{X}|} - 2\epsilon\frac{2|T_1|}{|\mathcal{X}|},$$

$$\Rightarrow 4\sqrt{\alpha|\mathcal{H}|r} + 4\epsilon > \epsilon^*.$$

But the latter contradicts the definition of ϵ^*. Hence we can deduce that $|T_1| < \beta|\mathcal{X}|$.

Similarly, define $T_2 = \{x | \sum_{h\in\mathcal{H}} \Pr(x|h)q_t(h|m) > \frac{1+\epsilon^*}{|\mathcal{X}|}\}$. Assume by a way of contradiction that $|T_2| \geq \beta|\mathcal{X}|$ then

$$\frac{(1+\epsilon^*)|T_2|}{|\mathcal{X}|} < \sum_{h\in\mathcal{H}} \Pr(T_2|h)q_t(h|m) \leq \frac{|T_2|}{|\mathcal{X}|} + 2\sqrt{\alpha|\mathcal{H}|r}\frac{2|T_2|}{|\mathcal{X}|} + 2\epsilon\frac{2|T_2|}{|\mathcal{X}|},$$

where the left inequality follows from the definition of T_2 and the right inequality follows from Claim 26. So again we conclude that $|T_2| < \beta|\mathcal{X}|$. ◀

3.6 Decomposition to Heavy and Many Steps

We show that the certainty does not increase much with a single step of the algorithm. To this end, we decompose almost all the transitions of the bounded space algorithm to two kinds: either a *heavy-sourced* or *many-sourced*. A heavy-sourced memory state at time $t+1$ is one to which the algorithm moves from a memory state at time t via any labeled example from a large family of labeled examples. A many-sourced memory state at time $t+1$ is one that has many possible time-t sources. We analyze each kind of transition separately using H-expansion and K-expansion. For more details see [6].

4 Knowledge Graph Remains Pseudorandom

In this section we define a pseudorandomness property for the knowledge graph and prove that the knowledge graph maintains it throughout the execution of the algorithm, provided that the certainty is low and the hypotheses graph is mixing. To complete the proof we use the pseudorandomness of the knowledge graph to deduce the main theorem by adapting the low certainty framework [6]. For details see [7].

▶ **Definition 27** (enlarging distribution)**.** We say that a distribution p over the memories is (β,γ)-enlarging with respect to a probability distribution q if for every memory m it holds that $p(m) \leq \frac{q(m)}{\beta}$ and if $p(m) > 0$ then $p(m) \geq \frac{q(m)}{\beta} \cdot \gamma$.

β and γ provide a certain measure of the entropy in p. As usual, it is useful to use a logarithmic scale to measure the entropy and our log scale will be with respect to a parameter γ_0 associated with the hypothesis class.

▶ **Definition 28** (entropy-level)**.** The (p,q,β,γ_0)-entropy-level of an element m is defined as

$$e_{\gamma_0}(m) = \log_{\gamma_0} \frac{p(m)\beta}{q(m)}.$$

In other words, if $p(m) = \frac{q_t(m)}{\beta}\gamma_0^i$, then $e_{\gamma_0}(m) = i$.

▶ **Definition 29** (entropy sampler)**.** We say that the knowledge graph G'_t is an $(\alpha,\beta,\ell,\gamma_0,k)-$ entropy sampler if for every $H \subseteq \mathcal{H}$ with $|H| \geq \alpha|\mathcal{H}|$ and every (β,γ_0^k)-enlarging distribution p it holds that

$$\sum_m \Pr(H|m)p(m)2^{e_{\gamma_0}(m)} \leq \ell \cdot \frac{|H|}{|\mathcal{H}|}$$

The usual definition of sampler with multiplicative error is

$$\sum_m \Pr(H|m)p(m) \leq \ell \cdot \frac{|H|}{|\mathcal{H}|}.$$

Our definition requires more and seeks to benefit from memory states whose probability is much lower than $q_t(m^t)/\beta$.

Denote by $S^{m^t,m^{t+1}} \subseteq \mathcal{X}$ the examples that cause the memory to change from m^t to m^{t+1}.

▶ **Claim 30.** *Let $t \geq 1$. Assume that the following conditions hold:*

1. *The hypotheses graph is* d*-mixing.*
2. *The graph G'_t is an $(\alpha',\beta',\ell,\gamma_0,k)-$ entropy sampler.*

3. *All the edges* (m^t, m^{t+1}) *with labeled example* x *in* G'_t *are representative, i.e.,* $q_{t+1}(x|m^t) \notin NRep(m^t, \epsilon^{rep})$.
4. *All memories have low certainty, i.e., for all* m^t *in* G'_t, $cer(m^t) \leq c \cdot cer^t(M_t)$ *and* $cer^t(M_t) \leq c/|\mathcal{H}|$.
5. $\beta' \geq \gamma_0^{k-1}$ *and* $\alpha' \geq 2^{k+2}\sqrt{\gamma_0} + 2^{k+2} \cdot c \cdot \sqrt{\frac{16}{\gamma_0^{11}} \cdot \frac{\mathrm{d}^2}{|\mathcal{X}||\mathcal{H}|}}$.
6. $\epsilon^{rep} \leq 1/2$, *and* $\gamma_0 \leq 1/16$.

Then, G'_{t+1} *is an* $\left(\alpha', \beta', \left(1 + 10\sqrt{\gamma_0} + 2\epsilon^{rep}\right)\ell, \gamma_0, k\right) -$ *entropy sampler*

Proof. We can define a distribution q_{t+1} over pairs $(m^t, S^{m^t,m^{t+1}})$ where m^t is a memory at time t and $S^{m^t,m^{t+1}} \subseteq \mathcal{X}$ is the set of labeled examples that lead from m^t to m^{t+1}, in the following way

$$q_{t+1}(m^t, S^{m^t,m^{t+1}}) := q_t(m^t) q_{t+1}(S^{m^t,m^{t+1}}|m^t).$$

Fix a β'-enlarging distribution p (with respect to q_{t+1}) over memories at time $t+1$ and denote its support by M_{t+1}. For each $m^{t+1} \in M_{t+1}$, denote $p(m^{t+1}) = \frac{q_{t+1}(m^{t+1})}{\beta'_{m^{t+1}}}$, for $\beta'_{m^{t+1}} \geq \beta'$. This induces the distribution $p(m^t, S^{m^t,m^{t+1}}) := \frac{q_t(m^t) q_{t+1}(S^{m^t,m^{t+1}}|m^t)}{\beta'_{m^{t+1}}}$. Indeed,

$$p(m^{t+1}) = \frac{q_{t+1}(m^{t+1})}{\beta'_{m^{t+1}}} = \frac{\sum_{m^t} q_{t+1}(m^t, S^{m^t,m^{t+1}})}{\beta'_{m^{t+1}}} = \sum_{m^t} p(m^t, S^{m^t,m^{t+1}})$$

The probability that p induces on memories at time t is

$$p(m^t) := \sum_{m^{t+1}} p(m^t, S^{m^t,m^{t+1}}) = q_t(m^t) \sum_{m^{t+1}} \frac{q_{t+1}(S^{m^t,m^{t+1}}|m^t)}{\beta'_{m^{t+1}}}.$$

Fix $H \subseteq \mathcal{H}$ with $|H| \geq \alpha'|\mathcal{H}|$. In order to prove the claim, we would like to bound the expression

$$\begin{aligned} &\sum_{m^{t+1} \in M_{t+1}} q_{t+1}(H|m^{t+1}) p(m^{t+1}) 2^{e_{\gamma_0}(m^{t+1})} \\ &\quad = \sum_{m^{t+1} \in M_{t+1}} q_{t+1}(H|m^{t+1}) p(m^{t+1}) 2^{\log_{\gamma_0} \frac{p(m^{t+1})\beta'}{q_{t+1}(m^{t+1})}} \end{aligned} \tag{1}$$

The proof consists of five steps:

Step 1: Rewrite Expression 1 in terms of memories at time t:

Since $p(m^{t+1}) = \frac{q_{t+1}(m^{t+1})}{\beta'_{m^{t+1}}}$, Expression (1) is equal to

$$\sum_{m^{t+1}\in M_{t+1}} q_{t+1}(H|m^{t+1})p(m^{t+1})2^{\log_{\gamma_0}\frac{\beta'}{\beta'_{m^{t+1}}}}$$

$$(\text{dfn. of } m^{t+1}) = \sum_{m^{t+1}\in M_{t+1}} q_{t+1}(H|\vee_{m^t}(m^t, S^{m^t,m^{t+1}}))p(m^{t+1})2^{\log_{\gamma_0}\frac{\beta'}{\beta'_{m^{t+1}}}}$$

$$(\text{Claim 7}) = \sum_{\substack{m^{t+1}\in M_{t+1}\\ m^t\in M_t}} q_{t+1}(H|m^t, S^{m^t,m^{t+1}})\frac{q_{t+1}(m^t, S^{m^t,m^{t+1}})}{q_{t+1}(m^{t+1})}p(m^{t+1})2^{\log_{\gamma_0}\frac{\beta'}{\beta'_{m^{t+1}}}}$$

$$(\text{dfn. of } p) = \sum_{\substack{m^{t+1}\in M\\ m^t\in M_t, h\in H}} q_{t+1}(h|m^t, S^{m^t,m^{t+1}})$$
$$\frac{q_{t+1}(m^t, S^{m^t,m^{t+1}})}{q_{t+1}(m^{t+1})}\frac{q_{t+1}(m^{t+1})}{\beta'_{m^{t+1}}}2^{\log_{\gamma_0}\frac{\beta'}{\beta'_{m^{t+1}}}}$$

$$= \sum_{\substack{m^{t+1}\in M\\ m^t\in M_t, h\in H}} q_{t+1}(h|m^t, S^{m^t,m^{t+1}})\frac{q_t(m^t)q_{t+1}(S^{m^t,m^{t+1}}|m^t)}{\beta'_{m^{t+1}}}2^{\log_{\gamma_0}\frac{\beta'}{\beta'_{m^{t+1}}}}$$

$$(\text{Claim 6}) = \sum_{\substack{m^{t+1}\in M_{t+1}\\ m^t\in M_t, h\in H}} q_t(h|m^t)$$
$$\frac{q_{t+1}(S^{m^t,m^{t+1}}|m^t, h)}{q_{t+1}(S^{m^t,m^{t+1}}|m^t)}\frac{q_t(m^t)q_{t+1}(S^{m^t,m^{t+1}}|m^t)}{\beta'_{m^{t+1}}}2^{\log_{\gamma_0}\frac{\beta'}{\beta'_{m^{t+1}}}}$$

$$(\text{dfn. of } q_{t+1}) = \sum_{m^t\in M_t, h\in H} q_t(h|m^t)q_t(m^t)\sum_{m^{t+1}\in M_{t+1}}\frac{\Pr(S^{m^t,m^{t+1}}|h)}{\beta'_{m^{t+1}}}2^{\log_{\gamma_0}\frac{\beta'}{\beta'_{m^{t+1}}}} \quad (2)$$

In the next steps we will prove that for most memories m^t and for most hypotheses h the term inside the outer sum in (2) is bounded, that is,

$$q_t(h|m^t)q_t(m^t)\sum_{m^{t+1}\in M_{t+1}}\frac{\Pr(S^{m^t,m^{t+1}}|h)}{\beta'_{m^{t+1}}}2^{\log_{\gamma_0}\frac{\beta'}{\beta'_{m^{t+1}}}} \lesssim q_t(h|m^t)p(m^t)2^{e_{\gamma_0}(m^t)} \quad (3)$$

Moreover, the effect of the other memories and hypothesis is negligible. Proving the latter will finish the proof since G'_t is a entropy sampler.

1. In step 2 we show that memories m_t with low $p(m^t)$ do not add much to Expression (2).
2. In step 3 we focus on a memory m_t whose $p(m^t)$ is now low. To show that Inequality (3) holds for most hypotheses h we first recall that since

$$p(m^t) = q_t(m^t)\sum_{m^{t+1}}\frac{q_{t+1}(S^{m^t,m^{t+1}}|m^t)}{\beta'_{m^{t+1}}},$$

we need to prove that

$$\sum_{m^{t+1}}\frac{\Pr(S^{m^t,m^{t+1}}|h)}{\beta'_{m^{t+1}}}2^{\log_{\gamma_0}\frac{\beta'}{\beta'_{m^{t+1}}}} \lesssim \sum_{m^{t+1}}\frac{q_{t+1}(S^{m^t,m^{t+1}}|m^t)}{\beta'_{m^{t+1}}}2^{e_{\gamma_0}(m^t)} \quad (4)$$

In step 3 we show that for most hypotheses h it holds that

$$\Pr(S^{m^t,m^{t+1}}|h) \sim \frac{|S^{m^t,m^{t+1}}|}{|\mathcal{X}|} \sim q_{t+1}(S^{m^t,m^{t+1}}|m^t).$$

3. In step 4 we show that the hypotheses that are not considered in the previous step do not add much to Expression (2).
4. In step 5 we would like to show that Inequality (4) holds. After step 3 and the definition of $e_{\gamma_0}(m)$ this is merely showing that

$$\sum_{m^{t+1}\in M_{t+1}} \frac{\Pr(S^{m^t,m^{t+1}}|m^t)}{\beta'_{m^{t+1}}} 2^{\log_{\gamma_0}\frac{\beta'}{\beta'_{m^{t+1}}}}$$

$$\lesssim \left(\sum_{m^{t+1}} \frac{q_{t+1}(S^{m^t,m^{t+1}}|m^t)}{\beta'_{m^{t+1}}}\right) 2^{\log_{\gamma_0}\beta' \sum_{m^{t+1}} \frac{q_{t+1}(S^{m^t,m^{t+1}}|m^t)}{\beta'_{m^{t+1}}}}$$

   This is proved in step 5 using Jensen's inequality.
5. In step 6 we sum everything up.

Step 2: getting rid of low p-weight memories at time t: In order to use the assumption in the claim regarding the entropy sampler property of G'_t, we need to make sure that for each memory m^t at time t, $p(m^t) = 0$ or $p(m^t) \geq \frac{q_t(m^t)}{\beta'/\gamma_0^k}$. Denote by Low the set of all memories m^t at time t with $0 < p(m^t) < \frac{q_t(m^t)}{\beta'/\gamma_0^k}$. Note that this set has low p-weight

$$p(Low) = \sum_{m^t\in Low} p(m^t) < \sum_{m^t\in Low} q_t(m^t)\frac{\gamma_0^k}{\beta'} \leq \frac{\gamma_0^k}{\beta'} \leq \gamma_0, \tag{5}$$

where the last inequality is true since $\beta' \geq \gamma_0^{k-1}$. Thus, by setting the probability of the memories in Low to 0, the remaining memories need to be multiplied by a factor of at most $1/(1-\gamma_0)$ (i.e., by a factor that is close to 1) so as to make it a distribution again. More formally, we divide the sum that we want to bound, Expression (2), into two sums depending on the membership in Low:

$$\sum_{m^t\in Low, h\in H} q_t(h|m^t)q_t(m^t) \sum_{m^{t+1}\in M_{t+1}} \frac{\Pr(S^{m^t,m^{t+1}}|h)}{\beta'_{m^{t+1}}} 2^{\log_{\gamma_0}\frac{\beta'}{\beta'_{m^{t+1}}}} +$$
$$+ \sum_{m^t\in M_t\setminus Low, h\in H} q_t(h|m^t)q_t(m^t) \sum_{m^{t+1}\in M_{t+1}} \frac{\Pr(S^{m^t,m^{t+1}}|h)}{\beta'_{m^{t+1}}} 2^{\log_{\gamma_0}\frac{\beta'}{\beta'_{m^{t+1}}}} \tag{6}$$

For $m^t \in Low$, the expression $2^{\log_{\gamma_0}\frac{\beta'}{\beta'_{m^{t+1}}}}$ is at most 2^k (since $\frac{q_{t+1}(m^{t+1})}{\beta'_{m^{t+1}}} = p(m^{t+1}) \geq$

$\frac{q_{t+1}(m^{t+1})\gamma_0^k}{\beta'}$ for any m^{t+1}). Thus, the first term in Expression (6) is at most

$$\begin{aligned}
&\sum_{m^t\in Low, h\in H} q_t(h|m^t)q_t(m^t) \sum_{m^{t+1}\in M_{t+1}} \frac{\Pr(S^{m^t,m^{t+1}}|h)}{\beta'_{m^{t+1}}}\cdot 2^k \\
(\text{see Claim 32}) &\leq \sum_{m^t\in Low, h\in H} q_t(h|m^t)q_t(m^t)\cdot \\
&\quad \sum_{m^{t+1}\in M_{t+1}} \frac{2(1+2\epsilon^{rep})q_{t+1}(S^{m^t,m^{t+1}}|m^t)}{\beta'_{m^{t+1}}}\cdot 2^k \\
(\text{definition of } p(m^t)) &= \sum_{m^t\in Low} q_t(H|m^t)2^{k+1}(1+2\epsilon^{rep})p(m^t) \\
(q_t(H|m^t)\leq 1, \epsilon^{rep}\leq 1/2) &\leq 2^{k+2}\sum_{m^t\in Low} p(m^t) \\
(\text{see Inequality (5)}) &\leq 2^{k+2}\gamma_0 \qquad (7)
\end{aligned}$$

Denote $s = p(Low)$. The second term in Expression (6) is equal to

$$(1-s)\sum_{m^t\in M_t\setminus Low, h\in H} q_t(h|m^t)q_t(m^t) \sum_{m^{t+1}\in M_{t+1}} \frac{\Pr(S^{m^t,m^{t+1}}|h)}{(1-s)\beta'_{m^{t+1}}} 2^{\log_{\gamma_0}\frac{\beta'}{\frac{1-s}{1-s}\cdot\beta'_{m^{t+1}}}}$$

which is at most

$$\sum_{m^t\in M_t\setminus Low, h\in H} q_t(h|m^t)q_t(m^t) \sum_{m^{t+1}\in M_{t+1}} \frac{\Pr(S^{m^t,m^{t+1}}|h)}{(1-s)\beta'_{m^{t+1}}} 2^{\log_{\gamma_0}\frac{\beta'}{(1-s)\beta'_{m^{t+1}}}}\cdot 2^{\log_{\gamma_0}1-s}$$

Using Claim 34, $\gamma_0\leq 1/16$, and Inequality (5), it is at most

$$(1+\sqrt{\gamma_0})\sum_{m^t\in M_t\setminus Low, h\in H} q_t(h|m^t)q_t(m^t) \sum_{m^{t+1}\in M_{t+1}} \frac{\Pr(S^{m^t,m^{t+1}}|h)}{(1-s)\beta'_{m^{t+1}}} 2^{\log_{\gamma_0}\frac{\beta'}{(1-s)\beta'_{m^{t+1}}}} \qquad (8)$$

Step 3: $\Pr(S^{m^t,m^{t+1}}|h)\sim\frac{|S^{m^t,m^{t+1}}|}{|\mathcal{X}|}\sim q_{t+1}(S^{m^t,m^{t+1}}|m^t)$: Focus on a memory $m^t\notin Low$. In this step we will prove that for most hypotheses h the term $\Pr(S^{m^t,m^{t+1}}|h)$ can be replaced by $\Pr(S^{m^t,m^{t+1}}|m^t)$. We would like to rewrite the inner sum,

$$\sum_{m^{t+1}\in M_{t+1}} \frac{\Pr(S^{m^t,m^{t+1}}|h)}{\beta'_{m^{t+1}}} 2^{\log_{\gamma_0}\frac{\beta'}{\beta'_{m^{t+1}}}},$$

in Expression (2). For this purpose we first sort all the memories in $m^{t+1}\in M_{t+1}$ according to ascending order of $2^{\log_{\gamma_0}\frac{\beta'}{\beta'_{m^{t+1}}}}/\beta'_{m^{t+1}}$. Denote by β'_i the value $\beta'_{m^{t+1}}$ for m^{t+1} that is the i-th member in the sorted order. Then we get that the inner sum in Expression (2) is equal

to

$$\begin{aligned}\sum_{m^i\in M_{t+1}}\Pr(S^{m^t,m^i}|h)\frac{2^{\log_{\gamma_0}\frac{\beta'}{\beta'_i}}}{\beta'_i} &= \sum_{j\geq 1}\Pr(S^{m^t,m^j}|h)\frac{2^{\log_{\gamma_0}\frac{\beta'}{\beta'_1}}}{\beta'_1}+\\ &+\sum_{j\geq 2}\Pr(S^{m^t,m^j}|h)\left(\frac{2^{\log_{\gamma_0}\frac{\beta'}{\beta'_2}}}{\beta'_2}-\frac{2^{\log_{\gamma_0}\frac{\beta'}{\beta'_1}}}{\beta'_1}\right)+\\ &+\sum_{j\geq 3}\Pr(S^{m^t,m^j}|h)\left(\frac{2^{\log_{\gamma_0}\frac{\beta'}{\beta'_3}}}{\beta'_3}-\frac{2^{\log_{\gamma_0}\frac{\beta'}{\beta'_2}}}{\beta'_2}\right)+\dots\end{aligned}$$

Denote by $S^{m^t,\geq i}$ all the examples that lead from the memory m^t to any of the time-$(t+1)$ memories that are not the first $i-1$ memories. For convenience, define $1/\beta'_0 := 0$. Thus, it holds that

$$\sum_{m_i\in M_{t+1}}\Pr(S^{m^t,m_i}|h)\frac{2^{\log_{\gamma_0}\frac{\beta'}{\beta'_i}}}{\beta'_i}=\sum_{i\geq 1}\Pr(S^{m^t,\geq i}|h)\left(\frac{2^{\log_{\gamma_0}\frac{\beta'}{\beta'_i}}}{\beta'_i}-\frac{2^{\log_{\gamma_0}\frac{\beta'}{\beta'_{i-1}}}}{\beta'_{i-1}}\right).$$

We divide this sum into two, using index $i_{(m^t)}$ which is the largest i such that $|S^{m^t,\geq i}|\geq\epsilon'|\mathcal{X}|$, for ϵ' to be determined later.

$$\begin{aligned}&\sum_{i=1}^{i_{(m^t)}}\Pr(S^{m^t,\geq i}|h)\left(\frac{2^{\log_{\gamma_0}\frac{\beta'}{\beta'_i}}}{\beta'_i}-\frac{2^{\log_{\gamma_0}\frac{\beta'}{\beta'_{i-1}}}}{\beta'_{i-1}}\right)+\\ &\sum_{i=(i_{(m^t)})+1}^{|M_{t+1}|}\Pr(S^{m^t,\geq i}|h)\left(\frac{2^{\log_{\gamma_0}\frac{\beta'}{\beta'_i}}}{\beta'_i}-\frac{2^{\log_{\gamma_0}\frac{\beta'}{\beta'_{i-1}}}}{\beta'_{i-1}}\right)\end{aligned}\tag{9}$$

Let us start with bounding the first term in Equation (9). From Claim 33, we know that except for a fraction of $\frac{1}{\epsilon^2}\cdot\frac{\mathrm{d}^2}{|\mathcal{X}||\mathcal{H}|}$ hypotheses $h\in\mathcal{H}$ for each $i\leq(1-\epsilon')|\mathcal{X}|$,

$$\Pr(S^{m^t,\geq i}|h)\leq\left(1+\epsilon'+\frac{4\epsilon}{(\epsilon')^2}\right)\frac{|S^{m^t,\geq i}|}{|\mathcal{X}|},\tag{10}$$

for $\epsilon>0$ to be determined later. From Claim 32 we know that the RHS is at most

$$\left(1+\epsilon'+\frac{4\epsilon}{(\epsilon')^2}\right)(1+2\epsilon^{rep})q_{t+1}(S^{m^t,\geq i}|m^t)$$

Denote the set of hypotheses that the bound in Inequality (10) does not apply to by $Err(m^t)$. We know that

$$\frac{|Err(m^t)|}{|\mathcal{H}|}\leq\frac{1}{\epsilon^2}\cdot\frac{\mathrm{d}^2}{|\mathcal{X}||\mathcal{H}|}\tag{11}$$

Let us now bound the second term in Expression (9). For each $i>i_{(m^t)}$ we use the simple bound given in Claim 32:

$$\Pr(S^{m^t,\geq i}|h)\leq 2(1+2\epsilon^{rep})q_{t+1}(S^{m^t,\geq i}|m^t).\tag{12}$$

We can now rewrite Expression (9) using Inequalities 10 and 12. Namely, for $m^t \notin Low$ and $h \notin Err(m^t)$ Expression (9) is at most

$$\left(1+\epsilon'+\frac{4\epsilon}{(\epsilon')^2}\right)(1+2\epsilon^{rep})\left[\sum_{i=1}^{i_{(m^t)}} q_{t+1}(S^{m^t,\geq i}|m^t)\left(\frac{2^{\log_{\gamma_0}\frac{\beta'}{\beta'_i}}}{\beta'_i}-\frac{2^{\log_{\gamma_0}\frac{\beta'}{\beta'_{i-1}}}}{\beta'_{i-1}}\right)+\right.$$
$$\left.\sum_{i=(i_{(m^t)})+1}^{|M_{t+1}|} 2\cdot q_{t+1}(S^{m^t,\geq i}|m^t)\left(\frac{2^{\log_{\gamma_0}\frac{\beta'}{\beta'_i}}}{\beta'_i}-\frac{2^{\log_{\gamma_0}\frac{\beta'}{\beta'_{i-1}}}}{\beta'_{i-1}}\right)\right]$$

Which is equal to

$$\left(1+\epsilon'+\frac{4\epsilon}{(\epsilon')^2}\right)(1+2\epsilon^{rep})\cdot$$
$$\left[\sum_{i=1}^{i_{(m^t)}} q_{t+1}(S^{m^t,m_i}|m^t)\frac{2^{\log_{\gamma_0}\frac{\beta'}{\beta'_i}}}{\beta'_i}+\sum_{i=(i_{(m^t)})+1}^{|M_{t+1}|} q_{t+1}(S^{m^t,m_i}|m^t)\frac{2\cdot 2^{\log_{\gamma_0}\frac{\beta'}{\beta'_i}}}{\beta'_i}\right] \tag{13}$$

Step 4: getting rid of "bad" hypotheses: We would like to bound the portion of Expression (2) that involves $h \in Err(m^t)$ for some m^t. Namely, we would like to bound

$$\sum_{m^t\in M_t, h\in Err(m^t)} q_t(h|m^t)q_t(m^t)\sum_{m^{t+1}\in M_{t+1}}\frac{\Pr(S^{m^t,m^{t+1}}|h)}{\beta'_{m^{t+1}}}2^{\log_{\gamma_0}\frac{\beta'}{\beta'_{m^{t+1}}}}. \tag{14}$$

For any m^{t+1}, from the definition of p we know that $\frac{q_{t+1}(m^{t+1})}{\beta'_{m^{t+1}}} = p(m^{t+1}) \geq \frac{q_{t+1}(m^{t+1})\gamma_0^k}{\beta'}$, hence $2^{\log_{\gamma_0}\frac{\beta'}{\beta'_{m^{t+1}}}} \leq 2^k$. Hence Expression (14) is at most

$$\sum_{m^t\in M_t, h\in Err(m^t)} q_t(h|m^t)q_t(m^t)\sum_{m^{t+1}\in M_{t+1}}\frac{\Pr(S^{m^t,m^{t+1}}|h)}{\beta'_{m^{t+1}}}2^k.$$

From Claim 32 we know that $\frac{\Pr(S^{m^t,m^{t+1}}|h)}{\beta'_{m^{t+1}}} \leq \frac{4q_{t+1}(S^{m^t,m^{t+1}}|m^t)}{\beta'_{m^{t+1}}}$. Hence, Expression (14) is at most

$$\begin{aligned}\sum_{\substack{m^t\in M_t\\ h\in Err(m^t)}} q_t(h|m^t)q_t(m^t) & \sum_{m^{t+1}\in M_{t+1}}\frac{q_{t+1}(S^{m^t,m^{t+1}}|m^t)}{\beta'_{m^{t+1}}}2^{k+2}\\ &= \sum_{\substack{m^t\in M_t\\ h\in Err(m^t)}} q_t(h|m^t)p(m^t)2^{k+2}\\ &\leq 2^{k+2}\sum_{m^t\in M_t} p(m^t)q_t(Err(m^t)|m^t).\end{aligned}$$

From Claim 4 and Inequality (11) we know that

$$q_t(Err(m^t)|m^t) \leq \sqrt{|Err(m^t)|c\cdot cer^t(M_t)} \leq c\cdot\sqrt{\frac{1}{\epsilon^2}\cdot\frac{\mathrm{d}^2}{|\mathcal{X}||\mathcal{H}|}},$$

where the second inequality follows from Inequality (11) and the assumption in the claim regarding the bound on $cer^t(M_t)$. To sum up this step, $Err(m^t)$ adds only a small additive error of $2^{k+2}\cdot c\cdot\sqrt{\frac{1}{\epsilon^2}\cdot\frac{\mathrm{d}^2}{|\mathcal{X}||\mathcal{H}|}}$ to Expression (2).

Step 5: towards using the entropy sampler property of G'_t: Recall that according to our plan at step 1 we want to prove now that for $m^t \notin Low, h \notin Err(m^t)$ it holds that

$$\sum_{m_j \in M_{t+1}} \frac{\Pr(S^{m^t,m_j}|m^t)}{\beta'_{m_j}} 2^{\log_{\gamma_0} \frac{\beta'}{\beta'_{m_j}}}$$

is at most

$$\left(\sum_{m_j} \frac{q_{t+1}(S^{m^t,m_j}|m^t)}{\beta'_{m_j}}\right) 2^{\log_{\gamma_0} \beta' \sum_{m_j} \frac{q_{t+1}(S^{m^t,m_j}|m^t)}{\beta'_{m_j}}} (1+\epsilon'_4)$$

for some small $\epsilon'_4 \in (0,1)$ to (implicitly) be determined in the next step. To this end we first prove, having in mind the expression in 13, that the following inequality holds

$$\frac{\sum_{i=1}^{i_{(m^t)}} \frac{q_{t+1}(S^{m^t,m_i}|m^t)}{\beta'_i} 2^{\log_{\gamma_0} \frac{\beta'}{\beta'_i}} + \sum_{i=(i_{(m^t)})+1}^{|M_{t+1}|} \frac{q_{t+1}(S^{m^t,m_i}|m^t)}{\beta'_i} 2^{\log_{\gamma_0} \frac{\beta' \cdot \gamma_0}{\beta'_i}}}{\sum_{m_j} \frac{q_{t+1}(S^{m^t,m_j}|m^t)}{\beta'_{m_j}}}$$
$$\leq 2^{\log_{\gamma_0} \beta' \sum_{m_j} \frac{q_{t+1}(S^{m^t,m_j}|m^t)}{\beta'_{m_j}}} (1+\epsilon_4) \tag{15}$$

for some small $\epsilon_4 \in (0,1)$ to be determined later.

Define the function $f(x) = 2^{\log_{\gamma_0} \frac{1}{x}}$ and the following distribution over memories at time $t+1$: $\bar{p}(m^i) \propto \frac{q_{t+1}(S^{m^t,m^i}|m^t)}{\beta'_{m^i}}$ and divide both sides by $2^{\log_{\gamma_0} \beta'}$ then Inequality (15) can be rewritten as

$$\sum_{m_i} \bar{p}(m_i) f\left(\beta'_i \cdot \left(\frac{1}{\gamma_0}\right)^{I_{i>i_{(m^t)}}}\right) \leq f\left(\left(\frac{1}{\gamma_0}\right)^{\log(1+\epsilon_4)} / \sum_{m_j} \frac{q_{t+1}(S^{m^t,m_j}|m^t)}{\beta'_{m_j}}\right),$$

where I is the indicator function. Use Jensen's inequality with the concave function f (see Claim 31) and get that the LHS is at most

$$f\left(\sum_{m_i} \frac{q_{t+1}(S^{m^t,m_i}|m^t)}{\sum_{m_j} \frac{q_{t+1}(S^{m^t,m_j}|m^t)}{\beta'_{m_j}}} \cdot \left(\frac{1}{\gamma_0}\right)^{I_{i>i_{(m^t)}}}\right)$$

Since f is monotonically increasing (see Claim 31), to prove Inequality (15) it is enough to show that

$$\sum_{m_i} q_{t+1}(S^{m^t,m_i}|m^t) \cdot \left(\frac{1}{\gamma_0}\right)^{I_{i>i_{(m^t)}}} \leq \left(\frac{1}{\gamma_0}\right)^{\log(1+\epsilon_4)}$$

Using the inequality $x/2 \leq \log(1+x)$ (which follows from Fact 35 and $\epsilon_4 < 1$) it is enough to prove that

$$\sum_{m_i} q_{t+1}(S^{m^t,m_i}|m^t) \cdot \left(\frac{1}{\gamma_0}\right)^{I_{i>i_{(m^t)}}} \leq \left(\frac{1}{\gamma_0}\right)^{\epsilon_4/2}. \tag{16}$$

Note that by separating the LHS into two and the definition of ϵ' we have that

$$\sum_{m_i} q_{t+1}(S^{m^t,m_i}|m^t)\cdot\left(\frac{1}{\gamma_0}\right)^{I_{i>i_{(m^t)}}} \leq 1+\sum_{i>i_{(m^t)}} q_{t+1}(S^{m^t,m_i}|m^t)\left(\frac{1}{\gamma_0}\right) \leq 1+\epsilon'\left(\frac{1}{\gamma_0}\right)$$

Thus, to show that Inequality (16) holds, it suffices to show that

$$1+\epsilon'\left(\frac{1}{\gamma_0}\right) \leq \left(\frac{1}{\gamma_0}\right)^{\epsilon_4/2}.$$

Which is true if and only if

$$\ln\left(1+\epsilon'\left(\frac{1}{\gamma_0}\right)\right) \leq \frac{\epsilon_4}{2}\ln\left(\frac{1}{\gamma_0}\right).$$

Using Fact 35 it is enough to show that

$$\epsilon'\left(\frac{1}{\gamma_0}\right) \leq \frac{\epsilon_4}{2}\ln\left(\frac{1}{\gamma_0}\right).$$

We choose $\epsilon_4 = 2\sqrt{\epsilon'}$. If $\sqrt{\epsilon'} \leq \gamma_0$ then the inequality will hold since $\gamma_0 \leq 1/16 < 1/e$.

Step 6: Summing up: Using Expressions (8), (13), (15) (recall that $\epsilon_4 = 2\sqrt{\epsilon'}$), the assumption is the claim regarding the entropy sampler of G'_t, Expression (7), and the conclusion of step 4 we have proven that Expression (1) is bounded by

$$(1+\sqrt{\gamma_0})\left(1+\epsilon'+\frac{4\epsilon}{(\epsilon')^2}\right)(1+2\epsilon^{rep})(1+2\sqrt{\epsilon'})\ell\cdot\frac{|H|}{|\mathcal{H}|}+2^{k+2}\gamma_0+2^{k+2}\cdot c\cdot\sqrt{\frac{1}{\epsilon^2}\cdot\frac{\mathrm{d}^2}{|\mathcal{X}||\mathcal{H}|}}$$

We choose $\epsilon' = \gamma_0^2$ (note that indeed $\sqrt{\epsilon'} \leq \gamma_0$) and $\epsilon = \gamma_0^5/4$. From the assumption in the claim we know that $\alpha'\sqrt{\gamma_0} \geq 2^{k+2}\gamma_0 + 2^{k+2}\cdot c\cdot\sqrt{\frac{16}{\gamma_0^{10}}\cdot\frac{\mathrm{d}^2}{|\mathcal{X}||\mathcal{H}|}}$. Hence, Expression (1) is at most

$$\left((1+\sqrt{\gamma_0})\left(1+\gamma_0^2+\gamma_0\right)(1+2\epsilon^{rep})(1+2\gamma_0)\ell+\sqrt{\gamma_0}\right)\cdot\frac{|H|}{|\mathcal{H}|} \leq (1+10\sqrt{\gamma_0}+2\epsilon^{rep})\,\ell\cdot\frac{|H|}{|\mathcal{H}|}$$

(in the RHS the constant 10 near $\sqrt{\gamma_0}$ was chosen arbitrarily) ◀

4.1 Auxiliary Claims

▶ **Claim 31.** *For any $\epsilon \leq 1/2$, the function $f(x) = 2^{\log_\epsilon \frac{1}{x}}$ for $x > 0$ is monotonically increasing and concave.*

In the next claim we lower bound $q_{t+1}(S|m^t)$ in terms of $\Pr(S|h)$ via the term $|S|/|\mathcal{X}|$.

▶ **Claim 32.** *Let $S \subseteq \mathcal{X}$. Let $h \in \mathcal{H}$.*

1. $\Pr(S|h) \leq \frac{2|S|}{|\mathcal{X}|}$
2. *Let m^t be a memory at time t. Assume $S \cap NRep(m^t, \epsilon^{rep}) = \emptyset$ and $\epsilon^{rep} \leq 1/2$. Then $\frac{|S|}{|\mathcal{X}|} \leq (1+2\epsilon^{rep})q_{t+1}(S|m^t)$.*

Proof. The first inequality follows from the fact that if $(x,h) \in E$ (i.e., hypothesis h and labeled example x are consistent) then $\Pr(x|h) = 2/|\mathcal{X}|$ and if $(x,h) \notin E$ then $\Pr(x|h) = 0$. To prove the second inequality, we use the definition of $NRep$ (see Definition 22) to deduce that

$$\frac{1-\epsilon^{rep}}{|\mathcal{X}|}|S| \leq q_{t+1}(S|m^t) \Rightarrow \frac{|S|}{|\mathcal{X}|} \leq \frac{1}{1-\epsilon^{rep}}q_{t+1}(S|m^t) \Rightarrow \frac{|S|}{|\mathcal{X}|} \leq (1+2\epsilon^{rep})q_{t+1}(S|m^t),$$

where the last inequality is true for $\epsilon^{rep} \leq 1/2$. ◀

Suppose that the labeled examples are sorted in some way and denote by $S^{\geq i}$ all the examples except the first $i-1$ examples.

▶ **Claim 33.** *If the hypotheses graph $(\mathcal{H},\mathcal{X},E)$ is* d*-mixing, then for any $\epsilon,\epsilon'>0$ except for a fraction of $\frac{1}{\epsilon^2}\cdot\frac{\mathrm{d}^2}{|\mathcal{X}||\mathcal{H}|}$ of the hypotheses $h\in\mathcal{H}$, for each $i\leq(1-\epsilon')|\mathcal{X}|$,*

$$\Pr(S^{\geq i}|h)\leq\left(1+\epsilon'+\frac{4\epsilon}{(\epsilon')^2}\right)\frac{|S^{\geq i}|}{|\mathcal{X}|}.$$

Proof. We will pick $\epsilon_1,\epsilon_2,\epsilon_3>0$ at the end. Divide all the labeled examples into $1/\epsilon_2$ consecutive equal parts, each of size $\epsilon_2|\mathcal{X}|$ (without loss of gnerality the integer $\epsilon_2|\mathcal{X}|$ divides $|\mathcal{X}|$). Focus for now on some part S. First we would like to show that for each part $S\subseteq\mathcal{X}$ most hypotheses h do not over-sample S, i.e.,

$$\Pr(S|h)\leq(1+\epsilon_1)\frac{|S|}{|\mathcal{X}|}.$$

Denote by $T\subseteq\mathcal{H}$ all the hypotheses $h\in\mathcal{H}$ such that $\Pr(S|h)>\frac{|S|}{|\mathcal{X}|}(1+\epsilon_1)$. Then $E(S,T)>\frac{|S|}{|\mathcal{X}|}(1+\epsilon_1)\frac{|\mathcal{X}|}{2}|T|$. From the d-mixing property we know that $E(S,T)\leq|S||T|/2+\mathrm{d}\sqrt{|S||T|}$. Combining these two inequalities we get that

$$\epsilon_1\frac{|S||T|}{2}<\mathrm{d}\sqrt{|S||T|}\Rightarrow|T|<\frac{4\mathrm{d}^2}{\epsilon_1^2|S|}=\frac{4\mathrm{d}^2}{\epsilon_1^2\epsilon_2|\mathcal{X}|}.$$

Denote by $Err\subseteq\mathcal{H}$ all the hypotheses that over-sample at least one part, i.e., hypothesis $h\notin Err$ if and only if for each of the $1/\epsilon_2$ parts, S, it holds that $\Pr(S|h)\leq(1+\epsilon_1)\frac{|S|}{|\mathcal{X}|}$. We can easily deduce, using a union bound, that the fraction of this set is at most $\frac{|Err|}{|\mathcal{H}|}\leq\frac{4\mathrm{d}^2}{\epsilon_1^2\epsilon_2^2|\mathcal{X}||\mathcal{H}|}$.

Let us go back to the expressions that we want to bound, namely $\Pr(S^{\geq i}|h)$ for each i. We will show that for each $h\notin Err$, and for each i, the probability

$$\Pr(S^{\geq i}|h)\leq(1+\epsilon_3)\frac{|S^{\geq i}|}{|\mathcal{X}|}. \tag{17}$$

For each i denote by i^* the largest index that is smaller than i and divides $\epsilon_2|\mathcal{X}|$. We have that $\Pr(x|h)\leq\frac{2}{|\mathcal{X}|}$ for each labeled example x and hypothesis h, thus $\Pr(S^{\geq i}\setminus S^{\geq i^*}|h)\leq2\epsilon_2$. Hence, the LHS of Inequality (17) is bounded by

$$\Pr(S^{\geq i}|h)\quad\leq\quad\Pr(S^{\geq i^*}|h)+2\epsilon_2\leq(1+\epsilon_1)\frac{|S^{\geq i}|}{|\mathcal{X}|}+2\epsilon_2,$$

So we need to make sure that

$$(1+\epsilon_1)\frac{|S^{\geq i}|}{|\mathcal{X}|}+2\epsilon_2\leq(1+\epsilon_3)\frac{|S^{\geq i}|}{|\mathcal{X}|},$$

which will happen only if $\epsilon_1\frac{|S^{\geq i}|}{|\mathcal{X}|}+2\epsilon_2\leq\epsilon_3\frac{|S^{\geq i}|}{|\mathcal{X}|}$, or equivalently $\frac{2\epsilon_2}{\epsilon_3-\epsilon_1}\leq\frac{|S^{\geq i}|}{|\mathcal{X}|}$ (assuming $\epsilon_3>\epsilon_1$ as we will choose later). Thus, except for a fraction of $\frac{4\mathrm{d}^2}{\epsilon_1^2\epsilon_2^2|\mathcal{X}||\mathcal{H}|}$ hypotheses $h\in\mathcal{H}$ for each $i\leq(1-\frac{2\epsilon_2}{\epsilon_3-\epsilon_1})|\mathcal{X}|$,

$$\Pr(S^{\geq i}|h)\leq(1+\epsilon_3)\frac{|S^{\geq i}|}{|\mathcal{X}|}.$$

Choose $\epsilon_1=\epsilon'$ and $\epsilon_2=\frac{2\epsilon}{\epsilon_1}$ and $\epsilon_3=\epsilon_1+\frac{2\epsilon_2}{\epsilon_1}$. ◀

▶ **Claim 34.** *For any $0<x\leq1/16$ it holds that $2^{\log_x(1-x)}\leq1+\sqrt{x}$.*

▶ **Fact 35.** *For any $x>-1$ it holds that $\frac{x}{1+x}\leq\ln(1+x)\leq x$.*

References

1. P. Beame, S. O. Gharan, and X. Yang. Time-space tradeoffs for learning from small test spaces: Learning low degree polynomial functions. Technical report, ECCC, 2017.
2. B. Chazelle. *The Discrepancy Method: Randomness and Complexity*. Randomness and Complexity. Cambridge University Press, 2000.
3. S. Garg, R. Raz, and A. Tal. Extractor-based time-space lower bounds for learning. Technical report, ECCC, 2017.
4. G. Kol, R. Raz, and A. Tal. Time-space hardness of learning sparse parities. In *Proc. 49th ACM Symp. on Theory of Computing*, 2017.
5. M. Krivelevich and B. Sudakov. Pseudo-random graphs. In *More sets, graphs and numbers*, pages 199–262. Springer, 2006.
6. D. Moshkovitz and M. Moshkovitz. Mixing implies lower bounds for space bounded learning. Technical report, ECCC Report TR17-017, 2017.
7. D. Moshkovitz and M. Moshkovitz. Mixing implies strong lower bounds for space bounded learning. Technical Report TR17-116, ECCC, 2017.
8. R. Raz. Fast learning requires good memory: A time-space lower bound for parity learning. In *Proc. 57th IEEE Symp. on Foundations of Computer Science*, 2016.
9. R. Raz. A time-space lower bound for a large class of learning problems. In *Proc. 58th IEEE Symp. on Foundations of Computer Science*, 2017.

Pseudorandom Generators for Low-Sensitivity Functions

Pooya Hatami[*1] **and Avishay Tal**[†2]

1 **University of Texas at Austin, Austin, TX, USA**
pooyahat@gmail.com
2 **Stanford University, Palo Alto, CA, USA**
avishay.tal@gmail.com

Abstract

A Boolean function is said to have maximal sensitivity s if s is the largest number of Hamming neighbors of a point which differ from it in function value. We initiate the study of pseudorandom generators fooling low-sensitivity functions as an intermediate step towards settling the sensitivity conjecture. We construct a pseudorandom generator with seed-length $2^{O(\sqrt{s})} \cdot \log(n)$ that fools Boolean functions on n variables with maximal sensitivity at most s. Prior to our work, the (implicitly) best pseudorandom generators for this class of functions required seed-length $2^{O(s)} \cdot \log(n)$.

1998 ACM Subject Classification F. Theory of Computation

Keywords and phrases Pseudorandom Generator, Sensitivity, Sensitivity Conjecture

Digital Object Identifier 10.4230/LIPIcs.ITCS.2018.29

1 Introduction

The sensitivity of a Boolean function $f : \{-1,1\}^n \to \{-1,1\}$ at a point $x \in \{-1,1\}^n$, denoted $s(f,x)$, is the number of neighbors of x in the Hypercube whose f-value is different than $f(x)$. The maximal sensitivity of f, denoted $s(f)$, is the maximum over $s(f,x)$ for all $x \in \{-1,1\}^n$. The sensitivity conjecture by Nisan and Szegedy [10, 11] asserts that low-sensitivity functions (also called "smooth" functions) are "easy". More precisely, the conjecture states that any Boolean function whose maximal sensitivity is s can be computed by a decision tree of depth $\mathrm{poly}(s)$. The conjecture remains wide open for several decades now, and the state-of-the-art upper bounds on decision tree complexity are merely $\exp(O(s))$.

Assuming the sensitivity conjecture, low-sensitivity functions are not any stronger than low-depth decision trees, which substantially limits their power. Hence, towards settling the conjecture, it is natural to inspect how powerful low-sensitivity functions are. One approach that follows this idea aims to prove limitations of low-sensitivity functions, which follow from the sensitivity conjecture, unconditionally. This line of work was initiated recently by Gopalan et al. [7], who considered low-sensitivity functions as a complexity class. Denote by $\mathsf{Sens}(s)$ the class of Boolean functions with sensitivity at most s. The sensitivity conjecture

* This work was conducted while the author was a member at the IAS and a postdoc at DIMACS. Supported by a Simons Investigator Award (#409864, David Zuckerman) and National Science Foundation grants CCF-1412958 and CCF-1445755.
† This work was conducted while the author was a member at the IAS. Supported by the Simons Collaboration on Algorithms and Geometry, and by the National Science Foundation grant No. CCF-1412958.

asserts that $\mathsf{Sens}(s) \subseteq \mathsf{DecTree\text{-}depth}(\mathrm{poly}(s))$, which then implies

$$\mathsf{Sens}(s) \subseteq \mathsf{DecTree\text{-}depth}(\mathrm{poly}(s)) \subseteq \mathsf{DNF\text{-}size}(2^{\mathrm{poly}(s)}) \subseteq \mathsf{AC}^0\text{-}\mathsf{size}(2^{\mathrm{poly}(s)}) \subseteq \mathsf{Formula\text{-}depth}(\mathrm{poly}(s)) \subseteq \mathsf{Circuit\text{-}size}(2^{\mathrm{poly}(s)}) ,$$

whereas Gopalan et al. [7] proved that $\mathsf{Sens}(s) \subseteq \mathsf{Formula\text{-}depth}(\mathrm{poly}(s))$ unconditionally. It remains open to prove that $\mathsf{Sens}(s)$ is contained in smaller complexity classes such as $\mathsf{AC}^0\text{-}\mathsf{size}(2^{\mathrm{poly}(s)})$ or even $\mathsf{TC}^0\text{-}\mathsf{size}(2^{\mathrm{poly}(s)})$.

One consequence of the sensitivity conjecture is the existence of pseudorandom generators (PRGs) with short seeds fooling low-sensitivity functions. This is since a depth d decision tree has ℓ_1 norm at most 2^d in Fourier domain, so is ϵ fooled by $\frac{\epsilon}{2^d}$-biased spaces. Thus, since under the conjecture $d \leq \mathrm{poly}(s)$, the standard construction of $\frac{\epsilon}{2^{poly(s)}}$-biased spaces gives a PRG with seed length $\mathrm{poly}(s) \cdot \log(1/\epsilon) + \log n$ fooling $\mathsf{Sens}(s)$.[1] The goal of our work is to construct PRGs fooling $\mathsf{Sens}(s)$ unconditionally. (As stated above, this is a necessary hurdle to overcome before proving the conjecture.) We fall short of achieving seed length $\mathrm{poly}(s) \cdot \log(n)$ and get the weaker seed length of $2^{O(\sqrt{s})} \cdot \log(n)$. Nonetheless, prior to our work, only seed-length $2^{O(s)} \cdot \log(n)$ was known, which follows implicitly from the state of the art upper bounds on degree in terms of sensitivity $\deg(f) \leq 2^{s(1+o(1))}$ [4].

Hardness vs Randomness? We note an unusual phenomenon in the hardness vs randomness paradigm with respect to the class $\mathsf{Sens}(s)$. The paradigm of Hardness vs Randomness, initiated by Nisan and Wigderson [12], asserts that PRGs and average-case lower bounds are essentially equivalent, for almost all reasonable complexity classes. For example, the average-case lower bound of Håstad [9] for the parity function by $\mathbf{AC^0}$ circuits implies a pseudorandom generator fooling $\mathbf{AC^0}$ circuits with poly-logarithmic seed-length. This general transformation of hardness to randomness is achieved via the NW-generator, which constructs a PRG based on the hard function. In [8], it was proved that low-sensitivity functions can be ε-approximated by real polynomials of degree $O(s \cdot \log(1/\varepsilon))$, which implies that the parity function on n variables can only have agreement $1/2 + 2^{-\Omega(n/s)}$ with Boolean functions of sensitivity s. In other words, the parity function on n variables is average-case hard for the class $\mathsf{Sens}(s)$. It thus seems very tempting to use the parity function in the NW-generator to construct a PRG fooling $\mathsf{Sens}(s)$, however, the proof does not follow through since the class of low-sensitivity functions is not closed under the transformations made by the analysis of the NW-generator (in particular it is not closed under identifying a set of the input variables with one variable). We do not claim that the NW-generator with the parity function does not fool $\mathsf{Sens}(s)$, but we point out that the argument in the standard proof breaks. (See more details in Appendix A).

1.1 Our Results

A function $G : \{-1,1\}^r \to \{-1,1\}^n$ is said to be a pseudorandom generator with seed-length r that ε-fools a class of Boolean functions $\mathcal{C}$ if for every $f \in \mathcal{C}$:

$$\left| \mathop{\mathbf{E}}_{z \in_R \{-1,1\}^r} [f(G(z))] - \mathop{\mathbf{E}}_{x \in_R \{-1,1\}^n} [f(x)] \right| \leq \varepsilon .$$

[1] Even under the weaker conjecture $\mathsf{Sens}(s) \subseteq \mathsf{AC}^0\text{-}\mathsf{size}(n^{\mathrm{poly}(s)})$, we would get that $\mathrm{poly}(s, \log n)$-wise independence fools $\mathsf{Sens}(s)$ via the result of [6].

In other words, any $f \in \mathcal{C}$ cannot distinguish (with advantage greater than ε) between an input sampled according to the uniform distribution over $\{-1,1\}^n$ and an input sampled according to the uniform distribution over $\{-1,1\}^r$ and expanded to an n-bit string using G.

The main contribution of this paper is the first pseudorandom generator for low-sensitivity Boolean functions with subexponential seed length in the sensitivity.

▶ **Theorem 1.** *There is a distribution $\mathcal{D}$ on $\{-1,1\}^n$ with seed-length $2^{O(\sqrt{s+\log(1/\varepsilon)})} \cdot \log(n)$ that ε-fools every $f : \{-1,1\}^n \to \{-1,1\}$ with $s(f) = s$.*

We prove the following strengthening of Friedgut's Theorem for low-sensitivity functions that is essential to our construction. (In the following, we denote by $\mathbf{W}^{\geq k}[f] = \sum_{S \subseteq [n], |S| \geq k} \hat{f}(S)^2$.)

▶ **Lemma 2.** *Let $f : \{-1,1\}^n \to \{-1,1\}$ with $s(f) \leq s$. Let $1 \leq k \leq s/10$. Assume $\mathbf{W}^{\geq k}[f] \leq 2^{-6s}$, and that at most 2^{-6s} fraction of the points in $\{-1,1\}^n$ have sensitivity at least k. Then, f is a 2^{20k}-junta.*

1.2 Proof Outline

Below we give a sketch of our proof of Theorem 1.

Similar to a construction of Ajtai and Wigderson [1], and more recent examples [14, 17], our pseudorandom generator involves repeated applications of "pseudorandom restrictions". Using Lemma 2 and studying the behavior of the Fourier spectrum of low-sensitivity functions under pseudorandom restrictions, we are able to prove the following. Let $f : \{-1,1\}^n \to \{-1,1\}$ be a Boolean function, let $S \subseteq [n]$ be randomly selected according to a k-wise independent distribution such that $|S| \approx pn$, and let $x_{\overline{S}} = (x_i)_{i \notin S} \in \{-1,1\}^{|S|}$ be selected uniformly at random. Then

$$\Pr_{S, x_{\overline{S}}}[f(x_{\overline{S}}, .) \text{ is not a } 2^{20k}\text{-junta}] \leq O(ps)^k \cdot 2^{6s}. \tag{1}$$

Since every 2^{20k}-junta is fooled by an almost 2^{20k}-wise independent distribution, we will fill the x_S coordinates according to efficient constructions of such distributions due to [3]. The final distribution involves applying the above process repeatedly over the remaining unset variables (i.e., $x_{\overline{S}}$) until all the coordinates are set, observing that for every $J \subseteq [n]$ and x_J, $f(., x_J)$ has sensitivity at most s. The subexponential seed-length is achieved by optimizing the parameters k and p from (1) while making sure that the overall error does not exceed ε.

Discussion

Our overall construction involves a combination of several samples from any k-wise independent distribution for an appropriate k. It is not clear whether simply one sample from a k-wise independent distribution suffices to fool low-sensitivity functions (recall that this is a consequence of the sensitivity conjecture with $k = \text{poly}(s)$). If this were true for all k-wise independent distributions, then via LP Duality (see the work of Bazzi [5]) we would get that every Boolean function f with sensitivity s has sandwiching real polynomials f_ℓ, f_u of degree k such that $\forall x : f_\ell(x) \leq f(x) \leq f_u(x)$ and $\mathbf{E}_x[f_u(x) - f_\ell(x)] \leq \epsilon$. We ask if a similar characterization can be obtained for the class of functions fooled by our construction.

2 Preliminaries

We denote by $[n] = \{1, \ldots, n\}$. We denote by $\mathcal{U}_n$ the uniform distribution over $\{-1,1\}^n$. We denote by log and ln the logarithms in bases 2 and e, respectively. For $f : \{-1,1\}^n \to \mathbb{R}$, we

denote by $\|f\|_p = \left(\mathbf{E}_{x\in\{-1,1\}^n}[|f(x)|^p]\right)^{1/p}$. For $x \in \{-1,1\}^n$, denote by $x \oplus e_i$ the vector obtained from x by changing the sign of x_i.

For a Boolean function $f : \{-1,1\}^n \to \{-1,1\}$, denote by $S(f,y)$, the set of sensitive coordinates of f at y, i.e.,

$$S(f,y) \triangleq \{i \in [n] : f(y) \neq f(y \oplus e_i)\} .$$

The sensitivity of f, denoted $s(f,x)$, is defined to be the number of sensitive coordinates of f, namely $s(f,x) = |S(f,x)|$. For example if $f(x_1,x_2,x_3) = x_1x_2$, then $s(f,111) = 2$ and $S(f,111) = \{1,2\}$. The sensitivity of a Boolean function f, denoted $s(f)$ is the maximum $s(f,x)$ over all choices of x.

2.1 Harper's Inequality and Simon's Theorem

▶ **Theorem 3** (Harper's Inequality). *Let $G = (V,E)$ be the n-dimensional hypercube, where $V = \{-1,1\}^n$. Let $A \subseteq V$ be a non-empty set. Then,*

$$\frac{|E(A,A^c)|}{|A|} \geq \log_2\left(\frac{2^n}{|A|}\right).$$

We will use the following simple corollary of Harper's inequality on multiple occasions. (This inequality was used in several previous works regarding the sensitivity conjecture, e.g. [15, 4].)

▶ **Corollary 4.** *Let $f : \{-1,1\}^n \to \{-1,1\}$ be a non-constant function with $s^1(f) \leq s$. Then, $|f^{-1}(1)| \geq 2^{n-s}$.*

Proof. Let $A = f^{-1}(1)$. Since f is non-constant, $|A| > 0$. By Harper's inequality the average sensitivity of f on A is at least $\log(2^n/|A|)$. However the average sensitivity of f on A is at most s, hence $\log(2^n/|A|) \leq s$, or equivalently, $|A| \geq 2^{n-s}$. ◀

We will also need the following result due to Simon [15].

▶ **Theorem 5** (Simon [15]). *For every Boolean function $f : \{-1,1\}^n \to \{-1,1\}$ we have*

$$s(f)4^{s(f)} \geq n',$$

where $n' \leq n$ is the number of variables on which f depends.

2.2 Restrictions

▶ **Definition 6** (Restriction). Let $f : \{-1,1\}^n \to \{-1,1\}$ be a Boolean function. A restriction is a pair (J,z) where $J \subseteq [n]$ and $z \in \{-1,1\}^{\bar{J}}$. We denote by $f_{J|z} : \{-1,1\}^n \to \{-1,1\}$ the function f restricted according to (J,z), defined by

$$f_{J|z}(x) = f(y), \quad \text{where} \quad y_i = \begin{cases} x_i, & i \in J \\ z_i, & \text{otherwise} \end{cases}.$$

▶ **Definition 7** (Random Valued Restriction). Let $n \in \mathbb{N}$. A random variable (J,z), distributed over restrictions of $\{-1,1\}^n$ is called random-valued if conditioned on J, the variable z is uniformly distributed over $\{-1,1\}^{\bar{J}}$.

▶ **Definition 8** ((k,p)-wise Random Selection). A random variable $J \subseteq [n]$ is said to be a (k,p)-wise random selection if the events $\{(1 \in J), (2 \in J), \ldots, (n \in J)\}$ are k-wise independent, and each one of them happens with probability p.

A (k,p)-wise independent restriction is a random-valued restriction in which J is chosen using a (k,p)-wise random selection.

2.3 Fourier Analysis of Boolean Functions

Any function $f : \{-1,1\}^n \to \mathbb{R}$ has a unique Fourier representation:

$$f(x) = \sum_{S \subseteq [n]} \hat{f}(S) \cdot \prod_{i \in S} x_i \ ,$$

where the coefficients $\hat{f}(S) \in \mathbb{R}$ are given by $\hat{f}(S) = \mathbf{E}_x[f(x) \cdot \prod_{i \in S} x_i]$. Parseval's identity states that $\sum_S \hat{f}(S)^2 = \mathbf{E}_x[f(x)^2] = \|f\|_2^2$, and in the case that f is Boolean (i.e., $f : \{-1,1\}^n \to \{-1,1\}$), all are equal to 1. The Fourier representation is the unique multilinear polynomial which agrees with f on $\{-1,1\}^n$. We denoted by $\deg(f)$ the degree of this polynomial, which also equals $\max\{|S| : \hat{f}(S) \neq 0\}$. We denote by

$$\mathbf{W}^k[f] \triangleq \sum_{S \subseteq [n], |S|=k} \hat{f}(S)^2$$

the *Fourier weight at level k of f*. Similarly, we denote $\mathbf{W}^{\geq k}[f] \triangleq \sum_{S \subseteq [n], |S| \geq k} \hat{f}(S)^2$. For $k \in \mathbb{N}$ we denote the k-th Fourier moment of f by

$$\mathrm{Inf}^k[f] \triangleq \sum_{S \subseteq [n]} \hat{f}(S)^2 \cdot \binom{|S|}{k} = \sum_{d=1}^{n} \mathbf{W}^d[f] \cdot \binom{d}{k} \ .$$

We will use the following result of Gopalan et al. [8].

▶ **Theorem 9** ([8, Lemma 5.6]). *Let f be a Boolean function with sensitivity at most s. Then, for all k, $\mathrm{Inf}^k[f] \leq (32 \cdot s)^k$.*

For more about Fourier moments of Boolean functions see [16, 8]. The following fact relates the Fourier coefficients of f and $f_{J|z}$, where (J, z) is a random valued restriction.

▶ **Fact 10** (Proposition 4.17, [13]). *Let $f : \{-1,1\}^n \to \mathbb{R}$, let $S \subseteq [n]$, and let D be a distribution of random valued restrictions. Then,*

$$\mathop{\mathbf{E}}_{(J,z)\sim D}\left[\widehat{f_{J|z}}(S)\right] = \hat{f}(S) \cdot \mathop{\mathbf{Pr}}_{(J,z)\sim D}[S \subseteq J]$$

and

$$\mathop{\mathbf{E}}_{(J,z)\sim D}\left[\widehat{f_{J|z}}(S)^2\right] = \sum_{U \subseteq [n]} \hat{f}(U)^2 \cdot \mathop{\mathbf{Pr}}_{(J,z)\sim D}[J \cap U = S]$$

We include the proof of this fact for completeness.

Proof. Let $(J, z) \sim D$. Then, by definition of random valued restriction, given J we have that z is a random string in $\{-1,1\}^{\bar{J}}$. Fix J, and rewrite f's Fourier expansion by splitting the variables to $(J, \bar{J})$.

$$f(x) = \sum_{S \subseteq [n]} \hat{f}(S) \cdot \prod_{i \in S} x_i = \sum_{T \subseteq J} \prod_{i \in T} x_i \cdot \sum_{T' \subseteq \bar{J}} \hat{f}(T \cup T') \cdot \prod_{j \in T'} x_j$$

Hence,

$$f_{J,z}(x) = \sum_{T \subseteq J} \prod_{i \in T} x_i \cdot \sum_{T' \subseteq \bar{J}} \hat{f}(T \cup T') \cdot \prod_{j \in T'} z_j$$

So the S-Fourier coefficient of $f_{J,z}$ is 0 if $S \not\subseteq J$ and it is $\sum_{T' \subseteq \bar{J}} \hat{f}(S \cup T') \cdot \prod_{j \in T'} z_j$ otherwise. In other words,

$$\widehat{f_{J,z}}(S) = \mathbb{1}_{S \subseteq J} \cdot \sum_{T' \subseteq \bar{J}} \hat{f}(S \cup T') \cdot \prod_{j \in T'} z_j \ ,$$

and its expectation in z in the case $S \subseteq J$ is $\hat{f}(S)$. As for the second moment,

$$\begin{aligned} \mathop{\mathbf{E}}_{J,z}[\widehat{f_{J,z}}(S)^2] &= \mathop{\mathbf{E}}_{J}[\mathop{\mathbf{E}}_{z}[\widehat{f_{J,z}}(S)^2]] = \mathop{\mathbf{E}}_{J}[\mathbb{1}_{S \subseteq J} \cdot \mathop{\mathbf{E}}_{z}[(\sum_{T' \subseteq \bar{J}} \hat{f}(S \cup T') \prod_{j \in T'} z_j)^2]] \\ &= \mathop{\mathbf{E}}_{J}[\mathbb{1}_{S \subseteq J} \cdot \sum_{T' \subseteq \bar{J}} \hat{f}(T \cup T')^2] = \sum_{U \subseteq [n]} \hat{f}(U)^2 \cdot \mathbf{Pr}[J \cap U = S] \ . \end{aligned}$$

◀

3 PRGs for Low-Sensitivity Functions

In this section we prove our main theorem.

▶ **Theorem 1.** *There is a distribution $\mathcal{D}$ on $\{-1,1\}^n$ with seed-length $2^{O(\sqrt{s+\log(1/\varepsilon)})} \cdot \log(n)$ that ε-fools every $f : \{-1,1\}^n \to \{-1,1\}$ with $s(f) = s$.*

Our main tool will be the following theorem stating that under k-wise independent random restrictions every low-sensitivity function becomes a junta with high probability. We postpone the proof of Theorem 11 to Section 4.

▶ **Theorem 11.** *Let $f : \{-1,1\}^n \to \{-1,1\}$ with $s(f) = s$. Let $1 \le k \le s/10$, and let $\mathcal{D}$ be a distribution of (k,p)-wise independent restrictions. Then,*

$$\mathop{\mathbf{Pr}}_{(J,z)\sim\mathcal{D}}[f_{J|z} \text{ is not a } (2^{20k})\text{-junta}] \le O(ps)^k \cdot 2^{6s}$$

Theorem 11 allows us to employ the framework of Trevisan and Xue [17] who used a derandomized switching lemma to construct pseudorandom generators for AC0 circuits. In what follows we will make the following choices of parameters

i. $k := O(\sqrt{s + \log(1/\varepsilon)})$.
ii. $p := 2^{-k}/s = 2^{-O(\sqrt{s+\log(1/\varepsilon)})}$
iii. $m := O(p^{-1} \cdot \log(s \cdot 4^s/\varepsilon)) = 2^{O(\sqrt{s+\log(1/\varepsilon)})}$

We select a sequence of disjoint sets $J_1, ..., J_m$ as follows. We pick $J_i \subseteq [n] \backslash (J_1 \cup \cdots \cup J_{i-1})$ by letting $J_i := K_i \backslash (J_1 \cup \cdots \cup J_{i-1})$ where $K_i \subseteq [n]$ is drawn from a (p,k)-wise random selection. For each i, we pick $x_{J_i} \in \{-1,1\}^{|J_i|}$ according to an $\frac{\varepsilon}{4m}$-almost 2^{20k}-wise independent distribution. Finally, we will fix $x_i := 0$ for any $i \in [n] \backslash (J_1 \cup \cdots \cup J_m)$.

To account for the seed-length:

- By a construction of [2] each K_i can be selected using $O(k \cdot \log n)$ random bits, and
- By constructions of [3] each $x_{J_i} \in \{-1,1\}^{|J_i|}$ can be selected using $O(2^{20k} + \log\log(n) + \log(1/\varepsilon))$ random bits.

Thus, the total seed-length is

$$O\left(m \cdot \left(2^{20k} + \log\log(n) + \log(1/\varepsilon) + k \cdot \log(n)\right)\right) \le 2^{O(\sqrt{s+\log(1/\varepsilon)})} \cdot \log(n) \ .$$

To conclude the proof, we show that the above distribution fools sensitivity s Boolean functions. Denote by $\mathcal{D}$ the distribution described above, and suppose $f : \{-1,1\}^n \to \{-1,1\}$ satisfies $s(f) = s$. We first note that by Theorem 5, f depends on at most $s \cdot 4^s$ variables,

denote this set S, so that $|S| \leq s \cdot 4^s$. By our choice of m, with probability at least $1 - \frac{\varepsilon}{2}$, $S \subseteq J_1 \cup \cdots \cup J_m$.

We use x to denote a vector drawn from $\mathcal{D}$ and y to denote a vector drawn according to the uniform distribution over $\{-1,1\}^n$. Moreover, for every $i = 0, 1, \ldots, m$, we let $z_i := (x_{J_1}, ..., x_{J_i}, y_{[n]\setminus(J_1 \cup \ldots J_i)})$. Note that $z_0 = y$. We first prove that for every $i = 0, 1, \ldots, m-1$,

$$\left| \mathop{\mathbf{E}}_{x\sim\mathcal{D}, y\sim\mathcal{U}} f(z_i) - \mathop{\mathbf{E}}_{x\sim\mathcal{D}, y\sim\mathcal{U}} f(z_{i+1}) \right| \leq \frac{\varepsilon}{2m}. \tag{2}$$

This holds since by Theorem 11, for every fixed choice of $J_1, \ldots, J_i$ and $x_{J_1}, \ldots x_{J_i}$, we have

$$\mathop{\mathbf{Pr}}_{J_{i+1}, y\sim\mathcal{U}} \left[f(x_{J_1}, \ldots, x_{J_i}, \cdot, y_{[n]\setminus(J_1 \cup \ldots J_{i+1})}) \text{ is not a } 2^{20k}\text{-junta} \right] \leq O(ps)^k \cdot 2^{6s} \leq \frac{\varepsilon}{4m},$$

and that every 2^{20k}-junta is $\varepsilon/4m$-fooled by any $\varepsilon/4m$-almost 2^{20k}-wise independent distribution. By triangle inequality and summing up (2) for all i we get

$$\left| \mathop{\mathbf{E}}_{y\sim\mathcal{U}} f(y) - \mathop{\mathbf{E}}_{x\sim\mathcal{D}, y\sim\mathcal{U}} f(z_m) \right| \leq \sum_{i=0}^{m-1} \left| \mathop{\mathbf{E}}_{x\sim\mathcal{D}, y\sim\mathcal{U}} f(z_i) - \mathop{\mathbf{E}}_{x\sim\mathcal{U}, y\sim\mathcal{D}} f(z_{i+1}) \right| \leq \frac{\varepsilon}{2}. \tag{3}$$

To finish the proof of Theorem 1, note that with probability at least $1-\varepsilon/2$, $f(x_{J_1}, \ldots, x_{J_m}, \cdot)$ is a constant function (which follows from $S \subseteq J_1 \cup \cdots \cup J_m$), and thus $|\mathbf{E}_{x,y} f(z_m) - \mathbf{E}_x f(x)| \leq \varepsilon/2$. Combining this with Eq. (3) gives $|\mathbf{E}_{y\sim\mathcal{U}} f(y) - \mathbf{E}_{x\sim\mathcal{D}} f(x)| \leq \varepsilon/2 + \varepsilon/2$.

4 Measures of Boolean Functions under k-Wise Independent Random Restrictions

▶ **Lemma 12.** *Let $f : \{-1,1\}^n \to \{-1,1\}$. Let $\mathcal{D}$ be a distribution of (k,p)-wise independent restrictions. Then,*

$$\mathop{\mathbf{E}}_{(J,z)\sim D} [\mathbf{W}^{\geq k}[f_{J|z}]] \leq p^k \cdot \mathrm{Inf}^k[f]. \tag{4}$$

Proof. Using Fact 10, we have

$$\mathop{\mathbf{E}}_{J,z} [\mathbf{W}^{\geq k}[f|_{J,z}]] = \sum_{U\subseteq[n]} \hat{f}(U)^2 \cdot \mathop{\mathbf{Pr}}_{J} [|U \cap J| \geq k]$$

Fix U. Let us upper bound $\mathbf{Pr}_J[|U \cap J| \geq k]$. It is at most $\binom{|U|}{k} \cdot p^k$ by taking a union bound over all $\binom{|U|}{k}$ subsets S of size k of U and observing that $\mathbf{Pr}_J[S \subseteq J] = p^k$ by the fact that J is a (k,p)-wise random selection. We thus have

$$\mathop{\mathbf{E}}_{J,z} [\mathbf{W}^{\geq k}[f|_{J,z}]] \leq \sum_{U\subseteq[n]} \hat{f}(U)^2 \cdot \binom{|U|}{k} \cdot p^k = \mathrm{Inf}^k[f] \cdot p^k. \quad \blacktriangleleft$$

Very analogously, we have the following statement with respect to sensitivity moments.

▶ **Lemma 13.** *Let $f : \{-1,1\}^n \to \{-1,1\}$. Let $\mathcal{D}$ be a distribution of (k,p)-wise independent restrictions. Then,*

$$\mathop{\mathbf{E}}_{(J,z)\sim\mathcal{D}} \left[\mathop{\mathbf{Pr}}_{x} [s(f_{J|z}, x) \geq k] \right] \leq p^k \cdot \mathop{\mathbf{E}}_{x\in\{-1,1\}^n} \left[\binom{s(f,x)}{k} \right].$$

Proof. We expand $\mathbf{E}_{(J,z)\sim\mathcal{D}}\left[\mathbf{Pr}_x[s(f_{J|z},x)\geq k]\right]$:

$$\begin{aligned}
\mathop{\mathbf{E}}_{J,z}\left[\mathop{\mathbf{Pr}}_{x}[s(f_{J|z},x)\geq k]\right] &= \mathop{\mathbf{E}}_{J}\mathop{\mathbf{E}}_{z\in\{-1,1\}^{\bar{J}}}\mathop{\mathbf{E}}_{x\in\{-1,1\}^n}\left[\mathbb{1}_{\{s(f(z,.),x_J)\geq k\}}\right]\\
&= \mathop{\mathbf{E}}_{J}\mathop{\mathbf{E}}_{z\in\{-1,1\}^{\bar{J}}}\mathop{\mathbf{E}}_{x_J\in\{-1,1\}^{J}}\left[\mathbb{1}_{\{s(f(z,.),x_J)\geq k\}}\right]\\
&= \mathop{\mathbf{E}}_{J}\mathop{\mathbf{E}}_{y\in\{-1,1\}^n}\left[\mathbb{1}_{\{s(f(y_{\bar{J}},.),y_J)\geq k\}}\right]\\
&= \mathop{\mathbf{E}}_{y\in\{-1,1\}^n}\left[\mathop{\mathbf{E}}_{J}\left[\mathbb{1}_{\{s(f(y_{\bar{J}},.),y_J)\geq k\}}\right]\right]\\
&= \mathop{\mathbf{E}}_{y\in\{-1,1\}^n}\left[\mathop{\mathbf{Pr}}_{J}[|J\cap S(f,y)|\geq k]\right]\\
&\leq \mathop{\mathbf{E}}_{y\in\{-1,1\}^n}\left[\binom{s(f,y)}{k}\cdot p^k\right]
\end{aligned}$$

where the last inequality is due to the following observation. We observe that for a given y and a set $S=\{i_1,...,i_k\}$ of k sensitive directions of f at y, the probability that $S\subseteq J$ is p^k. We then union-bound over all subsets S of cardinality k of $S(f,y)$. ◀

We are now ready to prove the main theorem of this section (restated next).

▶ **Theorem 11.** *Let $f:\{-1,1\}^n\to\{-1,1\}$ with $s(f)=s$. Let $1\leq k\leq s/10$, and let $\mathcal{D}$ be a distribution of (k,p)-wise independent restrictions. Then,*

$$\mathop{\mathbf{Pr}}_{(J,z)\sim\mathcal{D}}[f_{J|z} \text{ is not a } (2^{20k})\text{-junta}]\leq O(ps)^k\cdot 2^{6s}$$

Proof. We upper and lower bound the value of

$$(*)=\mathop{\mathbf{E}}_{(J,z)\sim\mathcal{D}}\left[\mathbf{W}^{\geq k}[f_{J|z}]+\mathop{\mathbf{Pr}}_{x}[s(f_{J|z},x)\geq k]\right].$$

For the upper bound we use Lemma 13 to get

$$\mathop{\mathbf{E}}_{(J,z)\sim\mathcal{D}}\left[\mathop{\mathbf{Pr}}_{x}[s(f_{J|z},x)\geq k]\right]\leq (ps)^k,$$

and Lemma 12 and Theorem 9 to get

$$\mathop{\mathbf{E}}_{(J,z)\sim\mathcal{D}}\left[\mathbf{W}^{\geq k}[f_{J|z}]\right]\leq O(ps)^k,$$

which gives $(*)\leq O(ps)^k$.

For the lower bound we use the following lemma, the proof of which we defer to Section 5.

▶ **Lemma 14.** *Let $f:\{-1,1\}^n\to\{-1,1\}$ with $s(f)\leq s$. Let $1\leq k\leq s/10$. Assume $\mathbf{W}^{\geq k}[f]\leq 2^{-6s}$, and that at most 2^{-6s} fraction of the points in $\{-1,1\}^n$ have sensitivity at least k. Then, f is a 2^{20k}-junta.*

Let $\mathcal{E}$ be the event that $f_{J|z}$ is not a 2^{20k}-junta. Whenever $\mathcal{E}$ occurs, Lemma 2 implies that either $\mathbf{Pr}_x[s(f_{J|z},x)\geq k]\geq 2^{-6s}$ or $\mathbf{W}^{\geq k}[f_{J|z}]\geq 2^{-6s}$. In both cases, $\mathbf{Pr}_x[s(f_{J|z},x)\geq k]+\mathbf{W}^{\geq k}[f_{J|z}]\geq 2^{-6s}$. Thus, we get the lower bound

$$(*)\geq \mathbf{Pr}[\mathcal{E}]\cdot\mathop{\mathbf{E}}_{(J,z)}\left[\mathbf{W}^{\geq k}[f_{J|z}]+\mathop{\mathbf{Pr}}_{x}[s(f_{J|z},x)\geq k]\mid\mathcal{E}\right]\geq\mathbf{Pr}[\mathcal{E}]\cdot 2^{-6s}$$

Comparing the upper and lower bound gives

$$\mathop{\mathbf{Pr}}_{(J,z)\sim\mathcal{D}}[f_{J|z} \text{ is not a } (2^{20k})\text{-junta}]=\mathbf{Pr}[\mathcal{E}]\leq 2^{6s}\cdot(*)\leq 2^{6s}\cdot O(ps)^k\,.$$ ◀

5 A Strengthening of Friedgut's Theorem for Low-Sensitivity Functions

▶ **Theorem 15** (Friedgut's Junta Theorem - [13, Thm 9.28]). *Let $f : \{-1,1\}^n \to \{-1,1\}$. Let $0 < \varepsilon \leq 1$ and $k \geq 0$. If $\mathbf{W}^{>k}[f] \leq \varepsilon$, then f is 2ε-close to a $(9^k \cdot \mathrm{Inf}[f]^3/\varepsilon^2)$-junta.*

▶ **Lemma 16.** *Let $f : \{-1,1\}^n \to \{-1,1\}$ with $s(f) \leq s$. Let $1 \leq k \leq s/10$. Assume $\mathbf{W}^{\geq k}[f] \leq 2^{-6s}$, and that at most 2^{-6s} fraction of the points in $\{-1,1\}^n$ have sensitivity at least k. Then, f is a 2^{20k}-junta.*

Proof. We first show that $\mathrm{Inf}[f] \leq k$. By Theorem 5, f depends on at most $4^s \cdot s$ variables[2]. Thus, $\mathrm{Inf}[f] \leq (k-1) + \mathbf{W}^{\geq k}[f] \cdot (4^s \cdot s) \leq (k-1) + 1 = k$. Apply Friedgut's theorem with $\varepsilon = 2^{-6k-1} \geq \mathbf{W}^{\geq k}[f]$. We get a K-junta h, for

$$K = 9^k \cdot \mathrm{Inf}[f]^3/\varepsilon^2 \leq 9^k \cdot k^3 \cdot 2^{12k+2} < 2^{20k},$$

that $2\varepsilon = 2^{-6k}$ approximates f. Let $C_1, \ldots, C_N$ be the subcubes corresponding to the $N = 2^K$ different assignments to the junta variables. Without loss of generality, under each C_i, h attains the constant value that is the majority-vote of f on C_i. In other words, f and h agree on at least $1/2$ of the points in each subcube C_i.

Let $p_i = |\{x \in C_i : f(x) \neq h(x)\}|/|C_i|$, for $i \in [N]$. By the above discussion, $0 \leq p_i \leq 1/2$. In addition, since $f|_{C_i}$ has sensitivity at most s, if $p_i > 0$, then $p_i \geq 2^{-s}$ using Corollary 4.

Assume towards contradiction that $h \neq f$. We will think of the hamming cube $\{-1,1\}^n$ as an outer cube of dimension K, and an inner cube of dimension $n-K$. Each subcube C_i is an instance of the inner cube $\{-1,1\}^{n-K}$. The graph of subcubes is an instance of the outer cube $\{-1,1\}^K$. Call a subcube C_i:
decisive if $p_i = 0$,
confused if $2^{-s} \leq p_i < 2^{-k-1}$, or
indecisive if $p_i \geq 2^{-k-1}$.
Denote by α, β, γ the fraction of decisive, confused and indecisive subcubes correspondingly.

Since we assumed (towards contradiction) that $h \neq f$, at least one subcube is confused or indecisive. Consider the graph G of subcubes, which is isomorphic to $\{-1,1\}^K$, in which each vertex represents either a decisive, confused or indecisive subcube, and two vertices are adjacent if and only if their corresponding subcubes are adjacent in $\{-1,1\}^n$. First, we show that at least 2^{-2s} fraction of the subcubes are confused or indecisive. Assume otherwise, then by Harper's inequality (Thm. 3) there is a confused or indecisive cube C_i with at least $2s+1$ decisive subcubes as neighbors. As there are points with both $\{-1,1\}$ values in C_i, we may pick a point $x \in C_i$ whose value is the opposite of the majority of the decisive neighbor subcubes of C_i, which gives $s(f,x) \geq s+1$, a contradiction. We thus have

$$\beta + \gamma \geq 2^{-2s} \tag{5}$$

Next, we show that β is very small and in particular much smaller than γ. Towards this end, we shall analyze the sensitivity within confused subcubes. If C_i is confused (i.e., $2^{-s} \leq p_i < 2^{-k-1}$), then by Harper's inequality (inside C_i) the average sensitivity on the minority of $f|_{C_i}$ is greater than $k+1$. Since sensitivity ranges between 0 to s, at least $1/s$ of the points with minority value in $f|_{C_i}$ have sensitivity at least k (otherwise the average

[2] Note that our final goal will be to show that f actually depends on 2^{20k} variables, and that k can be significantly smaller than s.

sensitivity among them will be less than $(1/s) \cdot s + k \leq k + 1$). As there are at least 2^{-s} points with the minority value on the subcube C_i, we get that at least $2^{-s}/s \geq 2^{-2s}$ fraction of the points in C_i have sensitivity at least k.

If the fraction of confused subcubes is more than $2^{-2s}/(K+1)$, then more than $2^{-4s}/(K+1) \geq 2^{-6s}$ fraction of the points in $\{-1,1\}^n$ has sensitivity at least k, which contradicts one of the assumptions. Thus,

$$\beta \leq 2^{-2s}/(K+1). \tag{6}$$

Furthermore, combining Eq. (5) and (6), we have that the fraction of indecisive subcubes, γ, is at least

$$\gamma \geq 2^{-2s} \cdot \frac{K}{K+1} \geq K \cdot \beta. \tag{7}$$

Consider again the graph G of subcubes (which is isomorphic to $\{-1,1\}^K$). Recall that each vertex in the graph G corresponds to a subcube which is either decisive, confused or indecisive. Call A the set of vertices that correspond to indecisive subcubes. Then, $|A| = \gamma \cdot 2^K$. By the fact that h approximates f with error at most 2^{-6k}, the size of A is at most $2^{-6k} \cdot 2^{k+1} \cdot 2^K \leq 2^{-4k} \cdot 2^K$, i.e., $\gamma \leq 2^{-4k}$. By Harper's inequality, $|E(A, \overline{A})| \geq |A| \cdot (4k)$. There are at most $\beta \cdot 2^K \cdot K \leq \gamma \cdot 2^K = |A|$ edges touching confused nodes, hence there are at least $|A| \cdot (4k-1)$ edges from A to decisive nodes. As before, the maximal number of edges from a node in A to decisive nodes is at most $2s$, otherwise we get a contradiction to $s(f) \leq s$. This implies that at least $1/2s$ fraction of the nodes in A have at least $4k-2$ edges to decisive subcubes. For each indecisive subcube C_i with at least $4k-2$ edges to decisive subcubes, let $b \in \{-1,1\}$ be the majority-vote among these decisive subcubes. All points with value $-b$ in C_i have sensitivity at least $(4k-2)/2 \geq 2k-1 \geq k$, and the fraction of such points in C_i is at least 2^{-k-1}. Using Eq. (7) we get that

$$\gamma \cdot \frac{1}{2s} \cdot 2^{-k-1} \geq 2^{-2s} \cdot \frac{K}{K+1} \cdot \frac{1}{2s} \cdot 2^{-k-1} \geq 2^{-6s}$$

of the points in $\{-1,1\}^n$ have sensitivity at least k, which yields a contradiction. ◀

References

1. M. Ajtai and A. Wigderson. Deterministic simulation of probabilistic constant depth circuits (preliminary version). In *FOCS*, pages 11–19, 1985.
2. N. Alon, L. Babai, and A. Itai. A fast and simple randomized parallel algorithm for the maximal independent set problem. *Journal of algorithms*, 7(4):567–583, 1986.
3. N. Alon, O. Goldreich, J. Håstad, and R. Peralta. Simple construction of almost k-wise independent random variables. *Random Structures and Algorithms*, 3(3):289–304, 1992.
4. Andris Ambainis, Mohammad Bavarian, Yihan Gao, Jieming Mao, Xiaoming Sun, and Song Zuo. Tighter relations between sensitivity and other complexity measures. In Javier Esparza, Pierre Fraigniaud, Thore Husfeldt, and Elias Koutsoupias, editors, *Automata, Languages, and Programming - 41st International Colloquium, ICALP 2014, Copenhagen, Denmark, July 8-11, 2014, Proceedings, Part I*, volume 8572 of *Lecture Notes in Computer Science*, pages 101–113. Springer, 2014. doi:10.1007/978-3-662-43948-7_9.
5. Louay M. J. Bazzi. Polylogarithmic independence can fool DNF formulas. *SIAM J. Comput.*, 38(6):2220–2272, 2009. doi:10.1137/070691954.
6. M. Braverman. Polylogarithmic independence fools AC^0 circuits. *J. ACM*, 57(5):28:1–28:10, 2010.

7 P. Gopalan, N. Nisan, R. A. Servedio, K. Talwar, and A. Wigderson. Smooth boolean functions are easy: Efficient algorithms for low-sensitivity functions. In *ITCS*, pages 59–70, 2016.

8 Parikshit Gopalan, Rocco A. Servedio, Avishay Tal, and Avi Wigderson. Degree and sensitivity: tails of two distributions. *Electronic Colloquium on Computational Complexity (ECCC)*, 23:69, 2016. URL: `http://eccc.hpi-web.de/report/2016/069`.

9 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`.

10 N. Nisan. Pseudorandom generators for space-bounded computation. *Combinatorica*, 12(4):449–461, 1992.

11 N. Nisan and M. Szegedy. On the degree of Boolean functions as real polynomials. *Computational Complexity*, 4:301–313, 1994.

12 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`.

13 R. O'Donnell. *Analysis of boolean functions*. Cambridge University Press, 2014.

14 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.

15 H. U. Simon. A tight $\Omega(\log\log n)$-bound on the time for parallel RAM's to compute nondegenerated boolean functions. In *Foundations of computation theory*, pages 439–444. Springer, 1983.

16 Avishay Tal. Tight bounds on the fourier spectrum of AC^0. *Electronic Colloquium on Computational Complexity (ECCC)*, 21:174, 2014. URL: `http://eccc.hpi-web.de/report/2014/174`.

17 Luca Trevisan and Tongke Xue. A derandomized switching lemma and an improved derandomization of AC0. In *Proceedings of the 28th Conference on Computational Complexity, CCC 2013, K.lo Alto, California, USA, 5-7 June, 2013*, pages 242–247. IEEE Computer Society, 2013. `doi:10.1109/CCC.2013.32`.

A Does the NW-Generator Fool Low-Sensitivity Functions?

In this section we recall the construction and analysis of the NW-Generator [12]. For ease of notation, we treat Boolean functions here as $f : \{0,1\}^n \to \{0,1\}$. Suppose we want to construct a pseudorandom generator fooling a class of Boolean functions $\mathcal{C}$. Nisan and Wigderson provide a generic way to construct such PRGs based on the premise that there is some explicit function f which is average-case hard for a class $\mathcal{C}'$ that slightly extends $\mathcal{C}$. Recall that $\mathsf{Sens}(s)$ is the class of all Boolean functions with sensitivity at most s. In the case $\mathcal{C} = \mathsf{Sens}(s)$, the argument may fail, because $\mathcal{C}'$ is not provably similar to $\mathcal{C}$. The difficulty comes from the fact that low-sensitivity functions are not closed under projections as will be explained later.

Let $f : \{0,1\}^\ell \to \{0,1\}$ be a function that is average-case hard for class $\mathcal{C}$. Let $S_1, \ldots, S_n \subseteq [r]$ be a design over a universe of size r where $|S_i| = \ell$, and $|S_i \cap S_j| \leq \alpha$ for all $i \neq j \in [n]$ (think of α as much smaller than ℓ). The NW-generator $G_f : \{0,1\}^r \to \{0,1\}^n$ is defined as

$$G_f(x_1, \ldots, x_r) = (f(x_{S_1}), f(x_{S_2}), \ldots, f(x_{S_n}))$$

where x_{S_i} is the restriction of x to the coordinates in S_i, for any set $S_i \subseteq [n]$.

The proof that the NW-generator fools $\mathcal{C}$ goes via a contrapositive argument. We assume that there is a distinguisher $c \in \mathcal{C}$ such that

$$\left| \mathop{\mathbf{E}}_{z \in_R \{0,1\}^r}[c(G_f(z))] - \mathop{\mathbf{E}}_{x \in_R \{0,1\}^n}[c(x)] \right| \geq \varepsilon \ ,$$

and prove that f can be computed on more than $1/2 + \Omega(\varepsilon)/n$ fraction of the inputs by some function c'' which is not much more complicated than c. First, by Yao's next-bit predictor lemma, there exists an $i \in [n]$ and constants $a_i, \ldots, a_n, b \in \{0,1\}$ such that

$$\mathop{\mathbf{Pr}}_{x \in \{0,1\}^r}[c\left(f(x_{S_1}), f(x_{S_2}), \ldots, f(x_{S_{i-1}}), a_i, \ldots, a_n\right) \oplus b = f(x_{S_i})] \geq \frac{1}{2} + \frac{\Omega(\varepsilon)}{n} \ .$$

Since the class of function with sensitivity s is closed under restrictions (i.e., fixing the input variables to constant values) and negations we have that $c'(z_1, \ldots, z_{i-1}) := c(z_1, \ldots, z_{i-1}, a_i, \ldots, a_n) \oplus b$ is of sensitivity at most s. We get

$$\mathop{\mathbf{Pr}}_{x \in \{0,1\}^r}[c'(f(x_{S_1}), f(x_{S_2}), \ldots, f(x_{S_{i-1}})) = f(x_{S_i})] \geq \frac{1}{2} + \frac{\Omega(\varepsilon)}{n} \ .$$

Next, we wish to fix all values in $[r] \setminus S_i$. By averaging there exists an assignment y to the variables in $[r] \setminus S_i$ such that

$$\mathop{\mathbf{Pr}}_{x \in \{0,1\}^{S_i}}[c'(f((x \circ y)_{S_1}), f((x \circ y)_{S_2}), \ldots, f((x \circ y)_{S_{i-1}})) = f(x_{S_i})] \geq \frac{1}{2} + \frac{\Omega(\varepsilon)}{n} \ .$$

Note that for $j = 1, \ldots, i-1$, the value of $f((x \circ y)_{S_j})$ depends only on the variables in $S_j \cap S_i$ and there aren't too many such variables (at most α). The next step is to consider $c'' : \{0,1\}^{S_i} \to \{0,1\}$, defined by $c''(x) = c'(f((x \circ y)_{S_1}), f((x \circ y)_{S_2}), \ldots, f((x \circ y)_{S_{i-1}}))$, that have agreement at least $1/2 + \Omega(\varepsilon)/n$ with $f(x_{S_i})$. If c'' is a "simple" function then we get a contradiction as f is average-case hard.

It seems that c'' is simple, since it is the composition of c' with α-juntas. However, the point that we want to make is that even if c' is low-sensitivity and even if $\alpha = 1$, we are not guaranteed that c'' is of low-sensitivity.

To see this, suppose that $\alpha = 1$, i.e., all $|S_j \cap S_i| \leq 1$ for $j < i$. This means that as a function of x, each $f((x \circ y)_{S_j})$ depends on at most one variable, i.e., $f((x \circ y)_{S_j}) = a_j \cdot x_{k_j} \oplus b_j$ for some index $k_j \in S_i$ and some constants $a_j, b_j \in \{0,1\}$. We get that

$$c''(x) = c'(a_1 \cdot x_{k_1} \oplus b_1, a_2 \cdot x_{k_2} \oplus b_2, \ldots, a_2 \cdot x_{k_{i-1}} \oplus b_{i-1}).$$

Next, we argue that c'' could potentially have very high sensitivity. To see that, observe that flipping one bit x_i in the input to c'' results in changing a block of variables in the input to c', as there may be several j for which $k_j = i$. In the worst-case scenario, the sensitivity of c'' could be as big as the block sensitivity of c'. However, the best known bound is only $bs(f) \leq 2^{s(f) \cdot (1+o(1))}$ for any Boolean function f [4]. This means that we can only guarantee that $s(c'') \leq bs(c') \leq 2^{s \cdot (1+o(1))}$, and we do not have average-case hardness for such high-sensitivity functions.

▶ Remark. The above argument shows that the standard analysis of the Nisan-Wigderson generator applied to low-sensitivity Boolean functions breaks, but it does not mean that the generator does not ultimately fool $\mathsf{Sens}(s)$. Indeed, assuming the sensitivity conjecture, the argument will follow through.

Acknowledgements. We would like to thank Li-Yang Tan for bringing the problem to our attention and for stimulating and helpful discussions. We also thank the anonymous referee who pointed out a better PRG under the sensitivity conjecture using the decision tree complexity as opposed to the degree as used in a previous version of the paper.

Scheduling with Explorable Uncertainty*†

Christoph Dürr[1], Thomas Erlebach‡[2], Nicole Megow[3], and Julie Meißner[4]

1 **Sorbonne Universités, UPMC Univ Paris 06, CNRS, LIP6, Paris, France**
christoph.durr@lip6.fr
2 **Department of Informatics, University of Leicester, Leicester, UK**
te17@leicester.ac.uk
3 **Department of Mathematics and Computer Science, University of Bremen, Bremen, Germany**
nicole.megow@uni-bremen.de
4 **Institute of Mathematics, Technical University of Berlin, Berlin, Germany**
jmeiss@math.tu-berlin.de

Abstract

We introduce a novel model for scheduling with explorable uncertainty. In this model, the processing time of a job can potentially be reduced (by an *a priori* unknown amount) by testing the job. Testing a job j takes one unit of time and may reduce its processing time from the given upper limit $\bar{p}_j$ (which is the time taken to execute the job if it is not tested) to any value between 0 and $\bar{p}_j$. This setting is motivated e.g. by applications where a code optimizer can be run on a job before executing it. We consider the objective of minimizing the sum of completion times on a single machine. All jobs are available from the start, but the reduction in their processing times as a result of testing is unknown, making this an online problem that is amenable to competitive analysis. The need to balance the time spent on tests and the time spent on job executions adds a novel flavor to the problem. We give the first and nearly tight lower and upper bounds on the competitive ratio for deterministic and randomized algorithms. We also show that minimizing the makespan is a considerably easier problem for which we give optimal deterministic and randomized online algorithms.

1998 ACM Subject Classification F.2.2 Nonnumerical Algorithms and Problems – Sequencing and scheduling, F.1.2 Modes of Computation – Online computation, G.3 Probability and Statistics – Probabilistic algorithms (including Monte Carlo)

Keywords and phrases online scheduling, explorable uncertainty, competitive ratio, makespan, sum of completion times

Digital Object Identifier 10.4230/LIPIcs.ITCS.2018.30

1 Introduction

Uncertainty in scheduling has been modeled and investigated in many different ways, particularly in the frameworks of online optimization, stochastic optimization, and robust optimization. All these different approaches have the common assumption that the uncertain information, e.g., the processing time of a job, cannot be explored before making scheduling

* This research was carried out in the framework of MATHEON supported by Einstein Foundation Berlin, the German Science Foundation (DFG) under contract ME 3825/1 and the Bayerisch-Französisches Hochschulzentrum (BFHZ).
† A full version of this paper is available at [2], https://arxiv.org/abs/1709.02592
‡ The second author was supported by a study leave granted by University of Leicester.

decisions. However, in many applications there is the opportunity to gain exact or more precise information at a certain additional cost, e.g., by investing time, money, or bandwidth.

In this paper, we introduce a novel model for scheduling with explorable uncertainty. Given a set of n jobs, every job j can optionally be *tested* prior to its execution. A job that is executed without testing has processing time $\bar{p}_j \in \mathbb{Q}^+$, while a tested job has processing time p_j with $0 \leq p_j \leq \bar{p}_j$. Testing a job takes one unit of time on the same resource (machine) that processes jobs. Initially the algorithm knows for each job j only the upper limit $\bar{p}_j$, and gets to know the time p_j only after a test. Tested jobs can be executed at any time after their test. An algorithm must carefully balance testing and execution of jobs by evaluating the benefit and cost for testing.

We focus on scheduling on a single machine. Unless otherwise noted, we consider the sum of completion times as the minimization objective. We use competitive analysis to assess the performance of algorithms.

For the standard version of this single-machine scheduling problem, i.e., without testing, it is well known that the Shortest Processing Time (SPT) rule is optimal for minimizing the sum of completion times. The addition of testing, combined with the fact that the processing times p_j are initially unknown to the algorithm, turns the problem into an online problem with a novel flavor. An algorithm must decide which jobs to execute untested and which jobs to test. Once a job has been tested, the algorithm must decide whether to execute it immediately or to defer its execution while testing or executing other jobs. At any point in the schedule, it may be difficult to choose between testing a job (which might reveal that it has a very short processing time and hence is ideally suited for immediate execution) and executing an untested or previously tested job. Testing a job yields information that may be useful for the scheduler, but may delay the completion times of many jobs. Finding the right balance between tests and executions poses an interesting challenge.

If the processing times p_j that jobs have after testing are known, an optimal schedule is easy to determine: Testing and executing job j takes time $1+p_j$, so it is beneficial to test the job only if $1+p_j < \bar{p}_j$. In the optimal schedule, jobs are therefore ordered by non-decreasing $\min\{1+p_j, \bar{p}_j\}$. In this order, the jobs with $1+p_j < \bar{p}_j$ are tested and executed while jobs with $1+p_j \geq \bar{p}_j$ are executed untested. (For jobs with $1+p_j = \bar{p}_j$ it does not matter whether the job is tested and executed, or executed untested.)

Motivation and applications. Scheduling with testing is motivated by a range of application settings where an operation that corresponds to a test can be applied to jobs before they are executed. We discuss some examples of such settings. First, consider the execution of computer programs on a processor. A test could correspond to a code optimizer that takes unit time to process the program and potentially reduces its running-time. The upper limit of a job describes the running-time of the program if the code optimizer is not executed.

As another application, consider the transmission of files over a network link. It is possible to run a compression algorithm that can reduce the size of a file by an *a priori* unknown amount. If a file is incompressible (e.g., if it is already compressed), its size cannot be reduced at all. Running the compression algorithm corresponds to a test.

In some systems, a job can be executed in two different modes, a *safe* mode and an *alternative* mode. The safe mode is always possible. The alternative mode may have a shorter processing time, but is not possible for every job. A test is necessary to determine whether the alternative mode is possible for a job and what the processing time in the alternative mode would be.

As a final application area, consider settings where a diagnosis can be carried out to determine the exact processing time of a job. For example, a fault diagnosis can determine the time needed for a repair job, or a medical diagnosis can determine the time needed for a consultation and treatment session with a patient. Assume that the resource that carries out the diagnosis is the same resource that executes the job (e.g., an engineer or a medical doctor), and that the resource must be allocated to a job for an uninterruptible period that is guaranteed to cover the actual time needed for the job. If the diagnosis takes unit time, we arrive at our problem of scheduling with testing.

Depending on the application under consideration, it may appear more natural to speak about 'compressing' or 'optimizing' rather than 'testing' a job, but for the sake of simplicity we use the term 'testing' throughout this paper.

In some applications, it may be appropriate to allow the time for testing a job to be different for different jobs (e.g., proportional to the upper limit of a job). We leave the consideration of such generalizations of the problem to future work.

Related work. Scheduling with testing can be viewed as a problem in the area of explorable (or queryable) uncertainty, where additional information about the input can be learned using a query operation (in our case, a test). The line of research on optimization with explorable uncertain data has been initiated by Kahan [8] in 1991. His work concerns selection problems with the goal of minimizing the number of queries that are necessary to find the optimal solution. Later, other problems studied in this uncertainty model include finding the k-th smallest value in a set of uncertainty intervals [8, 7, 5] (also with non-uniform query cost [5]), caching problems in distributed databases [12], computing a function value [9], and classical combinatorial optimization problems, such as shortest path [4], finding the median [5], the knapsack problem [6], and the MST problem [3, 11]. While most work aims for minimal query sets to guarantee exact optimal solutions, Olsten and Widom [12] initiate the study of trade-offs between the number of queries and the precision of the found solution. They are concerned with caching problems. Further work in this vein can be found in [9, 4, 5].

In all this previous work, the execution of queries is separate from the actual optimization problem being solved. In our case, the tests are executed by the same machine that runs the jobs. Hence, the tests are not considered separately, but they directly affect the objective value of the actual problem (by delaying the completion of other jobs while a job is being tested). Hence, instead of minimizing the number of tests needed until an optimal schedule can be computed (which would correspond to the standard approach in the work on explorable uncertainty discussed above), in our case the tests of jobs are part of the schedule, and we are interested in the sum of completion times as the single objective function.

Our adversarial model is inspired by (and draws motivation from) recent work on a stochastic model of scheduling with testing introduced in [10, 13]. They consider the problem of minimizing the weighted sum of completion times on one machine for jobs whose processing times and weights are random variables with a joint distribution, and are independent and identically distributed across jobs. In their model, testing a job does not make its processing time shorter, it only provides information for the scheduler (by revealing the exact weight and processing time for a job, whereas initially only the distribution is known). They present structural results about optimal policies and efficient optimal or near-optimal solutions based on dynamic programming.

Our contribution. A scheduling algorithm in the model of explorable uncertainty has to make two types of decisions: which jobs should be tested, and in what order should job executions and tests be scheduled. There is a subtle compromise to be found between

Table 1 New contributions for minimizing the sum of completion times. * holds asymptotically

competitive ratio	lower bound	upper bound	
deterministic algorithms	1.8546	2	THRESHOLD
randomized algorithms	1.6257	1.7453	RANDOM
det. alg. on uniform instances	1.8546	1.9338*	BEAT
det. alg. on extreme uniform instances	1.8546	1.8668	UTE
det. alg. on extreme uniform inst. with $\bar{p} \approx 1.9896$	1.8546	1.8552	UTE

investing time to test jobs and the benefit one can gain from these tests. We design scheduling algorithms that address this exploration-exploitation question in different ways and provide nearly tight bounds on the competitive ratio. In our analysis, we first show that worst-case instances have a particular structure that can be described by only a few parameters. This goes hand in hand with analyzing also the structure of both, an optimal and an algorithm's schedule. Then we express the total cost of both schedules as functions of these few parameters. It is noteworthy that, under the assumptions made, we typically characterize the *exact* worst-case ratio. Given the parameterized cost ratio, we analyze the worst-case parameter choice. This technical part involves second order analysis which can be performed with computer assistance. We use the algebraic solver Mathematica.

Our results are the following. For scheduling with testing on a single machine with the objective of minimizing the sum of completion times, we present a 2-competitive deterministic algorithm and prove that no deterministic algorithm can achieve competitive ratio less than 1.8546. We then present a 1.7453-competitive randomized algorithm, showing that randomization provably helps for this problem. We also give a lower bound of 1.626 on the best possible competitive ratio of any randomized algorithm. Both lower bounds hold even for instances with uniform upper limits where every processing time is either 0 or equal to the upper limit. We call such instances *extreme uniform instances*. For such instances we give a 1.8668-competitive algorithm. In the special case where the upper limit of all jobs is ≈ 1.9896, the value used in our deterministic lower bound construction, that algorithm is even 1.8552-competitive. For the case of uniform upper limits and arbitrary processing times, we give a deterministic 1.9338-competitive algorithm. An overview of these results is shown in Table 1. Finally, we also mention some results for the simpler problem of minimizing the makespan in scheduling with testing.

Omitted proofs can be found in a full version of the paper; see the preprint [2]. Computations performed with the support of Mathematica are provided as notebook- and pdf-files at a companion webpage.[1]

2 Preliminaries

Problem definition. The problem of scheduling with testing is defined as follows. We are given n jobs to be scheduled on a single machine. Each job j has an upper limit $\bar{p}_j$. It can either be executed untested (taking time $\bar{p}_j$), or be tested (taking time 1) and then executed at an arbitrary later time (taking time p_j, where $0 \leq p_j \leq \bar{p}_j$). Initially only $\bar{p}_j$ is known for each job, and p_j is only revealed after j is tested. The machine can either test or execute a job at any time. The completion time of job j is denoted by C_j. Unless noted otherwise, we consider the objective of minimizing the sum of completion times $\sum_j C_j$.

[1] `http://cslog.uni-bremen.de/nmegow/public/mathematica-SwT.zip`

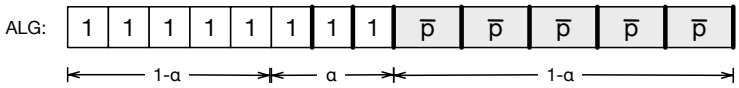


Figure 1 Typical schedule produced by an algorithm. Jobs are grey, tests are white. The completion time of a job is depicted by a thick bar. A job of length 0 completes immediately after testing.

Performance analysis. We compare the performance of an algorithm ALG to the optimal schedule using competitive analysis [1]. We denote by $\text{ALG}(I)$ the objective value (cost) of the schedule produced by ALG for an instance I, and by $\text{OPT}(I)$ the optimal cost. An algorithm ALG is *ρ-competitive* or *has competitive ratio at most ρ* if $\text{ALG}(I)/\text{OPT}(I) \leq \rho$ for all instances I of the problem. For randomized algorithms, $\text{ALG}(I)$ is replaced by $E[\text{ALG}(I)]$ in this definition.

When we analyze an algorithm or the optimal schedule, we will typically first argue that the schedule has a certain structure with different blocks of tests or job completions. Once we have established that structure, the cost of the schedule can be calculated by adding the cost for each block taken in isolation, plus the effect of the block on the completion times of later jobs. For example, assume that we have n jobs with upper limit $\bar{p}$, that αn of these jobs are *short*, with processing time 0, and $(1-\alpha)n$ jobs are *long*, with processing time $\bar{p}$. If an algorithm (in the worst case) first tests the $(1-\alpha)n$ long jobs, then tests the αn short jobs and executes them immediately, and finally executes the $(1-\alpha)n$ long jobs that were tested earlier (see also Figure 1), the total cost of the schedule can be calculated as

$$(1-\alpha)n^2 + \frac{\alpha n(\alpha n+1)}{2} + \alpha n(1-\alpha)n + \frac{(1-\alpha)n((1-\alpha)n+1)}{2}\bar{p}$$

where $(1-\alpha)n^2$ is the total delay that the $(1-\alpha)n$ tests of long jobs add to the completion times of all n jobs, $\frac{\alpha n(\alpha n+1)}{2}$ is the sum of completion times of a block with αn short jobs that are tested and executed, $\alpha n(1-\alpha)n$ is the total delay that the block of short jobs with total length αn adds to the completion times of the $(1-\alpha)n$ jobs that come after it, and $\frac{(1-\alpha)n((1-\alpha)n+1)}{2}\bar{p}$ is the sum of completion times for a block with $(1-\alpha)n$ job executions with processing time $\bar{p}$ per job.

Lower limits. A natural generalization of the problem would be to allow each job j to have, in addition to its upper limit $\bar{p}_j$, also a lower limit ℓ_j, such that the processing time after testing satisfies $\ell_j \leq p_j \leq \bar{p}_j$. We observe that the presence of lower limits has no effect on the optimal schedule, and can only help an algorithm. As we are interested in worst-case analysis, we assume in the remainder of the paper that every job has a lower limit of 0. Any algorithm that is ρ-competitive in this case is also ρ-competitive in the case with arbitrary lower limits (the algorithm can simply ignore the lower limits).

Jobs with small $\bar{p}_j$. We will consider several algorithms and prove competitiveness for them. We observe that any ρ-competitive algorithm may process jobs with $\bar{p}_j < \rho$ without testing in order of increasing $\bar{p}_j$ at the beginning of its schedule.

▶ **Lemma 1.** *Without loss of generality any algorithm* ALG *(deterministic or randomized) claiming competitive ratio ρ starts by scheduling untested all jobs j with $\bar{p}_j < \rho$ in increasing order of $\bar{p}_j$. Also worst-case instances for* ALG *consist solely of jobs j with $\bar{p}_j \geq \rho$.*

Increasing or decreasing Alg and Opt. Throughout the paper we sometimes consider worst-case instances consisting of only a few different job types. The following proposition allows us to do so in some cases.

▶ **Proposition 2.** *Fix some algorithm* Alg *and consider a family of instances described by some parameter $x \in [\ell, u]$, which could represent p_j or $\bar{p}_j$ for some job j or for some set of jobs. Suppose that both* Opt *and* Alg *are linear in x for the range $[\ell, u]$. Then the ratio* Alg/Opt *does not decrease for at least one of the two choices $x = \ell$ or $x = u$. Moreover, if* Opt *and* Alg *are increasing in x with the same slope, then this holds for $x = \ell$.*

We can make successive use of this proposition in order to show useful properties on worst-case instances. For this purpose we say that a schedule is *insensitive* to changes of the processing times, if the order of tests and job executions does not change, even though some completion times could be shifted.

▶ **Lemma 3.** *Suppose that there is an interval $[\ell', u']$ such that* Opt *schedules all jobs j with $p_j \in [\ell', u']$ either all tested or all untested, independently of the actual processing time in $[\ell', u']$. Suppose that this holds also for* Alg*. Moreover suppose that both* Opt *and* Alg *are insensitive to changes of the processing times in $[\ell', u']$ which maintain the ordering of processing times. Then there is a worst-case instance for* Alg *where every job j with $p_j \in [\ell', u']$ satisfies $p_j \in \{\ell', u'\}$.*

Proof sketch. The proof consists of fixing some set of jobs S of the same processing time and identifying an interval $[\ell, u]$ around it with $[\ell, u] \subseteq [\ell', u']$. Then we can show that both costs Opt and Alg are linear in x when changing the processing times of the jobs in S to any value $x \in [\ell, u]$. Proposition 2 implies that choosing $x \in \{\ell, u\}$ does not decrease the competitive ratio. This argument can be repeated until the number of distinct processing times in the open interval (ℓ', u') decreases to zero. ◀

3 Deterministic Algorithms

3.1 Algorithm Threshold

We show a competitive ratio of 2 for a natural algorithm that uses a threshold to decide whether to test a job or execute it untested.

▶ **Algorithm** (Threshold). *First jobs with $\bar{p}_j < 2$ are scheduled in order of non-decreasing upper limits without testing. Then all remaining jobs are tested. If the revealed processing time of job j is $p_j \leq 2$ (short jobs), then the job is executed immediately after its test. After all pending jobs (long jobs) have been tested, they are scheduled in order of increasing processing time p_j.*

By Lemma 1 we may restrict our analysis w.l.o.g. to instances with $\bar{p}_j \geq 2$. Note, that on such instances Threshold tests all jobs. From a simple interchange argument it follows that the structure of the algorithm's solution in a worst-case instance is as follows.

- The algorithm tests all jobs that have $p_j > 2$, and defers them.
- The algorithm tests short jobs ($p_j \leq 2$) and executes each of them right away. The jobs are tested in order of non-increasing processing time.
- The algorithm executes all deferred long jobs in order of non-decreasing processing times.

An optimal solution will not test jobs with $p_j + 1 \geq \bar{p}_j$. It sorts jobs in non-decreasing order of values $\min\{1 + p_j, \bar{p}_j\}$.

First, we analyze and simplify worst-case instances.

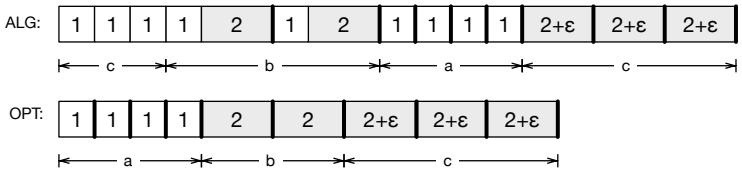


Figure 2 Worst-case instance for THRESHOLD.

▶ **Lemma 4.** *There is a worst-case instance for* THRESHOLD *in which all short jobs with* $p_j \leq 2$ *have processing time either* 0 *or* 2.

▶ **Lemma 5.** *There is a worst-case instance in which long jobs with* $p_j > 2$ *satisfy* $p_j = \bar{p}_j = 2 + \epsilon$ *for infinitesimally small* $\epsilon > 0$.

Now we are ready to prove the main result.

▶ **Theorem 6.** *Algorithm* THRESHOLD *has competitive ratio at most* 2.

Proof. We consider worst-case instances given by Lemmas 4 and 5. Let a be the number of short jobs with $p_j = 0$, let b be the number of short jobs with $\bar{p}_j = p_j = 2$, and let c be the number of long jobs with $\bar{p}_j = 2 + \epsilon$, see Figure 2.

THRESHOLD's solution for a worst-case instance first tests all long jobs, then tests and executes the short jobs in decreasing order of processing times, and completes with the executions of long jobs. The total objective value ALG is

$$\text{ALG} = (a+b+c)c + b(b+1)/2 \cdot 3 + 3b(a+c) + a(a+1)/2 + a \cdot c + c(c+1)/2 \cdot (2+\epsilon).$$

An optimum solution tests and schedules first all 0-length jobs and then executes the remaining jobs without tests. The objective value is

$$OPT = a(a+1)/2 + a(b+c) + b(b+1)/2 \cdot 2 + 2bc + c(c+1)/2 \cdot (2+\epsilon).$$

Simple transformation shows that $\text{ALG} \leq 2 \cdot OPT$ is equivalent to

$$2ab + 2c^2 \leq a^2 + b^2 + a + b + c(c+1)(2+\epsilon) \;\Leftrightarrow\; 0 \leq (a-b)^2 + a + b + c^2\epsilon + c(2+\epsilon),$$

which is obviously satisfied and the theorem follows. ◀

3.2 Deterministic lower bound

In this section we give a lower bound on the competitive ratio of any deterministic algorithm. The instances constructed by the adversary have a very special form: All jobs have the same upper limit $\bar{p}$, and the processing time of every job is either 0 or $\bar{p}$.

Consider instances of n jobs with uniform upper limit $\bar{p} > 1$, and consider any deterministic algorithm. We say that the algorithm *touches* a job when it either tests the job or executes it untested. We re-index jobs in the order in which they are first touched by the algorithm, i.e., job 1 is the first job touched by the algorithm and job n is the last. The adversary fixes a fraction $\delta \in [0,1]$ and sets the processing time of job j, $1 \leq j \leq n$, to:

$$p_j = \begin{cases} 0 & \text{, if } j \text{ is executed by the algorithm untested, or } j > \delta n \\ \bar{p} & \text{, if } j \text{ is tested by the algorithm and } j \leq \delta n \end{cases}$$

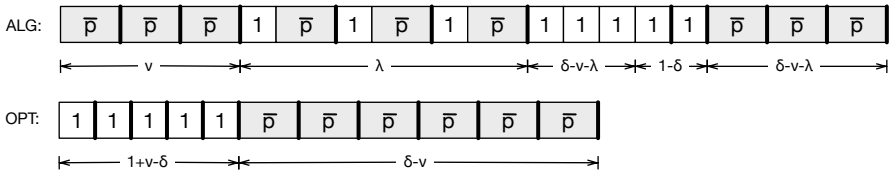


Figure 3 Lower bound construction.

A job j is called *short* if $p_j = 0$ and *long* if $p_j = \bar{p}$. Let j_0 be the smallest integer that is greater than δn. Job j_0 is the first of the last $(1-\delta)n$ jobs, which are short no matter whether the algorithm tests them or not.

We assume the algorithm knows $\bar{p}$ and δ, which can only improve the performance of the best-possible deterministic algorithm. Note that with δ and $\bar{p}$ known to the algorithm, it has full information about the actions of the adversary. Nevertheless, it is still non-trivial for an algorithm to decide for each of the first δn jobs whether to test it (which makes the job a long job, and hence the algorithm spends time $\bar{p}+1$ on it while the optimum executes it untested and spends only time $\bar{p}$) or to execute it untested (which makes it a short job, and hence the algorithm spends time $\bar{p}$ on it while the optimum spends only time 1).

Let us first determine the structure of the schedule produced by an algorithm that achieves the best possible competitive ratio for instances created by this adversary, as displayed in Figure 3.

▶ **Lemma 7.** *The schedule of a deterministic algorithm with best possible competitive ratio has the following form, where $\lambda, \nu \geq 0$ and $\nu + \lambda \leq \delta$: The algorithm first executes νn jobs untested, then tests and executes λn long jobs, then tests $(\delta - \nu - \lambda)n$ long jobs and delays their execution, then tests and executes the remaining $(1-\delta)n$ short jobs, and finally executes the $(\delta - \nu - \lambda)n$ delayed long jobs that were tested earlier.*

The cost of the algorithm in dependence on ν, λ, δ and $\bar{p}$ can now be expressed as:

$$\begin{aligned}\textsc{Alg}(\nu,\lambda,\delta,\bar{p}) &= n^2\big(\frac{\nu^2}{2}\bar{p} + \nu\bar{p}(1-\nu) + \frac{\lambda^2}{2}(1+\bar{p}) + \lambda(1+\bar{p})(1-\nu-\lambda) \\ &\quad + (\delta-\nu-\lambda)(1-\nu-\lambda) + \frac{(1-\delta)^2}{2} + (1-\delta)(\delta-\nu-\lambda) + \frac{(\delta-\nu-\lambda)^2}{2}\bar{p}\big) + O(n) \\ &= \frac{n^2}{2}(1 + 2\delta(1-\nu\bar{p}) + \delta^2(\bar{p}-1) + 2\nu(\nu+\bar{p}-2) + \lambda^2 + 2\lambda(\nu+\bar{p}-1-\delta\bar{p})) + O(n)\end{aligned}$$

The optimal schedule first tests and executes the $(\nu+1-\delta)n$ short jobs and then executes the $(\delta-\nu)n$ long jobs untested. Hence, the optimal cost, which depends only on ν, δ and $\bar{p}$, is:

$$\begin{aligned}\textsc{Opt}(\nu,\delta,\bar{p}) &= n^2\left(\frac{(\nu+1-\delta)^2}{2} + (\nu+1-\delta)(\delta-\nu) + \frac{(\delta-\nu)^2}{2}\bar{p}\right) + O(n) \\ &= \frac{n^2}{2}(1 + (\delta-\nu)^2(\bar{p}-1)) + O(n)\end{aligned}$$

Let $\textsc{Alg}'(\nu,\lambda,\delta,\bar{p}) = \lim_{n\to\infty}\frac{1}{n^2}\textsc{Alg}(\nu,\lambda,\delta,\bar{p})$ and $\textsc{Opt}'(\nu,\delta,\bar{p}) = \lim_{n\to\infty}\frac{1}{n^2}\textsc{Opt}(\nu,\delta,\bar{p})$. As the adversary can choose δ and $\bar{p}$, while the algorithm can choose ν and λ, the value

$$R = \max_{\delta,\bar{p}}\min_{\nu,\lambda}\frac{\textsc{Alg}'(\nu,\lambda,\delta,\bar{p})}{\textsc{Opt}'(\nu,\delta,\bar{p})}$$

gives a lower bound on the competitive ratio of any deterministic algorithm in the limit for $n \to \infty$. By making n sufficiently large, the adversary can create instances with finite n that give a lower bound that is arbitrarily close to R.

The exact optimization of δ and $\bar{p}$ is rather technical as it involves the optimization of rational functions of several variables. The choices $\delta = 0.6306655$ and $\bar{p} = 1.9896202$ give a lower bound of 1.854628. (The fully optimized value of R is less than $1.1 \cdot 10^{-7}$ larger.)

▶ **Theorem 8.** *No deterministic algorithm can achieve a competitive ratio below* 1.854628. *This holds even for instances with uniform upper limit where each processing time is either* 0 *or equal to the upper limit.*

4 Randomized Algorithms

4.1 Algorithm Random

▶ **Algorithm** (RANDOM). *The algorithm has parameters* $1 \leq T \leq E$ *and works in 3 phases. First it executes all jobs with* $\bar{p}_j < T$ *without testing in order of increasing* $\bar{p}_j$. *Then it tests all jobs with* $\bar{p}_j \geq T$ *in uniform random order. Each tested job* j *is executed immediately after its test if* $p_j \leq E$ *and is deferred otherwise. Finally all deferred jobs are executed in order of increasing processing-time.*

We analyze the competitive ratio of RANDOM, and optimize the parameters T, E such that the resulting competitive ratio is T.

▶ **Theorem 9.** *The competitive ratio of the algorithm* RANDOM *is at most* 1.7453.

Proof sketch. By Lemma 1 we restrict to instances with $\bar{p}_j \geq T$ for all jobs. Then, the schedule produced by RANDOM can be divided into two parts, see Figure 4. Part (1) contains all tests, of which those that yield processing time p_j at most E are immediately followed by the job's execution. Part (2) contains all jobs that are tested and have processing time larger than E. These jobs are ordered by increasing processing time. Jobs in the first part are completed in an arbitrary order.

Furthermore, we can assume $\bar{p}_j = \max\{p_j, T\}$ for all jobs. Reducing $\bar{p}_j$ to this value does not change the cost or behavior of RANDOM, but may decrease the cost of OPT. We make extensive use of Proposition 2 to show that the worst-case instance consists of only 4 types of jobs. As a result the adversary chooses fractions $\alpha, \beta, \gamma \geq 0$ (of total sum at most 1) describing the following instance.

- A $1 - \alpha - \beta - \gamma$ fraction of the jobs have $\bar{p}_j = T$ and $p_j = 0$.
- An α fraction of the jobs have $\bar{p}_j = T$ and $p_j = T$.
- A β fraction of the jobs have $\bar{p}_j = E$ and $p_j = E$.
- A γ fraction of the jobs have $\bar{p}_j = E + \epsilon$ and $p_j = E + \epsilon$ for some arbitrarily small $\epsilon > 0$.

In this setting the cost of RANDOM as well as the optimal cost can be expressed as a function in $\alpha, \beta, \gamma, T, E, n$ each consisting of two parts: one multiple of n^2 and the other multiple of n. We start to analyze the ratio only for the first part, and verify that the ratio also holds for the second part.

For the analysis we consider the expression $g := T \cdot \text{OPT} - \text{ALG}$ which is non-negative iff the ratio is at most T. Hence the adversary chooses fractions α, β, γ which minimize g. Our general approach is therefore to identify local minima for g inside the polytope $\{(\alpha, \beta, \gamma) | \alpha, \beta, \gamma \geq 0, \alpha + \beta + \gamma \leq 1\}$. This is done by standard second order analysis with the help of Mathematica, distinguishing the cases when such a minimum lies in the inner region

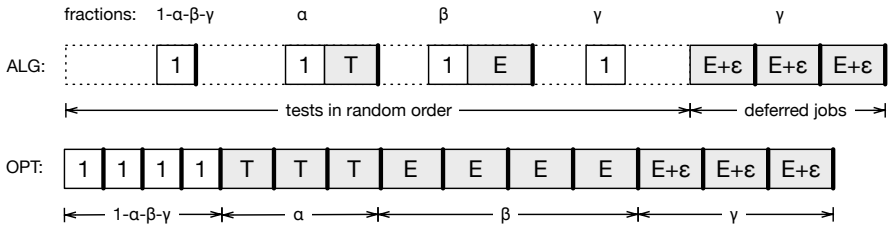


Figure 4 Worst-case analysis of the algorithm RANDOM.

of the polytope, or on one of the 4 triangular shaped facets or on one of the 6 one-dimensional edges. Each of these minima generates inequalities for T, E, generated by $g \geq 0$, which the algorithm needs to respect if the ratio is to be at most T in all cases. Our proof identifies the critical inequalities, and optimizes T, E for them, obtaining algebraic values, which are approximately $T \approx 1.7453$ and $E \approx 2.8609$. ◀

4.2 Lower bound for randomized algorithms

In this section we give a lower bound on the best possible competitive ratio of any randomized algorithm against an oblivious adversary. We do so by specifying a probability distribution over inputs and proving a lower bound on $E[\text{ALG}]/E[\text{OPT}]$ that holds for all deterministic algorithms ALG. By Yao's principle [14, 1] this gives the desired lower bound.

The probability distribution over inputs with n jobs has a constant parameter $0 < q < 1$ and is defined as follows: Each job j has upper limit $\bar{p}_j = 1/q > 1$, and its processing time p_j is set to 0 with probability q and to $1/q$ with probability $1 - q$. We show the expected optimal cost is

$$E[\text{OPT}] = \frac{n^2}{2}\left(\frac{1}{q} + 3q - 2 - q^2\right) + O(n).$$

We determine the algorithm cost by induction. With a case distinction on how the algorithm handles the first job, if it tests or executes it, we show the expected algorithm cost is at least $E[\text{ALG}(n)] \geq \frac{n^2}{2q}$. For $q = 1 - 1/\sqrt{3} \approx 0.4227$, this yields a lower bound of 1.6257 by Yao's principle.

▶ **Theorem 10.** *No randomized algorithm can achieve a competitive ratio less than* 1.6257.

5 Deterministic Algorithms for Uniform Upper Limits

5.1 An improved algorithm for uniform upper limits

In this section we present an algorithm for instances with uniform upper limit $\bar{p}$ that achieves a ratio strictly less than 2. We present a new algorithm BEAT that performs well on instances with upper limit roughly 2, but its performance becomes worse for larger upper limits. Thus, in this case we employ the algorithm THRESHOLD presented in Section 3.1.

To simplify the analysis, we consider the limit of $\text{ALG}(I)/\text{OPT}(I)$ when the number n of jobs approaches infinity. We say that an algorithm ALG is *asymptotically ρ_∞-competitive* or *has asymptotic competitive ratio at most ρ_∞* if $\lim_{n\to\infty} \sup_I \text{ALG}(I)/\text{OPT}(I) \leq \rho_\infty$.

▶ **Algorithm** (BEAT). *The algorithm* BEAT *balances the time testing jobs and the time executing jobs while there are untested jobs. A job is called* short *if its running time is at*

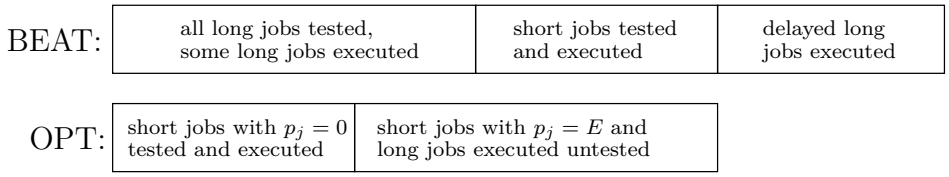


Figure 5 Structure of schedules produced by BEAT and OPT.

most $E = \max\{1, \bar{p} - 1\}$*, and* long *otherwise. Let TotalTest denote the time we spend testing long jobs and let TotalExec be the time long jobs are executed. We iterate testing an arbitrary job and then execute the job with smallest processing time either, if it is a short job, or if* $\text{TotalExec} + p_k$ *is at most TotalTest. Once all jobs have been tested, we execute the remaining jobs in order of non-decreasing processing time.*

We make a structural observation about the algorithm schedule for a worst-case instance.

▶ **Lemma 11.** *The adversary gives jobs with* $p_j \in \{0, E, \bar{p}\}$ *and at most one job with* $p_j \in (E, \bar{p})$ *in order of decreasing* p_j*.*

Consequently, the schedule produced by BEAT and the optimal schedule display a clear structure, which we depict in Figure 5.

We prove that the asymptotic competitive ratio of BEAT for $\bar{p} < 3$ is at most

$$\rho_{\infty}^{BEAT} = \frac{1 + 2(-2 + \bar{p})\bar{p} + \sqrt{(1 - 2\bar{p})^2(-3 + 4\bar{p})}}{2(-1 + \bar{p})\bar{p}},$$

which is a function decreasing in $\bar{p}$.

For small upper limit we can execute each job without test. Then there is a worst-case instance where all jobs have processing time $p_j = 0$. The optimal schedule tests each job only if the upper limit $\bar{p}$ is larger than 1 and executes it immediately. For $\bar{p} < 1$ this means the competitive ratio is 1 and otherwise it is $\bar{p}$, which monotonically increases. Thus, we choose a threshold $T_1 \approx 1.9338$ for $\bar{p}$, deciding if we run jobs untested or apply BEAT. T_1 is the fixpoint of the function ρ_{∞}^{BEAT}.

For upper limit $\bar{p} > 3$, the performance behavior of BEAT changes and the asymptotic competitive ratio increases. Thus, we employ the algorithm THRESHOLD for large upper limits. Recall that for $\bar{p} > 2$ THRESHOLD tests all jobs, executes those with $p_j \leq 2$ immediately and defers the other jobs. By Lemma 4, there is a worst-case instance with short jobs that have processing time either 0 or 2. Moreover, we argue that in a worst case long jobs have processing time $\bar{p}_j = \bar{p}$ and that no long job is tested in an optimal solution. This allows us to prove

$$\rho_{\infty}^{THRESH} = \begin{cases} \frac{-3+\bar{p}+\sqrt{-15+\bar{p}(18+\bar{p})}}{2(\bar{p}-1)} & \text{if } \bar{p} \in (2,3) \\ \sqrt{3} \approx 1.73 & \text{if } \bar{p} \geq 3. \end{cases}$$

The function for small $\bar{p}$ is a monotone function decreasing from 2 to $\sqrt{3}$ in the limits for $\bar{p} \in (2, 3)$. We choose a threshold for $\bar{p}$, where we change from applying BEAT to employing THRESHOLD at $T_2 \approx 2.2948$, the crossing point of the two functions describing the competitive ratio of BEAT and THRESHOLD in $(2, 3)$.

▶ **Algorithm.** *Execute all jobs without test, if the upper limit* $\bar{p}$ *is less than* $T_1 \approx 1.9338$*. Otherwise, if the upper limit* $\bar{p}$ *is greater than* $T_2 \approx 2.2948$*, execute the algorithm* THRESHOLD*. For all upper limits between* T_1 *and* T_2*, execute the algorithm* BEAT*.*

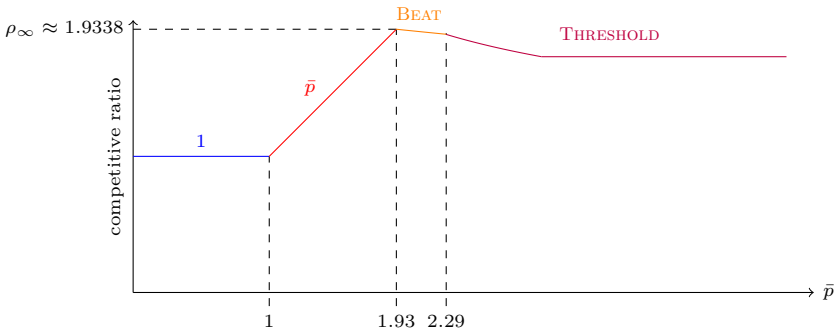


Figure 6 Competitive ratio depending on $\bar{p}$.

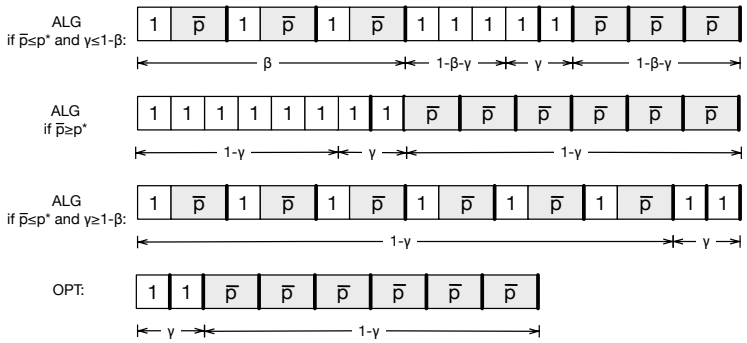


Figure 7 The schedule produced by UTE and the optimal schedule.

The function describing the asymptotic competitive ratio depending on $\bar{p}$ is displayed in Figure 6. Its maximum is attained at T_1, which is a fixpoint. Thus we have

▶ **Theorem 12.** *The asymptotic competitive ratio of our algorithm is* $\rho_\infty = T_1 \approx 1.9338$, *which is the only real root of* $2\bar{p}^3 - 4\bar{p}^2 + 4\bar{p} - 1 - \sqrt{(1-2\bar{p})^2(4\bar{p}-3)}$.

5.2 Nearly tight deterministic algorithm for extreme uniform instances

We present a deterministic algorithm for the class of *extreme uniform* instances, that is almost tight for the instance that yields the deterministic lower bound. An *extreme uniform* instance consists of jobs with uniform upper limit $\bar{p}$ and processing times in $\{0, \bar{p}\}$. Our algorithm UTE attains asymptotic competitive ratio $\rho_\infty \approx 1.8668$ for this class of instances.

▶ **Algorithm** (UTE). *If the upper limit* $\bar{p}$ *is at most* ρ, *then all jobs are executed without test. Otherwise, all jobs are tested. The first* $\max\{0, \beta\}$ *fraction of the jobs are executed immediately after their test. The remaining fraction of the jobs are executed immediately after their test if they have processing time* 0 *and are delayed otherwise, see Figure 7. The parameter* β *is defined as*

$$\beta = \frac{1 - \bar{p} + \bar{p}^2 - \rho + 2\bar{p}\rho - \bar{p}^2\rho}{1 - \bar{p} + \bar{p}^2 - \rho + \bar{p}\rho}. \tag{1}$$

▶ **Theorem 13.** *The competitive ratio of* UTE *is at most* $\rho = \frac{1+\sqrt{3+2\sqrt{5}}}{2} \approx 1.8668$.

Proof sketch. An instance is defined by the job number n, an upper limit $\bar{p}$ and a fraction γ such that the first $1-\gamma$ fraction of the jobs tested by UTE have processing time $\bar{p}$, while the jobs in the remaining γ fraction have processing time 0. The algorithm chooses β so as to have the smallest ratio ρ.

First we observe that there is a value p^* such that $\beta \geq 0$ only when $\bar{p} \leq p^*$. Then we analyze the competitive ratio of UTE, distinguishing the cases $\bar{p} \leq \rho$ (covered by Lemma 1), $\bar{p} \geq p^*$ and $\bar{p} \leq p^*$. The last case is furthermore subdivided into cases $\gamma \geq 1-\beta$ and $\gamma \leq 1-\beta$. For each of these cases we use first and second order analysis to determine the worst values $\bar{p}$ and γ for the ratio, and the best response β the algorithm can choose, optimizing β and ρ on the way, and obtaining the claimed values. ◀

▶ Remark. The deterministic lower bound 1.8546 in Theorem 8 uses the upper limit $\bar{p} \approx 1.9896$. Plugging this choice of $\bar{p}$ into a precise form of the competitive ratio which we obtained in the proof of the theorem, we can show that UTE has asymptotic competitive ratio $\rho_\infty \approx 1.8552$ on this instance. This is almost tight.

6 Optimal Testing for Minimizing the Makespan

We consider scheduling with testing with the objective of minimizing the makespan, i.e., the completion time of the last job. For this problem we give the best possible competitive ratio for deterministic and randomized algorithms. The key insight is that for any algorithm that treats each job independent of its position in the schedule, there is a worst-case instance containing only a single job. The reason is that the execution of a job (possibly including testing) has a linear contribution to the makespan.

▶ **Theorem 14.** *Let $\varphi \approx 1.618$ be the golden ratio. Testing each job j if and only if $\bar{p}_j > \varphi$ is an algorithm with competitive ratio φ. This is best possible for deterministic algorithms.*

▶ **Theorem 15.** *The randomized algorithm that tests each job with $\bar{p}_j > 1$ with probability $1 - 1/(\bar{p}_j^2 - \bar{p}_j + 1)$ has competitive ratio $4/3$. No randomized algorithm can achieve a better competitive ratio against an oblivious adversary.*

7 Conclusion

In this paper we have introduced an adversarial model of scheduling with testing where a test can shorten a job but the time for the test also prolongs the schedule, thus making it difficult for an algorithm to find the right balance between tests and executions. We have presented upper and lower bounds on the competitive ratio of deterministic and randomized algorithms for a single-machine scheduling problem with the objective of minimizing the sum of completion times or the makespan. An immediate open question is whether it is possible to achieve competitive ratio below 2 for minimizing the sum of completion times with a deterministic algorithm for arbitrary instances. Further interesting directions for future work include the consideration of job-dependent test times or other scheduling problems such as parallel machine scheduling or flow shop problems. More generally, the study of problems with explorable uncertainty in settings where the costs for querying uncertain data directly contribute to the objective value is a promising direction for future work.

References

1 Allan Borodin and Ran El-Yaniv. *Online Computation and Competitive Analysis*. Cambridge University Press, 1998.

2 Christoph Dürr, Thomas Erlebach, Nicole Megow, and Julie Meißner. Scheduling with explorable uncertainty. *CoRR*, abs/1709.02592, 2017. arXiv:1709.02592.

3 Thomas Erlebach, Michael Hoffmann, Danny Krizanc, Matús Mihalák, and Rajeev Raman. Computing minimum spanning trees with uncertainty. In *25th International Symposium on Theoretical Aspects of Computer Science (STACS 2008)*, pages 277–288, 2008.

4 Tomás Feder, Rajeev Motwani, Liadan O'Callaghan, Chris Olston, and Rina Panigrahy. Computing shortest paths with uncertainty. *Journal of Algorithms*, 62(1):1–18, 2007.

5 Tomás Feder, Rajeev Motwani, Rina Panigrahy, Chris Olston, and Jennifer Widom. Computing the median with uncertainty. *SIAM Journal on Computing*, 32(2):538–547, 2003.

6 Marc Goerigk, Manoj Gupta, Jonas Ide, Anita Schöbel, and Sandeep Sen. The robust knapsack problem with queries. *Computers & OR*, 55:12–22, 2015.

7 Manoj Gupta, Yogish Sabharwal, and Sandeep Sen. The update complexity of selection and related problems. In *IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2011)*, volume 13 of *LIPIcs*, pages 325–338. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2011.

8 Simon Kahan. A model for data in motion. In *23rd Annual ACM Symposium on Theory of Computing (STOC 1991)*, pages 267–277, 1991.

9 Sanjeev Khanna and Wang-Chiew Tan. On computing functions with uncertainty. In *20th Symposium on Principles of Da18th Annual Symposium on Database Systems (PODS 2001)*, pages 171–182, 2001.

10 Retsev Levi. Practice driven scheduling models. Talk at Dagstuhl Seminar 16081: Scheduling, 2016.

11 Nicole Megow, Julie Meißner, and Martin Skutella. Randomization helps computing a minimum spanning tree under uncertainty. In *Algorithms – Proceedings of the 23rd Annual European Symposium (ESA 2015)*, LNCS 9294, pages 878–890. Springer, 2015.

12 Chris Olston and Jennifer Widom. Offering a precision-performance tradeoff for aggregation queries over replicated data. In *26th International Conference on Very Large Data Bases (VLDB 2000)*, pages 144–155, 2000.

13 Yaron Shaposhnik. *Exploration vs. Exploitation: Reducing Uncertainty in Operational Problems*. PhD thesis, Sloan School of Management, MIT, 2016.

14 Andrew Chi-Chin Yao. Probabilistic computations: Toward a unified measure of complexity. In *18th Annual Symposium on Foundations of Computer Science (FOCS 1977)*, pages 222–227. IEEE, 1977.

A Local-Search Algorithm for Steiner Forest*†

Martin Groß[1], Anupam Gupta[2], Amit Kumar[3], Jannik Matuschke[4], Daniel R. Schmidt[5], Melanie Schmidt[6], and José Verschae[7]

1 **Institut für Mathematik, Technische Universität Berlin, Germany**
gross@math.tu-berlin.de
2 **Dept. of Computer Science, Carnegie Mellon University, Pittsburgh, USA**
anupamg@cs.cmu.edu
3 **Dept. of Comp.Sci. and Engg., Indian Institute of Technology, Delhi, India**
amitk@cse.iitd.ernet.in
4 **TUM School of Management, Technische Universität München, Germany**
jannik.matuschke@tum.de
5 **Institut für Informatik, Universität zu Köln, Germany**
schmidt@informatik.uni-koeln.de
6 **Institut für Informatik, Universität Bonn, Germany**
melanieschmidt@uni-bonn.de
7 **Facultad de Matemáticas & Escuela de Ingeniería, Pontificia Universidad Católica de Chile, Santiago, Chile**
jverschae@uc.cl

Abstract

In the *Steiner Forest* problem, we are given a graph and a collection of source-sink pairs, and the goal is to find a subgraph of minimum total length such that all pairs are connected. The problem is APX-Hard and can be 2-approximated by, e.g., the elegant primal-dual algorithm of Agrawal, Klein, and Ravi from 1995.

We give a local-search-based constant-factor approximation for the problem. Local search brings in new techniques to an area that has for long not seen any improvements and might be a step towards a combinatorial algorithm for the more general survivable network design problem. Moreover, local search was an essential tool to tackle the dynamic MST/Steiner Tree problem, whereas dynamic Steiner Forest is still wide open.

It is easy to see that any constant factor local search algorithm requires steps that add/drop many edges together. We propose natural local moves which, at each step, either (a) add a shortest path in the current graph and then drop a bunch of inessential edges, or (b) add a set of edges to the current solution. This second type of moves is motivated by the potential function we use to measure progress, combining the cost of the solution with a penalty for each connected component. Our carefully-chosen local moves and potential function work in tandem to eliminate bad local minima that arise when using more traditional local moves.

Our analysis first considers the case where the local optimum is a single tree, and shows optimality w.r.t. moves that add a single edge (and drop a set of edges) is enough to bound the locality gap. For the general case, we show how to "project" the optimal solution onto the different trees of the local optimum without incurring too much cost (and this argument uses optimality w.r.t. both kinds of moves), followed by a tree-by-tree argument. We hope both the potential function, and our analysis techniques will be useful to develop and analyze local-search algorithms in other contexts.

* This work was partially supported by the DFG within project A07 of CRC TRR 154, by the German Academic Exchange Service (DAAD), by the Alexander von Humboldt Foundation with funds of the German Federal Ministry of Education and Research (BMBF), by Nucleo Milenio Información y Coordinación en Redes ICM/FIC P10-024F, and by NSF awards CCF-1536002, CCF-1540541, and CCF-1617790.

† A full version of this paper is available at https://arxiv.org/abs/1707.02753

9th Innovations in Theoretical Computer Science Conference (ITCS 2018).
Editor: Anna R. Karlin; Article No. 31; pp. 31:1–31:17

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

1998 ACM Subject Classification F.2.2 Nonnumerical Algorithms and Problems

Keywords and phrases Local Search, Steiner Forest, Approximation Algorithms, Network Design

Digital Object Identifier 10.4230/LIPIcs.ITCS.2018.31

1 Introduction

The Steiner Forest problem is the following basic network design problem: given a graph $G = (V, E)$ with edge-lengths d_e, and a set of source-sink pairs $\{\{s_i, t_i\}\}_{i=1}^{k}$, find a subgraph H of minimum total length such that each $\{s_i, t_i\}$ pair lies in the same connected component of H. This problem generalizes the Steiner Tree problem, and hence is APX-hard. The Steiner Tree problem has a simple 2-approximation, namely the minimum spanning tree on the terminals in the metric completion; however, the forest version does not have such obvious algorithms.

Indeed, the first approximation algorithm for this problem was a sophisticated and elegant primal-dual 2-approximation due to Agrawal, Klein, and Ravi [1]. Subsequently, Goemans and Williamson streamlined and generalized these ideas to many other constrained network design problems [15]. These results prove an integrality gap of 2 for the natural cut-covering LP. Other proofs of this integrality gap were given in [21, 7]. No better LP relaxations are currently known (despite attempts in, e.g., [24, 25]), and improving the approximation guarantee of 2 remains an outstanding open problem. Note that all known constant-factor approximation algorithms for Steiner Forest were based on linear programming relaxations, until a recent greedy algorithm [18]. In this paper, we add to the body of techniques that give constant-factor approximations for Steiner Forest. The main result of this paper is the following:

▶ **Theorem 1.** *There is a (non-oblivious) local search algorithm for Steiner Forest with a constant locality gap. It can be implemented to run in polynomial time.*

The Steiner Forest problem is a basic network problem whose approximability has not seen any improvements in some time. We explore new techniques to attacking the problem, with the hope that these will give us more insights into its structure. Moreover, for many problems solved using the constrained forest approach of [15], the only constant factor approximations known are via the primal-dual/local-ratio approach, and it seems useful to bring in new possible techniques. Another motivation for our work is to make progress towards obtaining combinatorial algorithms for the survivable network design problem. In this problem, we are given connectivity requirements between various source-sink pairs, and we need to find a minimum cost subset of edges which provide this desired connectivity. Although we know a 2-approximation algorithm for the survivable network design problem [21] based on iterative rounding, obtaining a combinatorial constant-factor approximation algorithm for this problem remains a central open problem [34]. So far, all approaches of extending primal-dual or greedy algorithms to survivable network design have only had limited success. Local search algorithms are more versatile in the sense that one can easily *propose* algorithms based on local search for various network design problems. Therefore, it is important to understand the power of such algorithms in such settings. We take a step towards this goal by showing that such ideas can give constant-factor approximation algorithms for the Steiner Forest problem.

Finally, we hope this is a step towards solving the *dynamic Steiner Forest* problem. In this problem, terminal pairs arrive online and we want to maintain a constant-approximate Steiner Forest while changing the solution by only a few edges in each update. Several of the

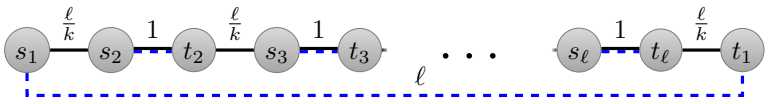


Figure 1 The black edges (continuous lines) are the current solution. If $\ell \gg k$, we should move to the blue forest (dashed lines), but any improving move must change $\Omega(k)$ edges. Details can be found in Section A.1.

approaches used for the Steiner *Tree* case (e.g., in [30, 16, 27]) are based on local-search, and we hope our local-search algorithm for Steiner Forest in the offline setting will help solve the dynamic Steiner Forest problem, too.

1.1 Our Techniques

One of the challenges with giving a local-search algorithm for Steiner Forest is to find the right set of moves. Indeed, it is easy to see that simpler approaches like just adding and dropping a constant number of edges at each step is not enough. E.g., in the example of Figure 1, the improving moves must add an edge and remove multiple edges. (This holds even if we take the metric completion of the graph.) We therefore consider a natural generalization of simple edge swaps in which we allow to add paths and remove multiple edges from the induced cycle.

Local Moves: Our first task is to find the "right" moves that add/remove many edges in each "local" step. At any step of the algorithm, our algorithm has a feasible forest, and performs one of these local moves (which are explained in more detail in Section 3):

- **edge/set swaps**: Add an edge to a tree in the current forest, and remove one or more edges from the cycle created.
- **path/set swaps**: Instead of one edge, add a set of edges to connect two vertices from the same tree T in the current forest, creating exactly one cycle, then remove edges from the cycle. The set of edges shall be a shortest path in the graph where all trees except T are contracted.
- **connecting moves**: Connect some trees of the current forest by adding edges between them.

At the end of the algorithm, we apply the following post-processing step to the local optimum:

- **clean-up**: Delete all inessential edges. (An edge is *inessential* if dropping it does not alter the feasibility of the solution.)

Given these local moves, the challenge is to bound the locality gap of the problem: the ratio between the cost of a local optimum and that of the global optimum.

The Potential. The connecting moves may seem odd, since they only increase the length of the solution. However, a crucial insight behind our algorithm is that we do not perform local search with respect to the total length of the solution. Instead we look to improve a different potential ϕ. (In the terminology of [3, 23], our algorithm is a *non-oblivious* local search.) The potential $\phi(T)$ of a tree T is the total length of its edges, plus the distance between the furthest source-sink pair in it, which we call its *width*. The potential of the forest $\mathcal{A}$ is the sum of the potentials of its trees. We only perform moves that cause the potential of the resulting solution to decrease.

In Section A.2 we give an example where performing the above moves with respect to the total length of the solution gives us local optima with cost $\Omega(\log n) \cdot \mathrm{OPT}$ — this example is useful for intuition for why using this potential helps. Indeed, if we have a forest where the distance between two trees in the forest is much less than both their widths, we can merge them and reduce the potential (even though we increase the total length). So the trees in a local optimum are "well-separated" compared to their widths, an important property for our analysis.

The Proof. We prove the constant locality gap in two conceptual steps.

As the first step, we assume that the local optimum happens to be a single tree. In this case we show that the essential edges of this tree T have cost at most $\mathcal{O}(\mathrm{OPT})$—hence the final removal of inessential edges gives a good solution. To prove this, we need to charge our edges to OPT's edges. However, we cannot hope to charge single edges in our solution to single edges in OPT—we need to charge multiple edges in our solution to edges of OPT. (We may just have more edges than OPT does. More concretely, this happens in the example from Figure 1, when $\ell = \Theta(k)$ and we are at the black tree and OPT is the blue forest.) So we consider edge/set swaps that try to swap some subset S of T's edges for an edge f of OPT. Since such a swap is non-improving at a local optimum, the cost of S is no more than that of f. Hence, we would like to partition T's edges into groups and find an $O(1)$-to-1 map of groups to edges of OPT of no less cost. Even if we cannot find an explicit such map, it turns out that Hall's theorem is the key to showing its existence.

Indeed, the intuition outlined above works out quite nicely if we imagine doing the local search with respect to the total length instead of the potential. The main technical ingredient is a partitioning of our edges into equivalence classes that behave (for our purposes) "like single edges", allowing us to apply a Hall-type argument. This idea is further elaborated in Section 4.1. However, if we go back to considering the potential, an edge/set swap adding f and removing S may create multiple components, and thus increase the width part of the potential. Hence we give a more delicate argument showing that similar charging arguments work out: basically we now have to charge to the width of the globally optimal solution as well. A detailed synopsis is presented in Section 4.2.

The second conceptual step is to extend this argument to the case where we can perform all possible local moves, and the local optimum is a forest $\mathcal{A}$. If OPT's connected components are contained in those of $\mathcal{A}$, we can do the above analysis for each $\mathcal{A}$-component separately. So imagine that OPT has edges that go between vertices in different components of $\mathcal{A}$. We simply give an algorithm that takes OPT and "projects" it down to another solution OPT' of comparable cost, such that the new projected solution OPT' has connected components that are contained in the components of $\mathcal{A}$. We find the existence of a cheap projected solution quite surprising; our proof crucially uses the optimality of the algorithm's solution under both path/set swaps and connecting moves. A summary of our approach is in Section 5.

Polynomial-time Algorithm. The locality gap with respect to the above moves is at most 46. The swap moves can be implemented in polynomial time, and connecting moves can be approximated to within constant factors. Indeed, a c-approximation for *weighted* k-MST gives a $23(1 + c) + \varepsilon$-guarantee for the local search algorithm. Applying a weighted version of Garg's 2-approximation [13, 14] yields $c = 2$. The resulting approximation guarantee is 69 (compared to 96 for [18]).

1.2 Related Work

Local search techniques have been very successful for providing good approximation guarantees for a variety of problems: e.g., network design problems such as low-degree spanning trees [12], min-leaf spanning trees [29, 33], facility location and related problems, both uncapacitated [26, 4] and capacitated [31], geometric k-means [22], mobile facility location [2], and scheduling problems [32]. Other examples can be found in, e.g., the book of Williamson and Shmoys [34]. More recent are applications to optimization problems on planar and low-dimensional instances [10, 6]. In particular, the new PTAS for low dimensional k-means in is based on local search [9, 11].

Local search algorithms have also been very successful in practice – e.g., the widely used Lin-Kernighan heuristic [28] for the travelling salesman problem, which has been experimentally shown to perform extremely well [19].

Imase and Waxman [20] defined the dynamic Steiner tree problem where vertices arrive/depart online, and a few edge changes are performed to maintain a near-optimal solution. Their analysis was improved by [30, 16, 17, 27], but extending it to Steiner Forest remains wide open.

2 Preliminaries

Let $G = (V, E)$ be an undirected graph with non-negative edge weights $d_e \in \mathbb{R}_{\geq 0}$. Let $n := |V|$. For $W \subseteq V$, let $G[W] = (W, E[W])$ be the vertex-induced subgraph, and for $F \subseteq E$, $G[F] = (V[F], F)$ the edge-induced subgraph, namely the graph consisting of the edges in F and the vertices contained in them. A forest is a set of edges $F \subseteq E$ such that $G[F]$ is acyclic.

For a node set $W \subseteq V$ and an edge set $F \subseteq E$, let $\delta_F(W)$ denote the edges of F leaving W. Let $\delta_F(A : B) := \delta_F(A) \cap \delta_F(B)$ for two disjoint node sets $A, B \subseteq V$ be the set of edges that go between A and B. For forests $F_1, F_2 \subseteq E$ we use $\delta_F(F_1 : F_2) := \delta_F(V[F_1] : V[F_2])$. We may drop the subscript if it is clear from the context.

Let $\mathfrak{T} \subseteq \{\{v, \bar{v}\} \mid v, \bar{v} \in V\}$ be a set of terminal pairs. Denote the shortest-path distance between u and $\bar{u}$ in (G, d) by $\mathrm{dist}_d(u, \bar{u})$. Let n_t be the number of terminal pairs. We number the pairs according to non-decreasing shortest path distance (ties broken arbitrarily). Thus, $\mathfrak{T} = \{\{u_1, \bar{u}_1\}, \ldots, \{u_{n_t}, \bar{u}_{n_t}\}\}$ and $i < j$ implies $\mathrm{dist}_d(u_i, \bar{u}_i) \leq \mathrm{dist}_d(u_j, \bar{u}_j)$. This numbering ensures consistent tie-breaking throughout the paper. We say that $G = (V, E)$, the weights d and $\mathfrak{T}$ form a *Steiner Forest instance*. We often use $\mathcal{A}$ to denote a feasible Steiner forest held by our algorithm and $\mathcal{F}$ to denote an optimal/good feasible solution to which we compare $\mathcal{A}$.

Width. Given a connected set of edges E', the *width* $w(E')$ of E' is the maximum distance (in the original graph) of any terminal pair connected by E': i.e., $w(E') = \max_{\{u,\bar{u}\} \in \mathfrak{T}, u, \bar{u} \in V[E']} \mathrm{dist}_d(u, \bar{u})$. Notice that $w(E')$ is the width of the pair $\{u_i, \bar{u}_i\}$ with the largest i among all pairs in $V[E']$. We set $\mathtt{index}(E') := \max\{i \mid u_i, \bar{u}_i \in V[E']\}$, i.e., $w(E') = \mathrm{dist}_d(u_{\mathtt{index}(E')}, \bar{u}_{\mathtt{index}(E')})$.

For a subgraph $G[F] = (V[F], F)$ given by $F \subseteq E$ with connected components $E_1, \ldots, E_l \subseteq F$, we define the *total width* of F to be the sum $w(F) := \sum_{i=1}^{l} w(E_i)$ of the widths of its connected components. Let $d(F) := \sum_{e \in F} d_e$ be the sum of edge lengths of edges in F and define $\phi(F) := d(F) + w(F)$. By the definition of the width, it follows that $d(F) \leq \phi(F) \leq 2d(F)$.

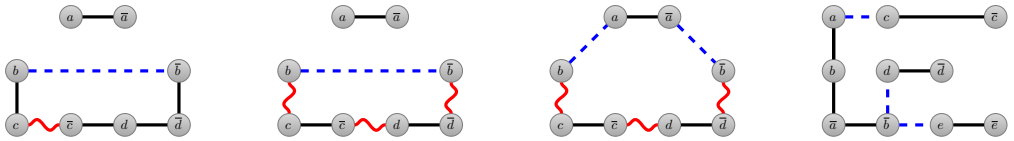


Figure 2 Our different moves. Black solid edges are not changed by the move. Blue dashed edges are added by the move. Red curled edges are removed by the move.

3 The Local Search Algorithm

Our local-search algorithm starts with a feasible solution $\mathcal{A}$, and iteratively tries to improve it. Instead of looking at the actual edge cost $d(\mathcal{A})$, we work with the potential $\phi(\mathcal{A})$ and decrease it over time.

In the rest of the paper, we say a move changing $\mathcal{A}$ into $\mathcal{A}'$ is *improving* if $\phi(\mathcal{A}') < \phi(\mathcal{A})$. A solution $\mathcal{A}$ is `<move>`-*optimal* with respect to a certain kind of move and with respect to a set of edges $\mathcal{F}$ if no move of that kind consisting of edges from $\mathcal{F}$ is improving.

Swaps. Swaps are moves that start with a cycle-free feasible solution $\mathcal{A}$, add some edges and remove others to get to another cycle-free feasible solution $\mathcal{A}'$.

- The most basic swap is: *add an edge e creating a cycle, remove an edge f from this cycle.* This is called an *edge/edge swap* (e, f).
- We can slightly generalize this: *add an edge e creating a cycle, and remove a subset S of edges from this cycle $C(e)$.* This is called the *edge/set swap* (e, S). Edge/edge swaps are a special case of edge/set swaps, so edge/set swap-optimality implies edge/edge swap-optimality.
  There may be many different subsets of $C(e)$ we could remove. A useful fact is that if we fix some edge $f \in C(e)$ to remove, this uniquely gives a maximal set $R(e, f) \subseteq C(e)$ of edges that can be removed along with f after adding e without violating feasibility. Indeed, $R(e, f)$ contains f, and also all edges on $C(e)$ that can be removed in $\mathcal{A} \cup \{e\} \setminus \{f\}$ without destroying feasibility. (See Lemma 13 in the full version for a formalization.)
  Moreover, given a particular $R(e, f)$, we could remove any subset $S \subseteq R(e, f)$. If we were doing local search w.r.t. $d(\mathcal{A})$, there would be no reason to remove a proper subset. But since the local moves try to reduce $\phi(\mathcal{A})$, removing a subset of $R(e, f)$ may be useful. If $e_1, \ldots, e_\ell$ are the edges in $R(e, f)$ in the order they appear on $C(e)$, we only need swaps where S consists of edges $e_i, \ldots, e_j$ that are consecutive in the above order. There are $\mathcal{O}(n^2)$ sets $S \subseteq R(e, f)$ that are consecutive.[1] Moreover, there are at most $n - 1$ choices for e and $O(n)$ choices for f, so the number of edge/set swaps is polynomial.
- A further generalization: we can pick two vertices u, v lying in some component T, add a shortest-path between them (in the current solution, where all other components are shrunk down to single points, and the vertices/edges in $T \setminus \{u, v\}$ are removed). This creates a cycle, and we want to remove some edges. We now imagine that we added a "virtual" edge $\{u, v\}$, and remove a subset of consecutive edges from some

[1] In fact, we only need five different swaps (e, S) for the following choices of consecutive sets S: The set $S = \{f\}$, the complete set $S = R(e, f)$, and three sets of the form $S = \{e_1, \ldots, e_i\}$, $S = \{e_{i+1}, \ldots, e_j\}$ and $S = \{e_{j+1}, \ldots, e_\ell\}$ for specific indices i and j. How to obtain the values for i and j is explained in Section 4.1.

$R(\{u,v\},f) \subseteq C(\{u,v\})$, just as if we'd have executed an edge/set swap with the "virtual" edge $\{u,v\}$. We call such a swap a *path/set swap* (u,v,S).
Some subtleties: Firstly, the current solution $\mathcal{A}$ may already contain an edge $\{u,v\}$, but the uv-shortest-path we find may be shorter because of other components being shrunk. So this move would add this shortest-path and remove the direct edge $\{u,v\}$—indeed, the cycle $C(uv)$ would consist of two parallel edges, and we'd remove the actual edge $\{u,v\}$. Secondly, although the cycle contains edges from many components, only edges within T are removed. Finally, there are a polynomial number of such moves, since there are $O(n^2)$ choices for u,v, $O(n)$ choices for f, and $O(n^2)$ consecutive removal sets S.

Note that edge/set swaps never decrease the number of connected components of $\mathcal{A}$, but path/set swaps may increase or decrease the number of connected components.

Connecting moves. Connecting moves reduce the number of connected components by adding a set of edges that connect some of the current components. Formally, let $G^{\text{all}}_{\mathcal{A}}$ be the (multi)graph that results from contracting all connected components of $\mathcal{A}$ in G, deleting loops and keeping parallel edges. A *connecting move* (denoted $conn(T)$) consists of picking a tree in $G^{\text{all}}_{\mathcal{A}}$, and adding the corresponding edges to $\mathcal{A}$. The number of possible connecting moves can be large, but we can show that an approximation for k-MST is sufficient to obtain an approximate connecting move that works appropriately (see full version).

Note that connecting moves cause $d(\mathcal{A}') > d(\mathcal{A})$, but since our notion of improvement is with respect to the potential ϕ, such a move may still cause the potential to decrease.

In addition to the above moves, the algorithm runs the following post-processing step at the end.

Clean-up. Remove the unique maximal edge set $S \subseteq \mathcal{A}$ such that $\mathcal{A} \setminus S$ is feasible, i.e., erase all unnecessary edges. This might increase $\phi(\mathcal{A})$, but it will never increase $d(\mathcal{A})$.

Checking whether an improving move exists is polynomial except for connecting moves, which we can do approximately. Thus, the local search algorithm can be made to run in polynomial time by using standard methods (see full version).

4 In Which the Local Optimum is a Single Tree

We want to bound the cost of a forest that is locally optimal with respect to the moves defined above. To start, let us consider a simpler case: suppose we were to find a single tree T that is optimal with respect to just the *edge/edge* and *edge/set* swaps. (Recall that edge/set swaps add an edge and remove a consecutive subset of the edges on the resulting cycle, while maintaining feasibility. Also, recall that optimality means that no such moves cause the potential ϕ to decrease.) Our main result of this section is the following:

▶ **Corollary 2.** *Let $G=(V,E)$ be a graph, let d_e be the cost of edge $e \in E$ and let $\mathfrak{T} \subseteq V \times V$ be a set of terminal pairs. Let $\mathcal{A},\mathcal{F} \subseteq E$ be two feasible Steiner forests for $(G,d,\mathfrak{T})$ with $V[\mathcal{A}] = V[\mathcal{F}]$. Assume that $\mathcal{A}$ is a tree and that $\mathcal{A}$ is swap-optimal with respect to $\mathcal{F}$ and ϕ under edge/edge and edge/set swaps. Denote by $\mathcal{A}'$ the modified solution where all inessential edges have been dropped from $\mathcal{A}$. Then,*

$$d(\mathcal{A}') \leq 10.5 \cdot d(\mathcal{F}) + w(\mathcal{F}) \leq 11.5 \cdot d(\mathcal{F}).$$

The actual approximation guarantee is 42 for this case: indeed, Corollary 2 assumes $V[\mathcal{A}] = V[\mathcal{F}]$, which can be achieved (by taking the metric completion on the terminals) at the cost of a factor 2.

The intuition here comes from a proof for the optimality of edge/edge swaps for the Minimum Spanning tree problem. Let $\mathcal{A}$ be the tree produced by the algorithm, and $\mathcal{F}$ the reference (i.e., optimal or near-optimal) solution, with $V[\mathcal{A}] = V[\mathcal{F}]$. Suppose we were looking for a minimum spanning tree instead of a Steiner forest: one way to show that edge/edge swaps lead to a global optimum is to build a bipartite graph whose vertices are the edges of $\mathcal{A}$ and $\mathcal{F}$, and which contains edge (e, f) when $f \in \mathcal{F}$ can be swapped for $e \in \mathcal{A}$ and $d_e \leq d_f$. Using the fact that all edge/edge swaps are non-improving, we can show that there exists a perfect matching between the edges in $\mathcal{A}$ and $\mathcal{F}$, and hence the cost of $\mathcal{A}$ is at most that of $\mathcal{F}$.

Our analysis is similar in spirit. Of course, we now have to (a) consider edge/*set* swaps, (b) do the analysis with respect to the potential ϕ instead of just edge-lengths, and (c) we cannot hope to find a perfect matching because the problem is NP-hard. These issues make the proofs more complicated, but the analogies still show through.

4.1 An approximation guarantee for trees and d

In this section, we conduct a thought-experiment where we imagine that we get a connected tree on the terminals which is optimal for edge/set swaps *with respect to just the edge lengths, not the potential*. In very broad strokes, we define an equivalence relation on the edges of $\mathcal{A}$, and show a constant-to-1 cost-increasing map from the resulting equivalence classes to edges of $\mathcal{F}$—again mirroring the MST analysis—and hence bounding the cost of $\mathcal{A}$ by a constant times the cost of $\mathcal{F}$. The analysis of the real algorithm in Section 4.2 builds on the insights we develop here.

Some Definitions. The crucial equivalence relation is defined as follows: For edges $e, f \in \mathcal{A}$, let $T_{e,f}$ be the connected component of $\mathcal{A} \setminus \{e, f\}$ that contains the unique e-f-path in $\mathcal{A}$. We say e and f are *compatible w.r.t.* $\mathcal{F}$ if $e = f$ or if there are no $\mathcal{F}$-edges leaving $T_{e,f}$, and denote it by $e \sim_{cp} f$. One can show that $\sim_{cp}$ is an equivalence relation (see full version). We denote the set of its equivalence classes by $\mathfrak{S}$.

An edge is *essential* if dropping it makes the solution infeasible. If T_1, T_2 are the connected components of $\mathcal{A} \setminus \{e\}$, then e is called *safe* if at least one edge from $\mathcal{F}$ crosses between T_1 and T_2. Observe that any essential edge is safe, but the converse is not true: safe edges can be essential or inessential. However, it turns out that the set S_u of all unsafe edges in $\mathcal{A}$ forms an equivalence class of $\sim_{cp}$. Hence, all other equivalence classes in $\mathfrak{S}$ contain only safe edges. Moreover, these equivalence classes containing safe edges behave like single edges in the sense of the following lemma. For a proof of the lemma, see Lemma 14 in the full version.

▶ **Lemma 3.** *Let $S \in \mathfrak{S} \setminus \{S_u\}$ be an equivalence class of safe edges. It holds that:*

1. *S lies on a path in $\mathcal{A}$.*
2. *For any edge $f \in \mathcal{F}$, either S is completely contained in the fundamental cycle $C_{\mathcal{A}}(f)$ obtained by adding f to $\mathcal{A}$, or $S \cap C_{\mathcal{A}}(f) = \emptyset$.*
3. *If $(\mathcal{A} \setminus \{e\}) \cup \{f\}$ is feasible, and e belongs to equivalence class S, then $(\mathcal{A} \setminus S) \cup \{f\}$ is feasible. (This last property also trivially holds for $S = S_u$.)*

Charging. We can now give the bipartite-graph-based charging argument sketched above.

▶ **Theorem 4.** *Let $I = (V, E, \mathfrak{T}, d)$ be a Steiner Forest instance and let $\mathcal{F}$ be a feasible solution for I. Furthermore, let $\mathcal{A} \subseteq E$ be a feasible* tree *solution for I. Assume that $V[\mathcal{F}] = V[\mathcal{A}]$. Let $\Delta : \mathfrak{S} \to \mathbb{R}$ be a cost function that assigns a cost to all $S \in \mathfrak{S}$. Suppose that $\Delta(S) \leq d_f$*

for all pairs of $S \in \mathfrak{S} \setminus \{S_u\}$ and $f \in \mathcal{F}$ such that the cycle in $\mathcal{A} \cup \{f\}$ contains S. Then,

$$\sum_{S \in \mathfrak{S} \setminus \{S_u\}} \Delta(S) \leq \frac{7}{2} \cdot \sum_{f \in \mathcal{F}} d_f.$$

Proof. Construct a bipartite graph $H = (A \cup B, E(H))$ with nodes $A := \{a_S \mid S \in \mathfrak{S} \setminus \{S_u\}\}$ and $B := \{b_f \mid f \in \mathcal{F}\}$. Add an edge $\{a_S, b_f\}$ whenever f closes a cycle in $\mathcal{A}$ that contains S. By our assumption, if $\{a_S, b_f\} \in E(H)$ then $\Delta(S) \leq d_f$. Suppose that we can show that $\frac{7}{2} \cdot |N(X)| \geq |X|$ for all $X \subseteq A$, where $N(X) \subseteq B$ is the set of neighbors of nodes in X[2]. By a generalization of Hall's Theorem, this condition implies that there is an assignment $\alpha : E \to \mathbb{R}_+$ such that $\sum_{e \in \delta_H(a)} \alpha(e) \geq 1$ for all $a \in A$ and $\sum_{e \in \delta_H(b)} \alpha(e) \leq \frac{7}{2}$ for all $b \in B$. Hence

$$\begin{aligned}\sum_{S \in \mathfrak{S} \setminus \{S_u\}} \Delta(S) &\leq \sum_{S \in \mathfrak{S} \setminus \{S_u\}} \sum_{e \in \delta_H(a_S)} \alpha(e) \Delta(S) \\ &= \sum_{f \in \mathcal{F}} \sum_{e \in \delta_H(b_f)} \alpha(e) \Delta(S) \leq \sum_{f \in \mathcal{F}} \sum_{e \in \delta_H(b_f)} \alpha(e) d_f \leq \frac{7}{2} \sum_{f \in \mathcal{F}} d_f.\end{aligned}$$

It remains to show that $\frac{7}{2} \cdot |N(X)| \geq |X|$ for all $X \subseteq A$. To that aim, fix $X \subseteq A$ and define $\mathfrak{S}' := \{S \mid a_S \in X\}$. In a first step, contract all $e \in U := \bigcup_{S \in \mathfrak{S} \setminus \mathfrak{S}'} S$ in $\mathcal{A}$, and denote the resulting tree by $\mathcal{A}' := \mathcal{A} / U$.[3] Note that edges in each equivalence class are either all contracted or none are contracted. Also note that all unsafe edges are contracted, as $S_u \notin \mathfrak{S}'$. Apply the same contraction to $\mathcal{F}$ to obtain $\mathcal{F}' := \mathcal{F} / U$, from which we remove all loops and parallel edges. Notice that $\mathcal{A}'$ does not contain loops and parallel edges, since we contracted a subset of $\mathcal{A}$. Furthermore, $\mathcal{A}'$ is a tree, while $\mathcal{F}'$ can contain cycles.

Let $f \in \mathcal{F}'$. Since $\mathcal{A}'$ is a tree, f closes a cycle C in $\mathcal{A}'$ containing at least one edge $e \in \mathcal{A}'$. Denoting the equivalence class of e by S_e, observing that all edges in $\mathcal{A}'$ are safe, and using Lemma 3, statement 2, we get that cycle C contains S_e. Hence the node $b_f \in B$ corresponding to f belongs to $N(a_{S_e}) \subseteq N(X)$. Thus, $|N(X)| \geq |\mathcal{F}'|$ and it remains to show that $\frac{7}{2}|\mathcal{F}'| \geq |X|$.

We want to find a unique representative for each $a_S \in X$. So we select an arbitrary root vertex $r \in V[\mathcal{A}']$ and orient all edges in $\mathcal{A}'$ away from r. Every non-root vertex now has exactly one incoming edge. Every equivalence class $S \in \mathfrak{S}'$ consists only of safe edges, so it lies on a path. Consider the two well-defined *endpoints* which are the outermost vertices of S on this path. For at least one of them, the unique incoming edge must be an edge in S. We represent S by one of the endpoints which has this property and call this representative r_S. Let $R \subseteq V[\mathcal{A}']$ be the set of all representative nodes. Since every vertex has an unique incoming edge, $S \neq S'$ implies that $r_S \neq r_{S'}$. Hence $|R| = |\mathfrak{S}'| = |X|$. Moreover, let R_1 and R_2 be the representatives with degrees 1 and 2 in $\mathcal{A}'$, and L be the set of leaves of $\mathcal{A}'$. As the number of vertices of degree at least 3 in a tree is bounded by the number of its leaves, the number of representatives of degree at least 3 in $\mathcal{A}'$ is bounded by $|L|$. So $|X| \leq |R_1| + |R_2| + |L|$.

We now show that every $v \in R_2 \cup L$ is incident to an edge in $\mathcal{F}'$. First, consider any $v \in L$ and let e be the only edge in $\mathcal{A}'$ incident to v. As e is safe, there must be an edge $f \in \mathcal{F}'$

[2] Notice that $N(X)$ is a set of nodes, in contrast to $\delta(X)$, which is the set of edges leaving X.

[3] Formally, we define the graph $G[T]/e = (V[T]/e, T/e)$ for a tree T by $V[T]/e := V[T] \cup \{uv\} \setminus \{u, v\}$ and $T/e := T \setminus \delta(\{u, v\}) \cup \{\{w, uv\} \mid \{u, w\} \in T \vee \{v, w\} \in T\}$ for an edge $e = \{u, v\} \in E$, then set $G/U := G/e_1/e_2/\ldots/e_k$ for $U = \{e_1, \ldots, e_k\}$ and let T/U be the edge set of this graph. If $U \subseteq T$, then the contraction causes no loops or parallel edges, otherwise, we delete all loops or parallel edges.

incident to v. Now consider any $r_S \in R_2$ and let $e_1, e_2 \in \mathcal{A}'$ be the unique edges incident to r_S. Because r_S is the endpoint of the path corresponding to the equivalence class S, the edges e_1 and e_2 are not compatible. Hence there must be an edge $f \in \mathcal{F}'$ incident to r_S. Because R_2 and L are disjoint and every edge is incident to at most two vertices, we conclude that $|\mathcal{F}'| \geq (|R_2| + |L|)/2$. This implies that $|X| \leq |R_1| + |R_2| + |L| \leq 2L + |R_2| \leq 4|\mathcal{F}|$. We can get a slightly better bound below by showing that $|\mathcal{F}'| \geq \frac{2}{3}|R_1|$.

Let $\mathcal{C}$ be the set of connected components of $\mathcal{F}'$ in G/U. Let $\mathcal{C}' := \{T \in \mathcal{C} \mid |V[T] \cap R_1| \leq 2\}$ and $\mathcal{C}'' := \{T \in \mathcal{C} \mid |V[T] \cap R_1| > 2\}$. Note that no representative $r_S \in R_1$ is a singleton as every leaf of $\mathcal{A}'$ is incident to an edge of $\mathcal{F}'$. We claim that $|T| \geq |V[T] \cap R_1|$ for every $T \in \mathcal{C}'$. Assume by contradiction that this was not true and let $T \in \mathcal{C}'$ with $|T| < |V(T) \cap R_1|$. This means that $V[T] \cap R_1$ contains exactly two representatives $r_S, r_{S'} \in R_1$ and T contains only the edge $\{r_S, r_{S'}\}$. Let $e \in S$ and $e' \in S'$ be the edges of $\mathcal{A}'$ incident to r_S and $r_{S'}$, respectively. As e and e' are not compatible, there must be an edge $f \in \mathcal{F}'$ with exactly one endpoint in $\{r_S, r_{S'}\}$, a contradiction as this edge would be part of the connected component T. We conclude that $|T| \geq |V[T] \cap R_1|$ for every $T \in \mathcal{C}'$. Additionally, we have that $|T| \geq |V[T]| - 1 \geq \frac{2}{3}|V[T]|$ for all $T \in \mathcal{C}''$ as $|V[T]| > 2$. Therefore,

$$|\mathcal{F}'| = \sum_{T \in \mathcal{C}} |T| \geq \sum_{T \in \mathcal{C}'} |V[T] \cap R_1| + \sum_{T \in \mathcal{C}''} \frac{2}{3}|V[T] \cap R_1| \geq \frac{2}{3}|R_1|.$$

The three bounds together imply $|X| \leq |R_1| + |R_2| + |L| \leq \frac{3}{2}|\mathcal{F}'| + 2|\mathcal{F}'| = \frac{7}{2}|\mathcal{F}'|$. ◀

We obtain the following corollary of Theorem 4.

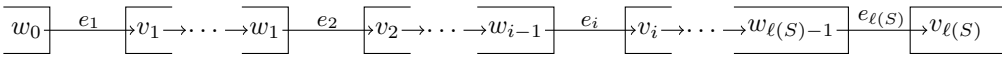
▶ **Corollary 5.** *Let $I = (V, E, \mathfrak{T}, d)$ be a Steiner Forest instance and let OPT be a solution for I that minimizes $d(OPT) = \sum_{e \in OPT} d_e$. Let $\mathcal{A} \subseteq E$ be feasible tree solution for I that does not contain inessential edges. Assume $V[\mathcal{A}] = V[OPT]$. If $\mathcal{A}$ is edge/edge and edge/set swap-optimal with respect to OPT and d, then it holds that $\sum_{e \in \mathcal{A}} d_e \leq (7/2) \cdot \sum_{e \in OPT} d_e$.*

Proof. Since there are no inessential edges, $S_u = \emptyset$. We set $\Delta(S) := \sum_{e \in S} d_e$ for all $S \in \mathfrak{S}$. Let $f \in \mathrm{OPT}$ be an edge that closes a cycle in $\mathcal{A}$ that contains S. Then, $(\mathcal{A} \setminus \{e\}) \cup \{f\}$ is feasible for any single edge $e \in S$ because it is still a tree. Statement 3 of Lemma 3 implies that $(\mathcal{A} \setminus S) \cup \{f\}$ is also feasible. Thus, we consider the swap that adds f and deletes S. It was not improving with respect to d, because $\mathcal{A}$ is edge/set swap-optimal with respect to edges from OPT and d. Thus, $\Delta(S) = \sum_{e \in S} d_e \leq d_f$, and we can apply Theorem 4 to obtain the result. ◀

4.2 An approximation guarantee for trees and ϕ

We now consider the case where a connected tree $\mathcal{A}$ is output by the algorithm when considering the edge/set swaps, but now with respect to the potential ϕ (instead of just the total length as in the previous section). These swaps may increase the number of components, which may have large widths, and hence edge/set swaps that are improving purely from the lengths may not be improving any more. This requires a more nuanced analysis, though relying on ideas we have developed in the previous section.

Here is the high-level summary of this section. Consider some equivalence class $S \in \mathfrak{S} \setminus \{S_u\}$ of safe edges, let $\ell(S)$ be the number of edges in S. The edges lie on a path, hence for an appropriate numbering $e_1, \ldots, e_{\ell(S)}$, the situation looks like this:



Notice that removing the $\ell(S)$ edges forms $\ell(S)+1$ connected components. We let In_S be set of the "inner" components (the ones containing $v_1, \ldots, v_{\ell(S)-1}$), and $\text{In}_{S'}$ be the inner components except the two with the highest widths. Just taking the definition of ϕ, and adding and subtracting the widths of these "not-the-two-largest" inner components, we get

$$\phi(\mathcal{A}) = w(\mathcal{A}) + \sum_{e \in S_u} d_e + \underbrace{\sum_{S \in \mathfrak{S} \setminus \{S_u\}} \Big(\sum_{i=1}^{\ell(S)} d_{e_i} - \sum_{K \in \text{In}_{S'}} w(K) \Big)}_{\leq 10.5 \cdot d(\mathcal{F}) \text{ (first proof step)}} + \underbrace{\sum_{S \in \mathfrak{S} \setminus \{S_u\}} \sum_{K \in \text{In}_{S'}} w(K)}_{\leq w(\mathcal{F}) \text{ (second proof step)}}.$$

As indicated above, the argument has two parts. For the first summation, look at the cycle created by adding edge $f \in \mathcal{F}$ to our solution $\mathcal{A}$. Suppose class S is contained in this cycle. We prove that edge/set swap optimality implies that $\sum_{i=1}^{\ell(S)} d_{e_i} - \sum_{K \in \text{In}_{S'}} w(K)$ is at most $3d_f$. (Think of this bound as being a weak version of the facts in the previous section, which did not have a factor of 3 but did not consider weights in the analysis.) Using this bound in Theorem 4 from the previous section gives us a matching that bounds the first summation by $3 \cdot (7/2) \cdot d(\mathcal{F})$. A couple of words about the proofs: the bound above follows from showing that three different swaps must be non-improving, hence the factor of 3. Basically, we break the above path into three at the positions of the two components of highest width, since for technical reasons we do not want to disconnect these high-width components. Details are in §7.1 in the full version.

For the second summation, we want to sum up the widths of all the "all-but-two-widest" inner components, over all these equivalence classes, and argue this is at most $w(\mathcal{F})$. This is where our notions of safe and compatible edges comes into play. The crucial observations are that (a) given the inner components corresponding to some class S, the edges of some class S' either avoid all these inner components, or lie completely within some inner component; (b) the notion of compatibility ensures that these inner components correspond to distinct components of $\mathcal{F}$, so we can get more width to charge to; and (c) since we don't charge to the two largest widths, we don't double-charge these widths. The details are in §7.3 in the full version.

5 In Which the Local Optimum may be a Forest

The main theorem. In the general case, both $\mathcal{A}$ and $\mathcal{F}$ may have multiple connected components. We assume that the distance function d is a metric. The first thing that comes to mind is to apply the previous analysis to the components individually. Morally, the main obstacle in doing so is in the proof of Theorem 4: There, we assume implicitly that no edge from $\mathcal{F}$ goes between connected components of $\mathcal{A}$.[4] This is vacuously true if $\mathcal{A}$ is a single tree, but may be false if $\mathcal{A}$ is disconnected. In the following, our underlying idea is to replace $\mathcal{F}$-edges that cross between the components of $\mathcal{A}$ by edges that lie within the components of $\mathcal{A}$, thereby re-establishing the preconditions of Theorem 4. We do this in a way that $\mathcal{F}$ stays feasible, and moreover, its cost increases by at most a constant factor. This allows us to prove that the local search has a constant locality gap.

Reducing to local tree optima. Suppose $\mathcal{F}$ has no inessential edges to start. Then we convert $\mathcal{F}$ into a collection of cycles (shortcutting non-terminals), losing a factor of 2 in

[4] More precisely, we need the slightly weaker condition that for each node $t \in V[\mathcal{A}]$, there is an $\mathcal{F}$-edge incident to t that does not leave the connected component of $\mathcal{A}$ containing t.

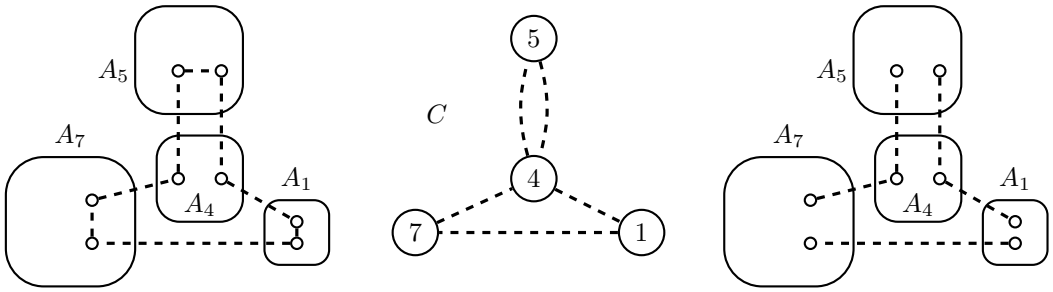


Figure 3 The charging argument with four components A_1, A_4, A_5 and A_7 of $\mathcal{A}$. The area of each component corresponds to its width. *On the left.* A cycle in $\mathcal{F}$. *In the middle.* The corresponding circuit (non-simple cycle) in $G_{\mathcal{A}}$. *On the right.* A suitable decomposition into connecting moves.

the cost. Now observe that each "offending" $\mathcal{F}$-edge (i.e., one that goes between different components of $\mathcal{A}$) must be part of a path P in $\mathcal{F}$ that connects some $s, \bar{s}$, and hence starts and ends in the same component of $\mathcal{A}$. This path P may connect several terminal pairs, and for each such pair $s, \bar{s}$, there is a component of $\mathcal{A}$ that contains s and $\bar{s}$. Thus, P could be replaced by direct connections between $s, \bar{s}$ within the components of $\mathcal{A}$. This would get rid of these "offending" edges, since the new connections would stay within components of $\mathcal{A}$. The worry is, however, that this replacement is too expensive. We show how to use connecting tree moves to bound the cost of the replacement.

Consider one cycle from $\mathcal{F}$, regarded as a circuit C in the graph $G_{\mathcal{A}}$ where the connected components $A_1, \ldots, A_p$ of $\mathcal{A}$ are shrunk to single nodes, i.e., C consists of offending edges. The graph $G_{\mathcal{A}}$ might contain parallel edges and C might have repeated vertices. So C is a circuit, meaning that it is a potentially non-simple cycle, or, in other words, a Eulerian multigraph. The left and middle of Figure 3 are an example.

Index the A_j's such that $w(A_1) \leq \cdots \leq w(A_p)$ and say that node j in $G_{\mathcal{A}}$ corresponds to A_j. Suppose C visits the nodes $v_1, \ldots, v_{|C|}, v_1$ (where several of these nodes may correspond to the same component A_j) and that component A_j is visited n_j times by C. In the worst case, we may need to insert n_j different $s, \bar{s}$ connections into component A_j of $\mathcal{A}$, for all j. The key observation is that the total cost of our direct connections is at most $\sum_{i=1}^{|C|} n_i w(A_i)$. We show how to pay for this using the length of C.[5]

To do so, we use optimality with respect to all moves, in particular connecting moves. The idea is simple: We cut C into a set of trees that each define a valid connecting move. For each tree, the connecting move optimality bounds the widths of some components of $\mathcal{A}$ by the length of the tree. E.g., $w(A_1)+w(A_4)$ is at most the length of the tree connecting A_1, A_4, A_5 in Figure 3. Observe that we did not list $w(A_5)$: Optimality against a connecting move with tree T relates the length of T to the width of all the components that T connects, *except* for the component with maximum width. We say a tree *pays* for A_j if it hits A_j, and also hits another A_j of higher width. So we need three properties: (a) the trees should collectively pack into the edges of the Eulerian multigraph C, (b) each tree hits each component A_j at most once, and (c) the number of trees that pay for A_j is at least n_j.

Assume that we found such a tree packing. For circuit C, if $A_{j^\star}$ is the component with greatest width hit by C, then using connecting move optimality for all the trees shows that

$$\sum_{j:A_j \text{ hit by } C, j \neq j^\star} n_j \, w(A_j) \leq d(C).$$

[5] We also need to take care of the additional width of the modified solution, but this is the easier part.

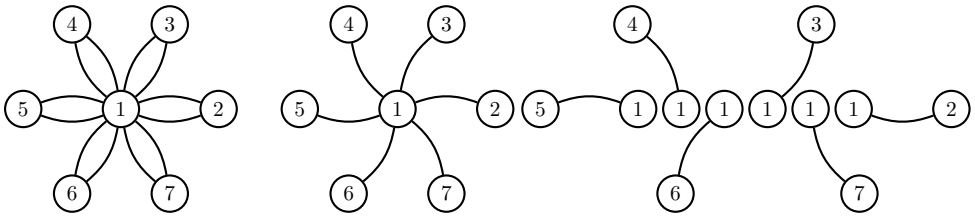


Figure 4 A flower graph. Even though the graph is a non-simple cycle, we can easily decompose it into trees that pay for each $j \neq 6$ at least n_j times (1 is payed for $7 = n_1 + 1$ times).

In fact, even if we have c-approximate connection-move optimality, the right-hand side just gets multiplied by c. But what about $n_{j^\star} w(A_{j^\star})$? We can cut C into sub-circuits, such that each subcircuit C' hits $A_{j^\star}$ exactly once. To get this one extra copy of $w(A_{j^\star})$, we use path/set swap optimality which tells us that the missing connection cannot be more expensive than the length of C. Thus, collecting all our bounds (see Lemma 36 in the full version), adding all the extra connections to $\mathcal{F}$ increases the cost to at most $2(1+c)d(\mathcal{F})$: the factor 2 to make $\mathcal{F}$ Eulerian, $(1+c)$ to add the direct connections, using c-approximate optimality with respect to connecting moves and optimality with respect to path/set swaps. §B.2 in the full version discusses that $c \leq 2$.

Now each component A_j of $\mathcal{A}$ can be treated separately, i.e., we can use Corollary 2 on each A_j and the portion of $\mathcal{F}$ that falls into A_j. By combining the conclusions for all connected components, we get that

$$d(\mathcal{A}') \overset{\text{Cor. 2}}{\leq} 10.5 d(\mathcal{F}') + w(\mathcal{F}') \leq 11.5 d(\mathcal{F}') \leq 23(1+c) \cdot d(\mathcal{F}) \leq 69 \cdot d(\mathcal{F})$$

for any feasible solution $\mathcal{F}$. This proves Theorem 1.

Obtaining a decomposition into connecting moves. It remains to show how to take C and construct the set of trees. If C had no repeated vertices (it is a simple cycle) then taking a spanning subtree would suffice. And even if C has some repeated vertices, decomposing it into suitable trees can be easy: E.g., if C is the "flower" graph on n vertices, with vertex 1 having two edges to each vertex $2, \ldots, n$. Even though 1 appears multiple times, we find a good decomposition (see Figure 4). Observe, however, that breaking C into simple cycles and than doing something on each simple cycle would not work, since it would only pay 1 multiple times and none of the others.

The flower graph has a property that is a generalization of a simple cycle: We say that C is minimally guarded if (a) the largest vertex is visited only once (b) between two occurrences of the same (non-maximal) vertex, there is at least one larger number. The flower graph and the circuit at the top of Figure 5 have this property. Indeed, every minimally guarded circuit can be decomposed suitably. The full algorithm is provided as Algorithm 1 in the full version. It iteratively finds trees that pay for all (non maximal) j with $j \leq z$ for increasing z. Figure 5 shows how the set of trees $\mathfrak{M}_5$ is converted into $\mathfrak{M}_6$ in order to pay for all occurrences of 6. Intuitively, we look where 6 falls into the trees in $\mathfrak{M}_5$. Up to one occurrence can be included in a tree. If there are more occurrences, the tree has to be split into multiple trees appropriately. §8.1.2 in the full version contains the details of the algorithm and its correctness.

Finally, we go from minimally guarded circuits to arbitrary C by extracting subcircuits in a recursive fashion (see Lemma 35 in the full version).

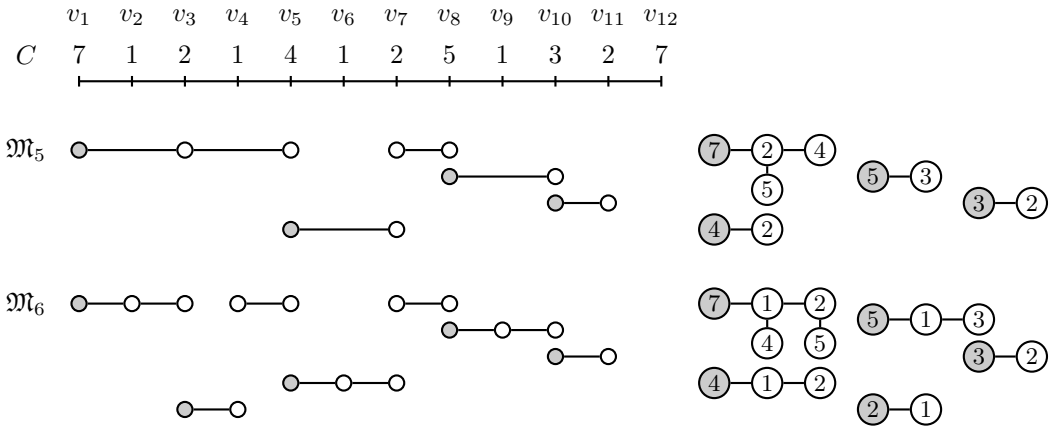


Figure 5 Two iterations of an example run of the Algorithm 1 in the full version.

References

1 Ajit Agrawal, Philip N. Klein, and R. Ravi. When trees collide: An approximation algorithm for the generalized steiner problem on networks. *SIAM J. Comput.*, 24(3):440–456, 1995. doi:10.1137/S0097539792236237.

2 Sara Ahmadian, Zachary Friggstad, and Chaitanya Swamy. Local-search Based Approximation Algorithms for Mobile Facility Location Problems. In *Proceedings of the Twenty-Fourth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '13, pages 1607–1621. SIAM, 2013. URL: http://dl.acm.org/citation.cfm?id=2627817.2627932.

3 Paola Alimonti. New local search approximation techniques for maximum generalized satisfiability problems. In Maurizio A. Bonuccelli, Pierluigi Crescenzi, and Rossella Petreschi, editors, *Algorithms and Complexity, Second Italian Conference, CIAC '94, Rome, Italy, February 23-25, 1994, Proceedings*, volume 778 of *Lecture Notes in Computer Science*, pages 40–53. Springer, 1994. doi:10.1007/3-540-57811-0_5.

4 Vijay Arya, Naveen Garg, Rohit Khandekar, Adam Meyerson, Kamesh Munagala, and Vinayaka Pandit. Local search heuristics for k-median and facility location problems. *SIAM J. Comput.*, 33(3):544–562, 2004. doi:10.1137/S0097539702416402.

5 Norman Biggs. Constructions for cubic graphs with large girth. *The Electronic Journal of Combinatorics*, 5(1):A1:1–A1:25, 1998.

6 Sergio Cabello and David Gajser. Simple ptas's for families of graphs excluding a minor. *Discrete Applied Mathematics*, 189:41–48, 2015. doi:10.1016/j.dam.2015.03.004.

7 Chandra Chekuri and F. Bruce Shepherd. Approximate integer decompositions for undirected network design problems. *SIAM J. Discrete Math.*, 23(1):163–177, 2008. doi:10.1137/040617339.

8 Ho-Lin Chen, Tim Roughgarden, and Gregory Valiant. Designing network protocols for good equilibria. *SIAM J. Comput.*, 39(5):1799–1832, 2010. doi:10.1137/08072721X.

9 Vincent Cohen-Addad, Philip N. Klein, and Claire Mathieu. Local search yields approximation schemes for k-means and k-median in euclidean and minor-free metrics. In *Proceedings of the 57th Annual Symposium on Foundations of Computer Science*, 2016. to appear.

10 Vincent Cohen-Addad and Claire Mathieu. Effectiveness of local search for geometric optimization. In Lars Arge and János Pach, editors, *31st International Symposium on Computational Geometry, SoCG 2015, June 22-25, 2015, Eindhoven, The Netherlands*, volume 34 of *LIPIcs*, pages 329–343. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2015. doi:10.4230/LIPIcs.SOCG.2015.329.

11 Zachary Friggstad, Mohsen Rezapour, and Mohammad R. Salavatipour. Local search yields a PTAS for k-means in doubling metrics. In *Proceedings of the 57th Annual Symposium on Foundations of Computer Science*, volume abs/1603.08976, 2016. to appear.

12 Martin Fürer and Balaji Raghavachari. Approximating the minimum-degree steiner tree to within one of optimal. *J. Algorithms*, 17(3):409–423, 1994. doi:10.1006/jagm.1994.1042.

13 Naveen Garg. Saving an epsilon: a 2-approximation for the k-mst problem in graphs. In Harold N. Gabow and Ronald Fagin, editors, *Proceedings of the 37th Annual ACM Symposium on Theory of Computing, Baltimore, MD, USA, May 22-24, 2005*, pages 396–402. ACM, 2005. doi:10.1145/1060590.1060650.

14 Naveen Garg, 2016. Personal Communication.

15 Michel X. Goemans and David P. Williamson. A general approximation technique for constrained forest problems. *SIAM J. Comput.*, 24(2):296–317, 1995. doi:10.1137/S0097539793242618.

16 Albert Gu, Anupam Gupta, and Amit Kumar. The power of deferral: maintaining a constant-competitive steiner tree online. 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 525–534. ACM, 2013. doi:10.1145/2488608.2488674.

17 Anupam Gupta and Amit Kumar. Online steiner tree with deletions. In Chandra Chekuri, editor, *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2014, Portland, Oregon, USA, January 5-7, 2014*, pages 455–467. SIAM, 2014. doi:10.1137/1.9781611973402.34.

18 Anupam Gupta and Amit Kumar. Greedy Algorithms for Steiner Forest. In Ronitt Rubinfeld and Rocco Servedio, editors, *Proceedings of the Forty-seventh Annual ACM Symposium on Theory of Computing*, STOC '15, pages 871–878. ACM, 2015.

19 Keld Helsgaun. An effective implementation of the lin-kernighan traveling salesman heuristic. *European Journal of Operational Research*, 126(1):106–130, 2000. doi:10.1016/S0377-2217(99)00284-2.

20 Makoto Imase and Bernard M. Waxman. Dynamic Steiner tree problem. *SIAM J. Discrete Math.*, 4(3):369–384, 1991.

21 Kamal Jain. A factor 2 approximation algorithm for the generalized steiner network problem. *Combinatorica*, 21(1):39–60, 2001. doi:10.1007/s004930170004.

22 Tapas Kanungo, David M. Mount, Nathan S. Netanyahu, Christine D. Piatko, Ruth Silverman, and Angela Y. Wu. A local search approximation algorithm for k-means clustering. *Comput. Geom.*, 28(2-3):89–112, 2004. doi:10.1016/j.comgeo.2004.03.003.

23 Sanjeev Khanna, Rajeev Motwani, Madhu Sudan, and Umesh V. Vazirani. On syntactic versus computational views of approximability. *SIAM J. Comput.*, 28(1):164–191, 1998. doi:10.1137/S0097539795286612.

24 Jochen Könemann, Stefano Leonardi, and Guido Schäfer. A Group-Strategyproof Mechanism for Steiner Forests. In *Proceedings of the Sixteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '05, pages 612–619. Society for Industrial and Applied Mathematics, 2005. doi:10.1.1.126.4369.

25 Jochen Könemann, Stefano Leonardi, Guido Schäfer, and Stefan H. M. van Zwam. A group-strategyproof cost sharing mechanism for the steiner forest game. *SIAM J. Comput.*, 37(5):1319–1341, 2008. doi:10.1137/050646408.

26 Madhukar R. Korupolu, C. Greg Plaxton, and Rajmohan Rajaraman. Analysis of a local search heuristic for facility location problems. *J. Algorithms*, 37(1):146–188, 2000. doi:10.1006/jagm.2000.1100.

27 Jakub Lacki, Jakub Ocwieja, Marcin Pilipczuk, Piotr Sankowski, and Anna Zych. The power of dynamic distance oracles: Efficient dynamic algorithms for the steiner tree. 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 11–20. ACM, 2015. doi:10.1145/2746539.2746615.

28 S. Lin and Brian W. Kernighan. An effective heuristic algorithm for the traveling-salesman problem. *Operations Research*, 21(2):498–516, 1973. doi:10.1287/opre.21.2.498.

29 Hsueh-I Lu and R. Ravi. The Power of Local Optimization: Approximation Algorithms for Maximum-leaf Spanning Tree. In *In Proceedings, Thirtieth Annual Allerton Conference on Communication, Control and Computing*, pages 533–542, 1996.

30 Nicole Megow, Martin Skutella, José Verschae, and Andreas Wiese. The power of recourse for online MST and TSP. In Artur Czumaj, Kurt Mehlhorn, Andrew M. Pitts, and Roger Wattenhofer, editors, *Automata, Languages, and Programming - 39th International Colloquium, ICALP 2012, Warwick, UK, July 9-13, 2012, Proceedings, Part I*, volume 7391 of *Lecture Notes in Computer Science*, pages 689–700. Springer, 2012. doi:10.1007/978-3-642-31594-7_58.

31 Martin Pál, Éva Tardos, and Tom Wexler. Facility location with nonuniform hard capacities. In *42nd Annual Symposium on Foundations of Computer Science, FOCS 2001, 14-17 October 2001, Las Vegas, Nevada, USA*, pages 329–338. IEEE Computer Society, 2001. doi:10.1109/SFCS.2001.959907.

32 Lukás Polácek and Ola Svensson. Quasi-polynomial local search for restricted max-min fair allocation. *ACM Trans. Algorithms*, 12(2):13:1–13:13, 2016. doi:10.1145/2818695.

33 Roberto Solis-Oba. 2-approximation algorithm for finding a spanning tree with maximum number of leaves. In Gianfranco Bilardi, Giuseppe F. Italiano, Andrea Pietracaprina, and Geppino Pucci, editors, *Algorithms - ESA '98, 6th Annual European Symposium, Venice, Italy, August 24-26, 1998, Proceedings*, volume 1461 of *Lecture Notes in Computer Science*, pages 441–452. Springer, 1998. doi:10.1007/3-540-68530-8_37.

34 David P Williamson and David B Shmoys. *The design of approximation algorithms*. Cambridge university press, 2011.

A Notes on simpler local search algorithms

A.1 Adding an edge and removing a constant number of edges

Let ℓ and $k < \ell$ be integers and consider Figure 6. Notice that adding a single edge and removing k edges does not improve the solution. However, the current solution costs more than ℓ^2/k and the optimal solution costs less than 2ℓ, which is a factor of $\ell/(2k)$ better.

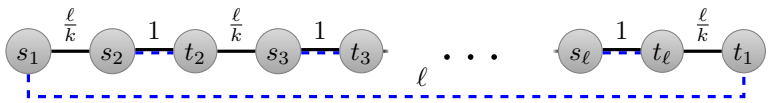


Figure 6 A bad example for edge/set swaps that remove a constant number of edges.

A.2 Regular graphs with high girth and low degree

Assume that G is a degree-3 graph with girth $g = c \log n$ like the graph used in Chen et al. [8]. Such graphs can be constructed, see [5]. Select a spanning tree $\mathcal{F}$ in G which will be the optimal solution. Let E' be the non-tree edges, notice that $|E'| \geq n/2$, and let M be a maximum matching in E'. Because of the degrees, we know that $|M| \geq n/10$. The endpoints of the edges in M form the terminal pairs $\mathfrak{T}$. Set the length of all edges in $\mathcal{F}$ to 1 and the

length of the remaining edges to $g/4$. The solution $\mathcal{F}$ is feasible and costs $n-1$. The solution M costs $\Omega(\log n)$.

Assume we want to remove an edge $e = \{v, w\} \in M$ and our swap even allows us to add a path to reconnect v and w (in the graph where $M \backslash \{e\}$ is contracted). Let P be such a path. Since M is a matching, at most every alternating edge on P is in M. Thus, we have to add $|P|/2 - 1 \geq g/2 - 1$ edges of length one at a total cost that is larger than the cost $g/4$ of e. Thus, no d-improving swap of this type exists (note that, in particular, path/set swaps are not d-improving for M). As a consequence, any oblivious local search with constant locality gap needs to sport a move that removes edges from multiple components of the current solution. In order to restrict to local moves that only remove edges from a single component, we therefore introduced the potential ϕ.

Quasipolynomial Representation of Transversal Matroids with Applications in Parameterized Complexity

Daniel Lokshtanov[1], Pranabendu Misra[2], Fahad Panolan[3], Saket Saurabh[4], and Meirav Zehavi[5]

1 **University of Bergen, Bergen, Norway**
daniello@ii.uib.no
2 **The Institute of Mathematical Sciences, HBNI, Chennai, India**
pranabendu@imsc.res.in
3 **University of Bergen, Bergen, Norway**
fahad.panolan@ii.uib.no
4 **University of Bergen, Bergen, Norway and The Institute of Mathematical Sciences, HBNI, Chennai, India**
saket@imsc.res.in
5 **Ben-Gurion University, Beersheba, Israel**
meiravze@bgu.ac.il

Abstract

Deterministic polynomial-time computation of a representation of a transversal matroid is a longstanding open problem. We present a deterministic computation of a so-called union representation of a transversal matroid in time quasipolynomial in the rank of the matroid. More precisely, we output a collection of linear matroids such that a set is independent in the transversal matroid if and only if it is independent in at least one of them. Our proof directly implies that if one is interested in preserving independent sets of size at most r, for a given $r \in \mathbb{N}$, but does not care whether larger independent sets are preserved, then a union representation can be computed deterministically in time quasipolynomial in r. This consequence is of independent interest, and sheds light on the power of union representation.

Our main result also has applications in Parameterized Complexity. First, it yields a fast computation of representative sets, and due to our relaxation in the context of r, this computation also extends to (standard) truncations. In turn, this computation enables to efficiently solve various problems, such as subcases of subgraph isomorphism, motif search and packing problems, in the presence of color lists. Such problems have been studied to model scenarios where pairs of elements to be matched may not be identical but only similar, and color lists aim to describe the set of compatible elements associated with each element.

1998 ACM Subject Classification I.1.2 Algorithms, F.2.2 Nonnumerical Algorithms and Problems

Keywords and phrases Travserval matroid, matroid representation, union representation, representative set

Digital Object Identifier 10.4230/LIPIcs.ITCS.2018.32

1 Introduction

Matroids are widely-studied mathematical objects. In the context of computer science, these objects are of particular importance to algorithm design, combinatorial optimization and

computational complexity. Specifically, from the viewpoint of algorithm design, analysis of these objects often leads to the discovery of algorithmic meta theorems. Such theorems unify classical results such as polynomial-time solvability of a wide-variety of problems as central as Minimum Weight Spanning Tree and Perfect Matching. In fact, if a problem admits a greedy algorithm, then it can be embedded in a matroid – solutions are thus associated with maximum independent sets in the matroid. Recently, matroids also stand in the forefront of studies of approximation algorithms, parameterized algorithms and kernels.

A matroid is a pair $M = (E, \mathcal{I})$, where $\mathcal{I}$ is a family of subsets of E (called *independent sets*), that satisfy three conditions called *matroid axioms* (see Section 2). As the size of $\mathcal{I}$ can be exponential in the size of E, the explicit listing of all independent sets is often rendered prohibitive. Then, it is necessary to have an *independence oracle* that, given a subset I of E, determines (in polynomial time) whether I is present in $\mathcal{I}$. For a wide class of matroids, known as *linear matroids*, such oracle is given by a matrix called a *representation*. Roughly speaking, the columns of the matrix are in bijection with the elements in E, and a set of columns is linearly independent if and only if the set of corresponding elements is independent. Unfortunately, for several important linear matroids, efficient computations of the desired representations are not known.

Specific well-known classes of matroids are those of uniform matroids, partition matroids, graphic matroids, cographic matroids, transversal matroids and gammoids. A common property of all of these classes is that all of them are contained in the wider class of linear matroids. However, for the last two classes in this list a polynomial-time deterministic computation of a representation is not known. Developing such a computation is a longstanding open problem. In this paper, we make significant progress towards the resolution of this question. We remark that as the dual of a transversal matroid is a gammoid and vice versa, a polynomial-time computation of a representation for one also yields such a computation for the other [13]. We specifically focus on the class of transversal matroids. Formally, a *transversal matroid* is a matroid derived from a bipartite graph G with a fixed bipartition (A, B) as follows: the ground set E is simply A, and a subset $X \subseteq A$ is independent if and only if G has a matching that saturates it. Matching constraints are ubiquitous in problems arising in all fields of research. Indeed, such constraints model scenarios where some set of objects relevant to our solution should be partitioned into pairs. Transversal matroids are precisely the translation of these constraints (in the bipartite setting) into the language of matroids.

To tackle the question above, we introduce the notion of *union representation*, which we believe to be worthy of independent study. For algorithmic purposes, such representation is as good as standard representation, given that the number of members in the union is small, and it may be useful also in cases where not only an efficient computation of a standard representation is not known, but a standard representation simply does not exist. Before we further discuss the power of this notion, let us first present it properly. Roughly speaking, a union representation of a matroid $M = (E, \mathcal{I})$ is a collection of matrices such that a subset X of E is independent in M if and only if for at least one of the matrices, the set of columns corresponding to X is linearly independent. Standard representation is precisely union representation where the size of the collection is one. While only linear matroids admit standard representations, note that all matroids admit union representations: to see this, for every base of the matroid, create one matrix with a set of linearly independent columns corresponding to the base, and vectors having only 0 entries as the rest of the columns. However, this procedure may create a huge number of matrices, and in order to make the

notion of union representation relevant to algorithmic purposes, we desire the number of matrices to be as small as possible.

In this work, we present a deterministic computation of a union representation of a transversal matroid consisting of a quasipolynomial (in the rank of the matroid) number of matrices in quasipolynomial time. Prior to our work, the fastest such computation was only slightly better than trivial brute-force. More precisely, Misra et al. [14] showed that given a bipartite graph G with a fixed bipartition (A, B), a representation of the transversal matroid can be computed deterministically in (exponential) time $\binom{|A|}{r}|A|^{\mathcal{O}(1)}$ where r is the rank of the matroid, which equals the maximum size of a matching in G. In this context, it is important to note that a randomized polynomial-time algorithm to compute a representation of a transversal matroid is well known (see, e.g., [13, 17]). Here, randomization means that with some (low) probability, the algorithm may output a matrix that is not a representation of the matroid. This algorithm utilizes the Schwartz-Zippel lemma [2, 22, 25], and hence it is inherently randomized. The above mentioned trivial brute-force, which runs in time $2^{\mathcal{O}(|A|^2|B|)}$ (see [14]), refers to a loop through all choices made by the randomized algorithm.

Our technique builds upon recent powerful derandomization tools, particularly a construction given by Fenner et al. [4]. This construction is essentially a (quasipolynomial-time) derandomization of a special case of the isolation lemma [15], namely, the isolation of a perfect matching (if one exists). Roughly speaking, given a positive integer $n \in \mathbb{N}$, the construction is a collection of $2^{\mathcal{O}(\log^2 n)}$ weight functions such that for any bipartite graph G on $2n$ vertices that has a perfect matching, there exists a weight function w in the collection such that, when the edges of G are assigned weights according to w,[1] G has a unique perfect matching of minimum weight. Fenner et al. [4] utilized this construction to prove that Perfect Matching on bipartite graphs is in quasi-NC. Soon after this paper was published, significant generalizations of it followed [10, 7, 24]. Briefly, Gurjar et al. [10] showed that Linear Matroid Intersection is in quasi-NC, Goldwasser et al. [7] showed that Perfect Matching on bipartite graphs is in pseudo-deterministic NC, and Svensson et al. [24] showed that Perfect Matching on general graphs is in quasi-NC.

We introduce the above derandomization tools (specifically, the construction of [4]) to the context of representation, incorporating a flavor of Parameterized Complexity to the representation itself. Indeed, the computation we derive can be viewed as a (quasipolynomial-time) fixed-parameter tractable algorithm with respect to r. Consequently, we also introduce these tools to Parameterized Complexity, as we observe that our union representation computation can be incorporated in the method of *representative sets* by Fomin et al. [5] (see below). On a high-level, our proof consists of the use of a splitter [16] to "color" vertices of the input graph G using "small" integers. Then, we view a weight function not as a function that assigns weights to edges of some specific graph, but as a function that assigns weights to pairs of integers (that are simultaneously associated to a possibly exponential number of induced subgraphs of G). This allows us to compose splitters functions with weight functions. For each composition, we are then able to define a matrix, in the spirit of [15], that is one member of our union representation. The crux of the correctness is that both a splitter and a collection of [4] are *universal* in the sense that neither of them is tailored to a specific input graph to highlight structures of that graph (such as a perfect matching, in the case of the collection). Specifically, the same splitter and collection are relevant simultaneously to an exponential number of graphs that are of interest to our purpose, namely, all the induced

[1] Informally, each weight function assigns weights to the edges of a complete bipartite graph on $2n$ vertices that has a perfect matching, and the assignment of weights to the edges of G can be derived from this.

subgraphs of our input graph G that are sufficient to witness the independence of all sets in $\mathcal{I}$ (the family of independent sets of the transversal matroid at hand).

Our main theorem is in fact slightly stronger than described above. Suppose that for algorithmic purposes, we would like to obtain a union representation of a structure where all independent sets are also independent in our transversal matroid of interest (i.e., we do not introduce false independent sets), and where all independent sets *of size up to* k in our transversal matroid are also independent in our structure. In other words, we are pleased with a structure that may "throw away" some large independent sets. For this purpose, we introduce the appropriate notion of a structure that is in fact a weakening of the well-known notion of a k-truncation of a matroid. Such a structure is useful for applications to problems where solution size is at most k, which can be significantly smaller than the rank of the matroid. Then, our computation of union representation runs in time quasipolynomial in k rather than the rank, and thereby enables the design of parameterized algorithms with respect to k.

Applications. Our main result also has applications in Parameterized Complexity. First, it yields a fast computation of representative sets (integrated in the framework of [5, 12]). Formally, given a matroid $M = (E, \mathcal{I})$ and a family $\mathcal{S}$ of subsets of E, a subfamily $\widehat{\mathcal{S}} \subseteq \mathcal{S}$ is *q-representative* for $\mathcal{S}$ if the following holds: for every set $Y \subseteq E$ of size at most q, if there is a set $X \in \mathcal{S}$ disjoint from Y with $X \cup Y \in \mathcal{I}$, then there is a set $\widehat{X} \in \widehat{\mathcal{S}}$ disjoint from Y with $\widehat{X} \cup Y \in \mathcal{I}$. Fomin et al. [5, 12] showed that if one is given a representation of the matroid M, then small representative sets can be computed efficiently (see Section 4). This computation has led, since its introduction in 2013, to the development of dozens of parameterized algorithms that are the current state-of-the-art for their respective problems. We observe that our computation of a union representation of a transversal matroid can be composed with the algorithm of Fomin et al. [5, 12] to obtain small representative sets efficiently also with respect to transversal matroids. As matching constraints naturally arise in various scenarios, we find it important that the powerful tool of representative sets can now be employed to handle them as well.

To illustrate the usefulness of the computation above using a simple didactic example, we consider the List k-Path problem, informally defined as follows. The input consists of an undirected graph G such that each of its vertices has its own list of *compatible* colors, and the objective is to determine whether G has a (simple) path on k vertices such that one can assign a compatible color to each vertex on this path to make it colorful. This problem is a natural generalization of the classical k-Path problem to the presence of color lists, and it was studied (in a slightly more general form) in [18] in the setting of randomized algorithms. Previously, to solve this problem using representative sets, we remark that one would have to use the direct sum of two uniform matroids, one to ensure distinctness of vertices and one to ensure distinctness of colors, which would result in running time $\mathcal{O}(6.86^k \cdot n^{\mathcal{O}(1)})$ – this would be, to the best of our knowledge, the state-of-the-art. We show that simply by using a transversal matroid rather than a direct sum of two uniform matroids, one obtains a running time of $\mathcal{O}(5.18^k \cdot n^{\mathcal{O}(1)})$.

We stress that the choice of List k-Path is only done for illustrative purposes. Indeed, by *only* considering transversal matroids rather than direct sums of two uniform matroids, a wide variety of problems can now immediately be solved in time $\mathcal{O}(5.18^k \cdot n^{\mathcal{O}(1)})$ rather than $\mathcal{O}(6.86^k \cdot n^{\mathcal{O}(1)})$ in the presence of color lists. Such problems have been studied to model scenarios where pairs of elements to be matched may not be identical but only similar, and color lists aim to describe the set of compatible elements associated with each element. This

includes, for example, graph problems such as subcases of subgraph isomorphism, which are of relevance (in the presence of color lists) to bioinformatics [19, 21, 23, 3]. We remark that this approach is applicable not only to graph problems, but also to various packing and matching problems (such as those studied in [8]) in the presence of list colors. The Graph Motif problem, in particular, was extensively studied also in the presence of color lists (see [20, 1, 9, 11] and references therein), where the previous fastest deterministic algorithm run in time $\mathcal{O}(6.86^k \cdot n^{\mathcal{O}(1)})$ [20]. By simply using a transversal matroid rather than a direct sum of two uniform matroids in the algorithm of [20], we immediately derive an improved running time of $\mathcal{O}(5.18^k \cdot n^{\mathcal{O}(1)})$.

Proofs of results marked by an asterisk ('*') are omitted.

2 Preliminaries

Given $t \in \mathbb{N}$, we use $[t]$ as a shorthand for $\{1, 2, \ldots, t\}$. Given a function $f : A \to B$ and a subset $A' \subseteq A$, we denote $f(A') = \{f(a) : a \in A'\}$, and we define $f|_{A'}$ as the restriction of f to A'. We slightly abuse notation, and given a function $g : A \to \mathbb{N}$, called a *weight function*, and a subset $A' \subseteq A$, we denote $g(A') = \sum_{a \in A'} g(a)$. Whenever we refer to a function that is a weight function, we use the second notation.

Given a graph G, we say that (A, B) is a *vertex bipartition* of G if it is a partition of $V(G)$ such that A and B are independent sets. Moreover, we say that G is a *bipartite graph* if it has a vertex bipartition. A *matching* μ is a family of pairwise-disjoint subsets of $E(G)$.

2.1 Matroids

Let us begin by presenting the definition of an independence system.

▶ **Definition 1** (Independence System). A pair $P = (\mathcal{I}, E)$, where E is a ground set and $\mathcal{I}$ is a family of subsets of E (called *independent sets*), is an *independence system* if it satisfies the following conditions:
(I1) $\phi \in \mathcal{I}$.
(I2) If $X \subseteq Y$ and $Y \in \mathcal{I}$, then $X \in \mathcal{I}$.

A matroid is an independence system with an additional property, formally defined as follows.

▶ **Definition 2** (Matroid). An independence system $M = (\mathcal{I}, E)$ is a *matroid* if it satisfies the following condition:
(I3) If $X, Y \in \mathcal{I}$ and $|X| < |Y|$, then there exists $e \in (Y \setminus X)$ such that $X \cup \{e\} \in \mathcal{I}$.
The *rank* of M is the maximum size of a set in $\mathcal{I}$.

We remark that conditions (I1), (I2) and (I3) are called *matroid axioms*. We say that two independence systems $P = (E, \mathcal{I})$ and $P' = (E', \mathcal{I}')$ are *isomorphic* if there exists a bijection $\varphi : E \to E'$ such that for every $X \subseteq E$, $X \in \mathcal{I}$ if and only if $\varphi(X) \in \mathcal{I}'$. In this paper, we are specifically interested in transversal matroids, defined as follows.

▶ **Definition 3** (Transversal Matroid). Let G be a bipartite graph with a fixed vertex bipartition (A, B). The *transversal matroid* M of G is the pair $(A, \mathcal{I})$ where $\mathcal{I}$ is the family that consists of every subset $X \subseteq A$ such that there exists a matching that saturates X.

It is well-known that a transversal matroid is indeed a matroid [17]. Having a *representation* of a matroid, given by a matrix that compactly encodes the matroid, is a central component in many algorithmic applications. Matroids having a representation are called *linear*, as formally defined below.

▶ **Definition 4.** Let A be a matrix over an arbitrary field $\mathbb{F}$, and let C be the set of columns of A. The *matroid represented by* A is the pair $M = (C, \mathcal{I})$, where a subset $X \subseteq C$ belongs to $\mathcal{I}$ if and only if the columns in X are linearly independent over $\mathbb{F}$.

It is well-known that the pair $M = (C, \mathcal{I})$ in Definition 4 indeed defines a matroid [17].

▶ **Definition 5** (Linear Matroid, Representation)**.** A matroid $M = (E, \mathcal{I})$ is a *linear matroid* if there exists a matrix A, called a *representation* of M, such that M and the matroid represented by A are isomorphic. Furthermore, M is *representable over a field* $\mathbb{F}$ if it has a representation A over $\mathbb{F}$.

We introduce a generalization of the concepts above, resulting in the notion of *union representation*, which is sufficient for many algorithmic purposes and may be of independent interest.

▶ **Definition 6.** Let $A_1, A_2, \ldots, A_t$ be t matrices over an arbitrary field $\mathbb{F}$, let E be a ground set, and for all $i \in [t]$, let $\varphi_i : E \to C_i$ be a bijection, where C_i is the set of columns of A_i. The *independence system represented by* $(E, \{A_i, \varphi_i\}|_{i\in[t]})$ is given by $P = (E, \mathcal{I})$, where a subset $X \subseteq E$ belongs to $\mathcal{I}$ if and only if there exists $i \in [t]$ such that the columns in $\varphi_i(X)$ are linearly independent over $\mathbb{F}$.

It should be clear that $P = (E, \mathcal{I})$ in Definition 6 is indeed an independence system, but we remark that it might not be a matroid since it may not satisfy axiom (I3) in Definition 2. If the bijective functions φ_i, $i \in [t]$, are clear from context, we do not specify them explicitly.

▶ **Definition 7** (Union Representation)**.** Let $P = (E, \mathcal{I})$ be an independence system. Let $(E, \{A_i, \varphi_i\}|_{i\in[t]})$ be defined as in Definition 6. Then, $(E, \{A_i, \varphi_i\}|_{i\in[t]})$ is a *t-union representation* of P if the independence system represented by $(E, \{A_i, \varphi_i\}|_{i\in[t]})$ is isomorphic to P. Furthermore, we say that $(E, \{A_i, \varphi_i\}|_{i\in[t]})$ is defined over $\mathbb{F}$ if $\mathbb{F}$ is the field over which $A_1, A_2, \ldots, A_t$ are defined.

Finally, we present the definition of a *k-truncation* of a matroid, which comes in handy in various algorithmic applications. Our main result directly captures structures that we call *weak k-truncations of transversal matroids* rather than only transversal matroids, and hence we present it in this context (see Section 3). Let us first present the standard definition of truncation.

▶ **Definition 8** (Truncation)**.** Let $M = (E, \mathcal{I})$ be a matroid, and let $k \in \mathbb{N}$. The *k-truncation* of M is the matroid $M' = (E, \mathcal{I}')$ where $\mathcal{I}' = \{I \in \mathcal{I} : |I| \leq k\}$.

We now turn the present the definition with whom we will be working. This definition is sufficient for our algorithmic purposes, since the motivation underlying the use of a k-truncation is to obtain a matrix of small rank (i.e. $k^{\mathcal{O}(1)}$), which is also attainable by a weak k-truncation.

▶ **Definition 9** (Weak Truncation)**.** Let $M = (E, \mathcal{I})$ be a matroid, and let $k \in \mathbb{N}$. A *weak k-truncation* of M is an independence system $P' = (E, \mathcal{I}')$ where $\{I \in \mathcal{I} : |I| \leq k\} \subseteq \mathcal{I}' \subseteq \mathcal{I}$.

2.2 Isolation

For the sake of clarity, let us first introduce the following notation. Given $n \in \mathbb{N}$, let G be a bipartite graph with a fixed bipartition (A, B) such that $|A|, |B| \leq n$, and fixed injective functions $\gamma_A : A \to [n]$ and $\gamma_B : B \to [n]$. Given a weight function $w : [n] \times [n] \to \mathbb{N}$, we define the weight of an edge $\{a, b\} \in E(G)$, where $a \in A$ and $b \in B$, by $\widetilde{w}(\{a, b\}) = w(\gamma_A(a), \gamma_B(b))$. Thus, $\widetilde{w}$ can be thought of as a function from $E(G)$ to $\mathbb{N}$. Let us remind that for a subset $U \subseteq E(G)$, $\widetilde{w}(U) = \sum_{e \in U} \widetilde{w}(e)$.

We remark that we need to define a weight function via injective functions of the form γ_A and γ_B as above (rather than letting the domain directly be an edge set) in order to prove the correctness of our main result, particularly in its general form. Now, for perfect matchings, isolating weight functions are defined as follows.

▶ **Definition 10** (Isolating Weight Function)**.** Let G be a bipartite graph with a fixed bipartition (A, B) such that $|A|, |B| \leq n$, and fixed injective functions $\gamma_A : A \to [n]$ and $\gamma_B : B \to [n]$. A weight function $w : [n] \times [n] \to \mathbb{N}$ is *isolating* if it satisfies the following condition: If G has a perfect matching, then G also has a unique perfect matching μ of minimum weight (i.e. for every perfect matching $\mu' \neq \mu$, $\widetilde{w}(\mu) < \widetilde{w}(\mu')$).

Such isolating weight functions are particularly relevant to the detection of a perfect matching. To see this, we first need to define the matrix associated with an isolating weight function.

▶ **Definition 11.** Let G be a bipartite graph with a fixed bipartition (A, B) such that $|A|, |B| \leq n$, and fixed injective functions $\gamma_A : A \to [n]$ and $\gamma_B : B \to [n]$. In addition, let $w : [n] \times [n] \to \mathbb{N}$ be a weight function. Then, $W_{(G,w)}$ is the matrix on $|A|$ columns indexed by the vertices in A and $|B|$ rows indexed by the vertices in B, where

$$W_{(G,w)}[b, a] = \begin{cases} 2^{\widetilde{w}(\{b,a\})} & \text{if } \{b, a\} \in E(G) \\ 0 & \text{otherwise} \end{cases}$$

for all $a \in A$ and $b \in B$.

The following well-known result, due to Mulmuley et al. [15], reveals a connection between isolating weight functions, determinants and perfect matchings.

▶ **Proposition 1** ([15])**.** *Let G be a bipartite graph with a fixed bipartition (A, B) such that $|A| = |B| \leq n$, and fixed injective functions $\gamma_A : A \to [n]$ and $\gamma_B : B \to [n]$. In addition, let $w : [n] \times [n] \to \mathbb{N}$ be a weight function. If $\mathsf{det}(W_{(G,w)}) \neq 0$, then G has a perfect matching. Moreover, if w is isolating and G has a perfect matching, then $\mathsf{det}(W_{(G,w)}) \neq 0$.*

Fenner et al. [4] presented a (deterministic) computation of a collection of weight functions that, for any bipartite graph, has at least one isolating weight function. Formally,

▶ **Definition 12** (Isolating Collection)**.** Let $n \in \mathbb{N}$. An *n-isolating collection* is a set $\mathcal{W}_n$ of weight functions $w : [n] \times [n] \to \mathbb{N}$ with the following property: For any bipartite graph G with a fixed bipartition (A, B) such that $|A|, |B| \leq n$, and fixed bijective functions $\gamma_A : A \to [n]$ and $\gamma_B : B \to [n]$, there exists a weight function $w \in \mathcal{W}_n$ such that w is isolating.

▶ **Proposition 2** ([4])**.** *Let $n \in \mathbb{N}$. An n-isolating collection $\mathcal{W}_n$ of $2^{\mathcal{O}(\log^2 n)}$ weight functions with the following property can be obtained in time $2^{\mathcal{O}(\log^2 n)}$: For any weight function $w \in \mathcal{W}_n$, every weight assigned by w can be represented (in binary) using $\mathcal{O}(\log^2 n)$ bits.*

2.3 Splitters, Representative Families

Splitters are well-known tools in derandomization, formally defined as follows.

▶ **Definition 13** (Splitter). Let $n, k, \ell \in \mathbb{N}$ where $k \leq \ell$. An (n, k, ℓ)-*splitter* is a family $\mathcal{F}$ of functions from $[n]$ to $[\ell]$ such that for every $S \subseteq [n]$ of size k, there is a function $f \in \mathcal{F}$ that satisfies $f(i) \neq f(j)$ for all distinct $i, j \in S$.

We are specifically interested in an (n, k, k^2)-splitter. The following lemma asserts that such a small splitter can be computed efficiently.

▶ **Proposition 3** ([16]). *Given $n, k \in \mathbb{N}$, an (n, k, k^2)-splitter of size $k^{\mathcal{O}(1)} \log n$ can be constructed in time $k^{\mathcal{O}(1)} n \log n$.*

The notion of a representative family (implicitly linked to that of a splitter), introduced by Fomin et al. [5], plays a central role in the design of fast deterministic parameterized algorithms.

▶ **Definition 14** (Representative Family). Given a matroid $M = (E, \mathcal{I})$ and a family $\mathcal{S}$ of subsets of E, a subfamily $\widehat{\mathcal{S}} \subseteq \mathcal{S}$ is *q-representative* for $\mathcal{S}$, denoted by $\widehat{\mathcal{S}} \subseteq^q_{rep} \mathcal{S}$, if the following holds: for every set $Y \subseteq E$ of size at most q, if there is a set $X \in \mathcal{S}$ disjoint from Y with $X \cup Y \in \mathcal{I}$, then there is a set $\widehat{X} \in \widehat{\mathcal{S}}$ disjoint from Y with $\widehat{X} \cup Y \in \mathcal{I}$.

3 Representation

The purpose of this section is to compute a union representation of a transversal matroid consisting of a quasipolynomial (in the rank of the matroid) number of matrices. As our proof directly works for weak truncations of transversal matroids rather than only transversal matroids, we present the statement of our result in the following more general form, and the objective above as a corollary.

▶ **Theorem 15.** *Let G be an n-vertex bipartite graph with a fixed vertex bipartition (A, B), and let $r \in \mathbb{N}$. A t-union representation $(E, \{A_i, \varphi_i\}|_{i \in [t]})$ of some weak r-truncation of the transversal matroid of G over $\mathbb{Q}$, where $t = 2^{\mathcal{O}(\log^2 r)} \log n$ and every entry in A_i, $i \in [t]$, is an integer of bit-length $2^{\mathcal{O}(\log^2 r)}$, can be computed in time $2^{\mathcal{O}(\log^2 r)} n \log n$.*

Let us remind that the maximum size of a matching in a graph G is denoted by $\kappa(G)$, and that it upper bounds the rank of the transversal matroid of G. In the theorem above, if $r = \kappa(G)$, then any weak r-truncation of the transversal matroid of G is equal to the transversal matroid of G. Hence, we have the following corollary.

▶ **Corollary 16.** *Let G be an n-vertex bipartite graph with a fixed vertex bipartition (A, B), and denote $r = \kappa(G)$. A t-union representation $(E, \{A_i, \varphi_i\}|_{i \in [t]})$ of the transversal matroid of G over $\mathbb{Q}$, where $t = 2^{\mathcal{O}(\log^2 r)} \log n$ and every entry in A_i, $i \in [t]$, is an integer of bit-length $2^{\mathcal{O}(\log^2 r)}$, can be computed in time $2^{\mathcal{O}(\log^2 r)} n \log n$.*

For the sake of clarity, we first analyze the special case where $|A|, |B| \leq (2r)^2$. More precisely, we prove a weaker version of Corollary 16, but it is conceptually convenient to think of this proof as the above special case given that we later map integers in $[2n]$ to integers in $[(2r)^2]$. Then, we present a more involved construction that handles the general case.

3.1 Special Case

For our analysis of the special case, we introduce the following definition.

▶ **Definition 17.** Let G be a bipartite graph with a fixed vertex bipartition (A, B) such that $|A|, |B| \leq n$, and fixed injective functions $\gamma_A : A \to [n]$ and $\gamma_B : B \to [n]$. A weight function $w : [n] \times [n] \to \mathbb{N}$ is *good for a subset* $X \subseteq A$ if $\mathsf{det}(W_{(G,w)}[Y, X]) \neq 0$ for some $Y \subseteq B$.

The heart of the proof of the special case is based on the two following lemmas.

▶ **Lemma 18 (*).** *Let G be a bipartite graph with a fixed vertex bipartition (A, B) such that $|A|, |B| \leq n$, and fixed injective functions $\gamma_A : A \to [n]$ and $\gamma_B : B \to [n]$. In addition, let $\mathcal{W}_n$ be an n-isolating collection. For every subset $X \subseteq A$, if X is independent in the transversal matroid of G, then there exists $w \in \mathcal{W}_n$ that is good for X.*

▶ **Lemma 19 (*).** *Let G be a bipartite graph with a fixed vertex bipartition (A, B) such that $|A|, |B| \leq n$, and fixed injective functions $\gamma_A : A \to [n]$ and $\gamma_B : B \to [n]$. In addition, let $\mathcal{W}_n$ be an n-isolating collection. For every subset $X \subseteq A$, if there exists $w \in \mathcal{W}_n$ that is good for X, then X is independent in the transversal matroid of G.*

Lemmas 18 and 19 lead us to the following result.

▶ **Lemma 20 (*).** *Let G be a bipartite graph with a fixed vertex bipartition (A, B) such that $|A|, |B| \leq n$, and fixed injective functions $\gamma_A : A \to [n]$ and $\gamma_B : B \to [n]$. In addition, let $\mathcal{W}_n$ be an n-isolating collection. Then, $(A, \{W_{(G,w)}\}|_{w \in \mathcal{W}_n})$ is a t-union representation of the transversal matroid of G over $\mathbb{Q}$, where $t = |\mathcal{W}_n|$.*

Due to Proposition 2, we have the following consequence of Lemma 20.

▶ **Lemma 21 (*).** *Let G be an n-vertex bipartite graph with a fixed vertex bipartition (A, B). A t-union representation $(E, \{A_i, \varphi_i\}|_{i \in [t]})$ of the transversal matroid of G over $\mathbb{Q}$, where $t = 2^{\mathcal{O}(\log^2 n)}$ and every entry in A_i, $i \in [t]$, is an integer of bit-length $2^{\mathcal{O}(\log^2 n)}$, can be computed in time $2^{\mathcal{O}(\log^2 n)}$.*

3.2 General Case

We begin by adapting the definition of the matrix $W_{(G,w)}$ to the presence of a "splitter functions", which is a function from $[2n]$ to $[(2r)^2]$ where $n, r \in \mathbb{N}$ will be clear from context.

▶ **Definition 22.** Let G be a bipartite graph with a fixed bipartition (A, B) such that $|A|, |B| \leq n$, and fixed injective functions $\gamma_A : A \to [n]$ and $\gamma_B : B \to [n]$. In addition, let $w : [(2r)^2] \times [(2r)^2] \to \mathbb{N}$ be a weight function and $f : [2n] \to [(2r)^2]$ be a splitter function for some $r \in \mathbb{N}$. Then, $W_{(G,w,f)}$ is the matrix on $|A|$ columns indexed by the vertices in A and $|B|$ rows indexed by the vertices in B, where

$$W_{(G,w,f)}[b, a] = \begin{cases} 2^{w(f(\gamma_A(a)), f(n+\gamma_B(b)))} & \text{if } \{b, a\} \in E(G) \\ 0 & \text{otherwise} \end{cases}$$

for all $a \in A$ and $b \in B$.

In order to proceed, we need to generalize Definition 17 to pairs of a weight function and a splitter function.

▶ **Definition 23.** Let G be a bipartite graph with a fixed vertex bipartition (A, B) such that $|A|, |B| \leq n$, and fixed injective functions $\gamma_A : A \to [n]$ and $\gamma_B : B \to [n]$. In addition, let $r \in \mathbb{N}$. For a weight function $w : [(2r)^2] \times [(2r)^2] \to \mathbb{N}$ and a splitter function $f_A : [2n] \to [(2r)^2]$, the pair (w, f) is *good for a subset* $X \subseteq A$ if $\mathsf{det}(W_{(G,w,f)}[Y, X]) \neq 0$ for some $Y \subseteq B$.

We first need to establish the following lemma.

▶ **Lemma 24 (*).** *Let G be a bipartite graph with a fixed vertex bipartition (A, B) such that $|A|, |B| \leq n$, and fixed injective functions $\gamma_A : A \to [n]$ and $\gamma_B : B \to [n]$. In addition, let $\mathcal{W}$ be a $(2r)^2$-isolating collection, and $\mathcal{F}$ be a $(2n, 2r, (2r)^2)$-splitter for some $r \in \mathbb{N}$. For every subset $X \subseteq A$ of size at most r, if X is independent in the transversal matroid of G, then there exist $w \in \mathcal{W}$ and $f \in \mathcal{F}$ such that (w, f) is good for X.*

▶ **Lemma 25 (*).** *Let G be a bipartite graph with a fixed vertex bipartition (A, B) such that $|A|, |B| \leq n$, and fixed injective functions $\gamma_A : A \to [n]$ and $\gamma_B : B \to [n]$. In addition, let $\mathcal{W}$ be a $(2r)^2$-isolating collection, and $\mathcal{F}$ be a $(2n, 2r, (2r)^2)$-splitter for some $r \in \mathbb{N}$. For every subset $X \subseteq A$, if there exist $w \in \mathcal{W}$ and $f \in \mathcal{F}$ such that (w, f) is good for X, then X is independent in the transversal matroid of G.*

Lemmas 24 and 25 lead us to the following result.

▶ **Lemma 26 (*).** *Fix $r \in \mathbb{N}$. Let G be a bipartite graph with a fixed vertex bipartition (A, B) such that $|A|, |B| \leq n$, and fixed injective functions $\gamma_A : A \to [n]$ and $\gamma_B : B \to [n]$. In addition, let $\mathcal{W}$ be a $(2r)^2$-isolating collection, and $\mathcal{F}$ be a $(2n, 2r, (2r)^2)$-splitter. Then, $(A, \{W_{(G,w,f)}\}|_{w\in\mathcal{W}, f\in\mathcal{F}})$ is a t-union representation of some weak r-truncation of the transversal matroid of G over $\mathbb{Q}$, where $t = |\mathcal{W}| \cdot |\mathcal{F}|$.*

We are now ready to prove Theorem 15.

Proof. First, we apply Proposition 2 to obtain a $(2r)^2$-isolating collection $\mathcal{W}$ of size $2^{\mathcal{O}(\log^2 r)}$ in time $2^{\mathcal{O}(\log^2 r)}$. Second, we apply Proposition 13 to obtain a $(2n, 2r, (2r)^2)$-splitter $\mathcal{F}$ of size $r^{\mathcal{O}(1)} \log n$ in time $r^{\mathcal{O}(1)} n \log n$. We select arbitrary bijective functions $\gamma_A : A \to [|A|]$ and $\gamma_B : B \to [|B|]$. By Lemma 26, $(A, \{W_{(G,w,f)}\}|_{w\in\mathcal{W}, f\in\mathcal{F}})$ is a t-union representation of some weak r-truncation of the transversal matroid of G over $\mathbb{Q}$, where $t = |\mathcal{W}| \cdot |\mathcal{F}| = 2^{\mathcal{O}(\log^2 r)} \log n$. By Proposition 2 and Definition 11, every entry in $W_{(G,w,f)}$, $w \in \mathcal{W}$ and $f \in \mathcal{F}$, is an integer of bit-length $2^{\mathcal{O}(\log^2 r)}$. Thus, the time to construct $(A, \{W_{(G,w,f)}\}|_{w\in\mathcal{W}, f\in\mathcal{F}})$ is bounded by $2^{\mathcal{O}(\log^2 r)} n \log n$. This concludes the proof. ◀

4 Representative Families

In this section, we give applications of Theorem 15 in the design of parameterized algorithms. First, we give a fast deterministic algorithm to compute representative families over a transversal matroid. Prior to our work, only randomized algorithms were known from the works of Fomin et al. [5] and Lokshtanov et al. [12], since no fast deterministic algorithm was known for computing a linear representation of transversal matroids. Later in this section, we will use this deterministic algorithm to give a deterministic parameterized algorithm for the List k-Path problem. We remind that, as explained in Introduction, we selected List k-Path for illustrative purposes, and that the approach described to solve it is readily applicable to problems such as Graph Motif, d-Dimensional k-Matching and d-Set k-Packing in the presence of color lists.

We begin by stating the following known results about the computation of representative families over linear matroids.

▶ **Proposition 4** ([12]). *Let $M = (E, \mathcal{I})$ be a linear matroid of rank n, and let $\mathcal{S}$ be a family of ℓ independent sets, each of size p. Let A be an $n \times |E|$ matrix representing M over a field $\mathbb{F}$, and let $\omega < 2.373$ be the exponent of matrix multiplication [6]. Then, there are deterministic algorithms computing $\widehat{\mathcal{S}} \subseteq^q_{rep} \mathcal{S}$ as follows.*

- **(i)** *A family $\widehat{\mathcal{S}}$ of size $\binom{p+q}{p}$ in $\mathcal{O}\left(\binom{p+q}{p}^2 \ell p^3 n^2 + \ell \binom{p+q}{q}^{\omega} np\right) + (n+|E|)^{\mathcal{O}(1)}$ operations over $\mathbb{F}$.*
- **(ii)** *A family $\widehat{\mathcal{S}}$ of size $np\binom{p+q}{p}$ in $\mathcal{O}\left(\binom{p+q}{p} \ell p^3 n^2 + \ell \binom{p+q}{q}^{\omega-1} (pn)^{\omega-1}\right) + (n+|E|)^{\mathcal{O}(1)}$ operations over $\mathbb{F}$.*

▶ **Proposition 5** ([5]). *Let $M = (E, \mathcal{I})$ be a matroid and $\mathcal{S}$ be a subset of $\mathcal{I}$. If $\mathcal{S}' \subseteq^q_{rep} \mathcal{S}$ and $\widehat{\mathcal{S}} \subseteq^q_{rep} \mathcal{S}'$, then $\widehat{\mathcal{S}} \subseteq^q_{rep} \mathcal{S}$.*

Now we will apply the above results to prove the following theorem.

▶ **Theorem 27** (*). *Let $M = (E, \mathcal{I})$ be a transversal matroid of rank n and let $\mathcal{S}$ be a family of ℓ independent sets, each of size p. Let q be a positive integer, $r = p + q$ and ω be the exponent of matrix multiplication. Then, there are deterministic algorithms computing $\widehat{\mathcal{S}} \subseteq^q_{rep} \mathcal{S}$ as follows.*

- **(i)** *A family $\widehat{\mathcal{S}}$ of size $2^{\mathcal{O}(\log^2 r)}\binom{r}{p} \log n$ in time $\left(\binom{r}{p}^2 \ell p^3 n^2 + \ell \binom{r}{q}^{\omega} np\right) 2^{\mathcal{O}(\log^2 r)} \log n + (n + |E|)^{\mathcal{O}(1)} 2^{\mathcal{O}(\log^2 r)}$.*
- **(ii)** *A family $\widehat{\mathcal{S}}$ of size $2^{\mathcal{O}(\log^2 r)}\binom{r}{p} np \log n$ in time $\left(\binom{r}{p} \ell p^3 n^2 + \ell \binom{r}{q}^{\omega-1} (pn)^{\omega-1}\right)$ $2^{\mathcal{O}(\log^2 r)} \log n + (n + |E|)^{\mathcal{O}(1)} 2^{\mathcal{O}(\log^2 r)}$.*

Now we will use Theorem 27 to design a deterministic algorithm for List k-Path, which is defined as follows.

List k-Path	**Parameter:** k
Input: A graph G, a set of colors C, a function $L : V(G) \to 2^C$, and $k \in \mathbb{N}$,	
Question: Is there a path P on k vertices and an injective map $g : V(P) \to C$ such that $g(v) \in L(v)$ for all $v \in V(P)$?	

Note that, unlike k-Path where the objective is to check whether there is a path P on k vertices in a given input graph, in List k-Path we must also find an injective map g that assigns distinct colors to the vertices on P from their respective lists of colors. Towards that, we create an auxiliary bipartite graph H with bipartition $V(G) \uplus C$, such that for each $v \in V(G)$, the neighborhood $N_H(v)$ of v in H is $L(v)$. The following lemma states that any solution to the instance (G, C, L, k) is also an independent set in the transversal matroid of H.

▶ **Lemma 28** (*). *Let (G, C, L, k) be an instance of* List k-Path. *Let $H = (V(G) \uplus C, F)$ be a bipartite graph such that $N_H(v) = L(v)$ for all $v \in V(G)$. Let P be a path on k vertices in G. Then, P is a solution to* List k-Path *if and only if $V(P)$ is an independent set in the transversal matroid M of H over the ground set $V(G)$.*

Using Lemma 28, a dynamic programming (DP) algorithm for List k-Path can be designed using representative families. This algorithm will follow the outline of the algorithm of Fomin et al. [5] for k-Path. In the rest of this section, we will present this DP algorithm

for List k-Path. Recall that (G, C, L, k) is the input and $M = (V(G), \mathcal{I})$ is the transversal matroid of H over the ground set $V(G)$, where $H = (V(G) \uplus C, F)$ is the bipartite graphs such that $N_H(v) = L(v)$ for all $v \in V(G)$. For any $i \in [k]$ and $v \in V(G)$, define the following.

$$\mathcal{P}_v^i = \Big\{ X \;\Big|\; X \subseteq V(G),\ v \in X,\ |X| = i, X \in \mathcal{I},\ \text{and } G \text{ has a path of length } i-1 \text{ whose vertex set is precisely } X \text{ and whose end vertex is } v \Big\}$$

The following lemma gives an efficient computation of $\widehat{\mathcal{P}_v^i} \subseteq_{rep}^{k-i} \mathcal{P}_v^i$ for all $i \in [k]$ where the underlying matroid is M, i.e. the transversal matroid of H over the ground set $V(G)$.

▶ **Lemma 29 (*).** *For every $i \in [k]$ and $v \in V(G)$, $\widehat{\mathcal{P}_v^i} \subseteq_{rep}^{k-i} \mathcal{P}_v^i$ of size $2^{\mathcal{O}(\log^2 k)}\binom{k}{i} n \cdot i \log n$ can be computed in time $2^{\omega k} 2^{\mathcal{O}(\log^2 k)} n^{\mathcal{O}(1)}$.*

▶ **Theorem 30 (*).** List k-Path *can be solved in time $2^{\omega k} 2^{\mathcal{O}(\log^2 k)} n^{\mathcal{O}(1)}$, where $\omega < 2.373$ is the exponent of matrix multiplication [6].*

We remark that in the theorem above, $2^{\omega k} 2^{\mathcal{O}(\log^2 k)} n^{\mathcal{O}(1)} = 5.18^k n^{\mathcal{O}(1)}$. If the computation in Proposition 4 is sped-up or the bound on ω is improved, then our algorithm is automatically sped-up as well.

References

1. Andreas Björklund, Petteri Kaski, and Lukasz Kowalik. Constrained multilinear detection and generalized graph motifs. *Algorithmica*, 74(2):947–967, 2016. doi:10.1007/s00453-015-9981-1.
2. R A DeMillo and R J Lipton. A probabilistic remark on algebraic program testing. *Inform. Process Lett.*, 7(4):193–195, 1978.
3. Banu Dost, Tomer Shlomi, Nitin Gupta, Eytan Ruppin, Vineet Bafna, and Roded Sharan. Qnet: A tool for querying protein interaction networks. *Journal of Computational Biology*, 15(7):913–925, 2008. doi:10.1089/cmb.2007.0172.
4. Stephen A. Fenner, Rohit Gurjar, and Thomas Thierauf. Bipartite perfect matching is in quasi-nc. 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 754–763. ACM, 2016. doi:10.1145/2897518.2897564.
5. Fedor V. Fomin, Daniel Lokshtanov, Fahad Panolan, and Saket Saurabh. Efficient computation of representative families with applications in parameterized and exact algorithms. *J. ACM*, 63(4):29:1–29:60, 2016. doi:10.1145/2886094.
6. François Le Gall. Powers of tensors and fast matrix multiplication. In Katsusuke Nabeshima, Kosaku Nagasaka, Franz Winkler, and Ágnes Szántó, editors, *International Symposium on Symbolic and Algebraic Computation, ISSAC '14, Kobe, Japan, July 23-25, 2014*, pages 296–303. ACM, 2014. doi:10.1145/2608628.2608664.
7. Shafi Goldwasser and Ofer Grossman. Bipartite perfect matching in pseudo-deterministic NC. 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 87:1–87:13. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2017. doi:10.4230/LIPIcs.ICALP.2017.87.
8. Prachi Goyal, Neeldhara Misra, Fahad Panolan, and Meirav Zehavi. Deterministic algorithms for matching and packing problems based on representative sets. *SIAM J. Discrete Math.*, 29(4):1815–1836, 2015. doi:10.1137/140981290.
9. Sylvain Guillemot and Florian Sikora. Finding and counting vertex-colored subtrees. *Algorithmica*, 65(4):828–844, 2013. doi:10.1007/s00453-011-9600-8.

10 Rohit Gurjar and Thomas Thierauf. Linear matroid intersection is in quasi-nc. 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 821–830. ACM, 2017. doi:10.1145/3055399.3055440.

11 Ioannis Koutis. Constrained multilinear detection for faster functional motif discovery. *Inf. Process. Lett.*, 112(22):889–892, 2012. doi:10.1016/j.ipl.2012.08.008.

12 Daniel Lokshtanov, Pranabendu Misra, Fahad Panolan, and Saket Saurabh. Deterministic truncation of linear matroids. 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 922–934. Springer, 2015. doi:10.1007/978-3-662-47672-7_75.

13 Dániel Marx. A parameterized view on matroid optimization problems. *Theor. Comput. Sci.*, 410(44):4471–4479, 2009. doi:10.1016/j.tcs.2009.07.027.

14 Pranabendu Misra, Fahad Panolan, M. S. Ramanujan, and Saket Saurabh. Linear representation of transversal matroids and gammoids parameterized by rank. In Yixin Cao and Jianer Chen, editors, *Computing and Combinatorics - 23rd International Conference, COCOON 2017, Hong Kong, China, August 3-5, 2017, Proceedings*, volume 10392 of *Lecture Notes in Computer Science*, pages 420–432. Springer, 2017. doi:10.1007/978-3-319-62389-4_35.

15 Ketan Mulmuley, Umesh V. Vazirani, and Vijay V. Vazirani. Matching is as easy as matrix inversion. *Combinatorica*, 7(1):105–113, 1987. doi:10.1007/BF02579206.

16 Moni Naor, Leonard J. Schulman, and Aravind Srinivasan. Splitters and near-optimal derandomization. In *36th Annual Symposium on Foundations of Computer Science, Milwaukee, Wisconsin, 23-25 October 1995*, pages 182–191. IEEE Computer Society, 1995. doi:10.1109/SFCS.1995.492475.

17 J.G. Oxley. *Matroid Theory*. Oxford graduate texts in mathematics. Oxford University Press, 2006.

18 Fahad Panolan and Meirav Zehavi. Parameterized algorithms for list k-cycle. 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 22:1–22:15. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2016. doi:10.4230/LIPIcs.FSTTCS.2016.22.

19 Ron Y. Pinter, Oleg Rokhlenko, Esti Yeger Lotem, and Michal Ziv-Ukelson. Alignment of metabolic pathways. *Bioinformatics*, 21(16):3401–3408, 2005. doi:10.1093/bioinformatics/bti554.

20 Ron Y. Pinter, Hadas Shachnai, and Meirav Zehavi. Deterministic parameterized algorithms for the graph motif problem. *Discrete Applied Mathematics*, 213:162–178, 2016. doi:10.1016/j.dam.2016.04.026.

21 Ron Y. Pinter and Meirav Zehavi. Algorithms for topology-free and alignment network queries. *J. Discrete Algorithms*, 27:29–53, 2014. doi:10.1016/j.jda.2014.03.002.

22 J T Schwartz. Fast probabilistic algorithms for verification of polynomial identities. *J. Assoc. Comput. Mach.*, 27(4):701–717, 1980.

23 Tomer Shlomi, Daniel Segal, Eytan Ruppin, and Roded Sharan. Qpath: a method for querying pathways in a protein-protein interaction network. *BMC Bioinformatics*, 7:199, 2006. doi:10.1186/1471-2105-7-199.

24 Ola Svensson and Jakub Tarnawski. The matching problem in general graphs is in quasi-nc. *CoRR*, abs/1704.01929, 2017. arXiv:1704.01929.

25 R Zippel. Probabilistic algorithms for sparse polynomials. In *EUROSAM*, pages 216–226, 1979.

Selection Problems in the Presence of Implicit Bias*

Jon Kleinberg[†1] **and Manish Raghavan**[‡2]

1 **Cornell University, Ithaca, USA**
kleinber@cs.cornell.edu
2 **Cornell University, Ithaca, USA**
manish@cs.cornell.edu

Abstract

Over the past two decades, the notion of implicit bias has come to serve as an important component in our understanding of bias and discrimination in activities such as hiring, promotion, and school admissions. Research on implicit bias posits that when people evaluate others – for example, in a hiring context – their unconscious biases about membership in particular demographic groups can have an effect on their decision-making, even when they have no deliberate intention to discriminate against members of these groups. A growing body of experimental work has demonstrated the effect that implicit bias can have in producing adverse outcomes.

Here we propose a theoretical model for studying the effects of implicit bias on selection decisions, and a way of analyzing possible procedural remedies for implicit bias within this model. A canonical situation represented by our model is a hiring setting, in which recruiters are trying to evaluate the future potential of job applicants, but their estimates of potential are skewed by an unconscious bias against members of one group. In this model, we show that measures such as the *Rooney Rule*, a requirement that at least one member of an underrepresented group be selected, can not only improve the representation of the affected group, but also lead to higher payoffs in absolute terms for the organization performing the recruiting. However, identifying the conditions under which such measures can lead to improved payoffs involves subtle tradeoffs between the extent of the bias and the underlying distribution of applicant characteristics, leading to novel theoretical questions about order statistics in the presence of probabilistic side information.

1998 ACM Subject Classification F.2 Analysis of Algorithms and Problem Complexity

Keywords and phrases algorithmic fairness, power laws, order statistics, implicit bias, Rooney Rule

Digital Object Identifier 10.4230/LIPIcs.ITCS.2018.33

1 Introduction

Over the past two decades, the notion of *implicit bias* [12] has come to provide on important perspective on the nature of discrimination. Research on implicit bias argues that unconscious attitudes toward members of different demographic groups – for example, defined by gender, race, ethnicity, national origin, sexual orientation, and other characteristics – can have a

* A full version of the paper is available at http://arxiv.org/abs/1801.03533.
† JK is supported in part by a Simons Investigator Award, an ARO MURI grant, a Google Research Grant, and a Facebook Faculty Research Grant.
‡ MR is supported by an NSF Graduate Research Fellowship (DGE-1650441).

non-trivial impact on the way in which we evaluate members of these groups; and this in turn may affect outcomes in employment [1, 2, 20], education [21], law [13, 14], medicine [11], and other societal institutions.

In the context of a process like hiring, implicit bias thus shifts the question of bias and discrimination to be not just about identifying bad actors who are intentionally discriminating, but also about the tendency of all of us to reach discriminatory conclusions based on unconscious application of stereotypes. An understanding of these issues also helps inform the design of interventions to mitigate implicit bias – when essentially all of us have a latent tendency toward low-level discrimination, a set of broader practices may be needed to guide the process toward the desired outcome.

A basic mechanism: The Rooney Rule

One of the most basic and widely adopted mechanisms in practice for addressing implicit bias in hiring and selection is the *Rooney Rule* [6], which, roughly speaking, requires that in recruiting for a job opening, one of the candidates interviewed must come from an underrepresented group. The Rooney Rule is named for a protocol adopted by the National Football League (NFL) in 2002 in response to widespread concern over the low representation of African-Americans in head coaching positions; it required that when a team is searching for a new head coach, at least one minority candidate must be interviewed for the position. Subsequently the Rooney Rule has become a guideline adopted in many areas of business [4]; for example, in 2015 then-President Obama exhorted leading tech firms to use the Rooney Rule for hiring executives, and in recent years companies including Amazon, Facebook, Microsoft, and Pinterest have adopted a version of the Rooney Rule requiring that at least one candidate interviewed must be a woman or a member of an underrepresented minority group [17]. Earlier this year, the much-awaited set of recommendations made by Eric Holder and colleagues to address workplace bias and discrimination at Uber advocated for the use of the Rooney Rule as one of its key points [7, 18].

The Rooney Rule is the subject of ongoing debate, and one crucial aspect of this debate is the following tension. On one side is the argument that implicit (or explicit) bias is preventing deserving candidates from underrepresented groups from being fairly considered, and the Rooney Rule is providing a force that counter-balances and partially offsets the consequences of this underlying bias. On the other side is the concern that if a job search process produces a short-list of top candidates all from the majority group, it may be because these are genuinely the strongest candidates despite the underlying bias – particularly if there is a shortage of available candidates from other groups. In this case, wholesale use of the Rooney Rule may lead firms to consider weaker candidates from underrepresented groups, which works against the elimination of unconscious stereotypes. Of course, there are other reasons to seek diversity in recruiting that may involve broader considerations or longer time horizons than just the specific applicants being evaluated; but even these lines of argument generally incorporate the more local question of the effect on the set of applicants.

Given the widespread consideration of the Rooney Rule from both legal and empirical perspectives [6], it is striking that prior work has not attempted to formalize the inherently mathematical question that forms a crucial ingredient in these debates: given some estimates of the extent of bias and the prevalence of available minority candidates, does the expected quality of the candidates being interviewed by a hiring committee go up or down when the Rooney Rule is implemented? When the bias is large and there are many minority candidates, it is quite possible that the hiring committee's bias caused it to choose a weaker candidate over a stronger minority one, and the Rooney Rule may be strengthening the pool

of interviewees by reversing this decision and forcing them to swap the stronger minority candidate in. But when the bias is small or there are few minority candidates, the Rule might be reversing a decision that in fact chose the stronger applicant.

In this paper, we propose a formalization of this family of questions, via a simplified model of selection with implicit bias, and we give a tight analysis of the consequences of using the Rooney Rule in this setting. In particular, when selecting for a fixed number of slots, we find that a sharp threshold on the effectiveness of the Rooney Rule in our model depends on three parameters: not just the extent of bias and the the prevalence of available minority candidates, but a third quantity as well – essentially, a parameter governing the heavy-tailed behavior of candidates' expected future job performance. We emphasize that our model is deliberately stylized, to abstract the trade-offs as cleanly as possible. Moreover, in interpreting these results, we emphasize a point noted above, that there are other reasons to consider using the Rooney Rule beyond the issues that motivate this particular formulation; but an understanding of the trade-offs in our model seems informative in any broader debate about such hiring and selection measures.

We now describe the basic ingredients of our model, followed by a summary of the main results.

1.1 A Model of Selection with Implicit Bias

Our model is based on the following scenario. Suppose that a hiring committee is trying to fill an open job position, and it would like to choose the $k \geq 2$ best candidates as *finalists* to interview from among a large set of applicants. We will think of k as a small constant, and indeed most of the subtlety of the question already arises for the case $k = 2$.

X-candidates and Y-candidates

The set of all applicants is partitioned into two groups X and Y, where we think of Y as the majority group, and X as a minority group within the domain that may be subject to bias. For some positive real number $\alpha \leq 1$ and an integer n, there are n applicants from group Y and αn applicants from group X. If a candidate i belongs to X, we will refer to them as an *X-candidate*, and if i belongs to Y, we will refer to them as a *Y-candidate*. (The reader is welcome, for example, to think of the setting of academic hiring, with X as the female job applicants and Y as the male job applicants, but the formulation is general.)

Each candidate i has a (hidden) numerical value that we call their *potential*, representing their future performance over the course of their career. For example, in faculty hiring, we might think of the potential of each applicant in terms of some numerical proxy like their future lifetime citation count (with the caveat that any numerical measure will of course be an imperfect representation). Or in hiring executives, the potential of each applicant could be some measure of the revenue they will bring to the firm.

We assume that there is a common distribution Z that these numerical potentials come from: each potential is an independent draw from Z. (Thus, the applicants can have widely differing values for their numerical potentials; they just arise as draws from a common distribution.) For notational purposes, when i is an X-candidate, we write their potential as X_i, and when j is a Y-candidate, we write their potential as Y_j. We note an important modeling decision in this formulation: we are assuming that all X_i and all Y_j values come from this same distribution Z. While it is also of interest to consider the case in which the numerical potentials of the two groups X and Y are drawn from different group-specific distributions, we focus on the case of identical distributions for two reasons. First, there are

many settings where differences between the underlying distributions for different groups appear to be small compared to the bias-related effects we are seeking to measure; and second, in any formal analysis of bias between groups, the setting in which the groups begin with identical distributions is arguably the first fundamental special case that needs to be understood.

In the domains that we are considering – hiring executives, faculty members, athletes, performers – there is a natural functional form for the distribution Z of potentials, and this is the family of *power laws* (also known as *Pareto distributions*), with $\Pr[Z \geq t] = t^{-(1+\delta)}$ and support $[1, \infty)$ for a fixed $\delta > 0$. Extensive empirical work has argued that the distribution of individual output in a wide range of creative professions can be approximated by power law distributions with small positive values of δ [5]. For example, the distribution of lifetime citation counts is well-approximated by a power law, as are the lifetime downloads, views, or sales by performers, authors, and other artists. In the last part of the paper, we also consider the case in which the potentials are drawn from a distribution with bounded support, but for most of the paper we will focus on power laws.

Selection with Bias

Given the set of applicants, the hiring committee would like to choose k *finalists* to interview. The *utility* achieved by the committee is the sum of the potentials of the k finalists it chooses; the committee's goal is to maximize its utility.[1]

If the committee could exactly evaluate the potential of each applicant, then it would have a straightforward way to maximize the utility of the set of finalists: simply sort all applicants by potential, and choose the top k as finalists. The key feature of the situation we would like to capture, however, is that the committee is biased in its evaluations; we look for a model that incorporates this bias as cleanly as possible.

Empirical work in some of our core motivating settings – such as the evaluation of scientists and faculty candidates – indicates that evaluation committees often systematically downweight female and minority candidates of a given level of achievement, both in head-to-head comparisons and in ranking using numerical scores [22]. It is thus natural to model the hiring committee's evaluations as follows: they correctly estimate the potential of a Y-applicant j at the true value Y_j, but they estimate the potential of an X-applicant i at a reduced value $\tilde{X}_i < X_i$. They then rank candidates by these values $\{Y_j\}$ and $\{\tilde{X}_i\}$, and they choose the top k according to this biased ranking.

For most of the paper, we focus on the case of *multiplicative bias*, in which $\tilde{X}_i = X_i/\beta$ for a bias parameter[2] $\beta > 1$. This is a reasonable approximation to empirical data from human-subject studies [22]; and moreover, for power law distributions this multiplicative form is in a strong sense the "right" parametrization of the bias, since biases that grow either faster or slower than multiplicatively have a very simple asymptotic behavior in the power law case.

In this aspect of the model, as in others, we seek the cleanest formulation that exposes the key underlying issues; for example, it would be an interesting extension to consider versions in which the estimates for each individual are perturbed by random noise. A line of previous work [3, 9, 10] has analyzed models of ranking under noisy perturbations; while our

[1] Since our goal is to model processes like the Rooney Rule, which apply to the selection of finalists for interviewing, rather than to the hiring decision itself, we treat the choice of k finalists as the endpoint rather than modeling the interviews that subsequently ensue.

[2] When $\beta = 1$, the ranking has no bias.

scenario is quite different in that the entities being ranked are partitioned into a fixed set of groups with potentially different levels of bias and noise, it would be natural to see if these techniques could potentially be extended to handle noise in the context of implicit bias.

1.2 Main Questions and Results

This then is the basic model in which we analyze interventions with the structure of the Rooney Rule: (i) a set of n Y-applicants and αn X-applicants each have an independent future potential drawn from a power law distribution; (ii) a hiring committee ranks the applicants according to a sorted order in which each X-applicant's potential is divided down by β, and chooses the top k in this ordering as *finalists*; and (iii) the hiring committee's *utility* is the sum of the potentials of the k finalists.

Qualitatively, the motivation for the Rooney Rule in such settings is that hiring committees are either unwilling or unable to reasonably correct for their bias in performing such rankings, and therefore cannot be relied on to interview X-candidates on their own. The difficulty in removing this skew from such evaluations is a signature aspect of phenomena around implicit bias.

The decision to impose the Rooney Rule is made at the outset, before the actual values of the potentials $\{Y_j\}$ and $\{\tilde{X}_i\}$ are materialized. All that is known at the point of this initial decision to use the Rule or not are the parameters of the domain: the bias β, the relative abundance of X-candidates α, the power law exponent $1+\delta$, and the number of finalists to be chosen k. The question is: as a function of these four parameters, will the use of the Rooney Rule produce a positive or negative expected change in utility, where the expectation is taken over the random draws of applicant values? We note that one could instead ask about the probability that the Rooney Rule produces a positive change in utility as opposed to the expected change; in fact, our techniques naturally generalize to characterize not only the expected change, but the probability that this change is positive, as we will show in Section 2.

Our model lets us make precise the trade-off in utility that underpins the use of the Rooney Rule. If the committee selects an X-candidate on its own – even using its biased ranking – then their choice already satisfies the conditions of the Rule. But if all k finalists are Y-candidates, then the Rooney Rule requires that the committee replace the lowest-ranked of these finalists j with the highest-ranked X-candidate i. Because i was not already a finalist, we know that $\tilde{X}_i = X_i/\beta < Y_j$. But to see whether this yields a positive change in utility, we need to understand which of X_i or Y_j has a larger expected value, conditional on the information contained in the committee's decision, that $X_i/\beta < Y_j$.

Our main result is an exact characterization of when the Rooney Rule produces a positive expected change in terms of the four underlying parameters, showing that it non-trivially depends on all four. For the following theorem, and for the remainder of the paper, we assume $\alpha \in (0,1]$, $\beta > 1$, and $\delta > 0$. We begin with the case where $k = 2$.

▶ **Theorem 1.** *For $k = 2$ and sufficiently large n, the Rooney Rule produces a positive expected change if and only if $\phi_2(\alpha,\beta,\delta) > 1$ where*

$$\phi_2(\alpha,\beta,\delta) = \frac{\alpha^{1/(1+\delta)}\left[1-(1+c^{-1})^{-\delta/(1+\delta)}\left[1+\frac{\delta}{1+\delta}(1+c)^{-1}\right]\right]}{\frac{\delta}{1+\delta}(1+c)^{-1-\delta/(1+\delta)}} \tag{1}$$

and $c = \alpha\beta^{-(1+\delta)}$. Moreover, $\phi_2(\alpha,\beta,\delta)$ is increasing in β, so for fixed α and δ there exists β^* such that $\phi_2(\alpha,\beta,\delta) > 1$ if and only if $\beta > \beta^*$.

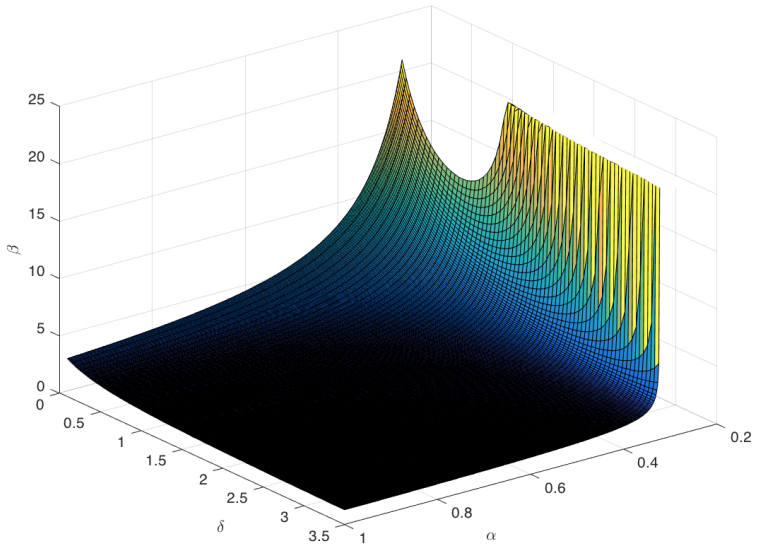


Figure 1 Fixing $k = 2$, the (α, β, δ) values for which the Rooney Rule produces a positive expected change for sufficiently large n lie above a surface defined by the function $\phi_2(\alpha, \beta, \delta) = 1$.

Thus, we have an explicit characterization for when the Rooney Rule produces positive expected change. The following theorem extends this to larger values of k.

▶ **Theorem 2.** *There is an explicit function $\phi_k(\alpha, \beta, \delta)$ such that the Rooney Rule produces a positive expected change, for n sufficiently large and $k = O(\ln n)$, if and only if $\phi_k(\alpha, \beta, \delta) > 1$.*

Surprisingly, even for larger values of k, there are parts of the parameter space for which the Rooney Rule increases and decreases expected quality, independent of the number of applicants n.

Figure 1 depicts a view of the function ϕ_2, by showing the points in three-dimensional (α, β, δ) space for which ϕ takes the value 1. The values for which the Rooney Rule produces a positive expected change for sufficiently large n lie above this surface. The surface is fairly complex, and it displays unexpected non-monotonic behavior; for example, on certain regions of fixed (α, β), it is non-monotonic in δ, a fact which is not a priori obvious. Moreover, there exist (α, δ) pairs above which the surface does not exist. (One example in Figure 1 occurs at $\alpha \approx 0.3$ and $\delta \approx 3$). Characterizing the function ϕ and its level set $\phi = 1$ is challenging, and it is noteworthy that the complexity of this function is arising from a relatively bare-bones formulation of the trade-off in the Rooney Rule; this suggests the function and its properties are capturing something inherent in the process of biased selection.

One monotonocity result we are able to establish for the function ϕ is the following, showing that for fixed (α, β, δ) increasing the number of positions can't make the Rooney Rule go from beneficial to harmful.

▶ **Theorem 3.** *For sufficiently large n and $k = O(\ln n)$, if the Rooney Rule produces a positive expected change at a given number of finalists k, it also produces a positive expected change when there are $k + 1$ finalists (at the same (α, β, δ)).*

We prove these theorems through an analysis of the *order statistics* of the underlying power law distribution. Specifically, if we draw m samples from the power law Z and sort them in ascending order from lowest to highest, then the ℓ^{th} item in the sorted list is a random variable denoted $Z_{(\ell:m)}$. To analyze the effect of the Rooney Rule, we are comparing $Y_{(n-k+1:n)}$ with $X_{(\alpha n:\alpha n)}$. Crucially, we are concerned with their expected values conditional on the fact that the committee chose the k^{th}-ranked Y-candidate over the top-ranked X-candidate, implying as noted above that $X_{(\alpha n:\alpha n)}/\beta < Y_{(n-k+1:n)}$. The crucial comparison is therefore between $\mathbb{E}\left[Y_{(n-k+1:n)} | X_{(\alpha n:\alpha n)} < \beta Y_{(n-k+1:n)}\right]$ and $\mathbb{E}\left[X_{(\alpha n:\alpha n)} | X_{(\alpha n:\alpha n)} < \beta Y_{(n-k+1:n)}\right]$. Order statistics conditional on side information turn out to behave in complex ways, and hence the core of the analysis is in dealing with these types of conditional order statistics for power law distributions.

More generally, given the ubiquity of power law distributions [5], we find it surprising how little is known about how their order statistics behave qualitatively. In this respect, the techniques we provide may prove to be independently useful in other applications. For example, we develop a tight asymptotic characterization of the expectations of order statistics from a power law distribution that to our knowledge is novel.

We also note that although our results are expressed for sufficiently large n, the convergence to the asymptotic behavior happens very quickly as n grows; to handle fixed values of n, we need only modify the bounds by correction terms that grow like $\left(1 \pm O\left(\frac{(\ln n)^2}{n}\right)\right)$. In particular, the errors in the asymptotic analysis are small once n reaches 50, which is reasonable for settings in which a job opening receives many applications.

Estimating the level of bias β

The analysis techniques we develop for proving Theorem 2 can also be used for related problems in this model. A specific question we are able to address is the problem of estimating the amount of bias from a history of hiring decisions.

In particular, suppose that over m years the hiring committee makes one offer per year; in N of the m years this offer goes to an X-candidate, and in $m - N$ of the m years this offer goes to a Y-candidate. Which value of the bias parameter β maximizes the probability of this sequence of observations?

We provide a tight characterization of the solution to this question, finding again that it depends not only on α (in this case, the sequence of α values for each year), but also on the power law exponent $1 + \delta$. The solution has a qualitatively natural structure, and produces $\beta = 1$ (corresponding to no bias) as the estimate when the fraction of X-candidates hired over the m years is equal to the expected number that would be hired under random selection.

Generalizations to other distributions

Finally, at the end of the paper we consider how to adapt our approach for classes of distributions other than power laws. A different category of distributions that can be motivated by the considerations discussed here is the set of bounded distributions, which take values only over a finite interval. Just as power laws are characteristic of the performance of employees in certain professions, bounded distributions are appropriate when there are absolute constraints on the maximum effect a single employee can have.

Moreover, bounded distributions are also of interest because they contain the uniform distribution on $[0, 1]$ as a special case. We can think of this special case as describing an instance in which each candidate is associated with their *quantile* (between 0 and 1) in a

ranking of the entire population, and the bias then operates on this quantile value, reducing it in the case of X-candidates.

For bounded distributions, we can handle much more general forms for the bias – essentially, any function that reduces the values X_i strictly below the maximum of the distribution (for instance, a bias that always prefers a Y-candidate to an X-candidate when they are within some ε of each other). When $k = 2$ and there are equal numbers of X-candidates and Y-candidates, we show that for any bounded distribution and any such bias, the Rooney Rule produces a positive expected change in utility for all sufficiently large n.

1.3 An Illustrative Special Case: Infinite Bias

To illustrate some of the basic considerations that go into our analysis and its interpretation, we begin with a simple special case that we can think of as "infinite bias" – the committee deterministically ranks every Y-candidate above every X-candidate. This case already exhibits structurally rich behavior, although the complexity is enormously less than the case of general β. We also focus here on $k = 2$. In terms of Figure 1, we can visualize the infinite bias case as if we are looking at the plot from infinitely high up; thus, reasoning about infinite bias amounts to finding what parts of the (α, δ) plane are covered by the surface $\phi_2(\alpha, \beta, \delta) = 1$.

With infinite bias, the committee is guaranteed to choose the two highest-ranked Y-candidates in the absence of an intervention; with the Rooney Rule, the committee will choose the highest-ranked Y-candidate and the highest-ranked X-candidate. As we discuss in the next section, for power law distributions with exponent $1 + \delta$, if z^* is the expected maximum of n draws from the distribution, then (i) the expected value of the second-largest of the n draws is $\frac{\delta}{(1+\delta)}z^*$; and (ii) the expected maximum of αn draws from the distribution is asymptotically $\alpha^{1/(1+\delta)}z^*$.

This lets us directly evaluate the utility consequences of the intervention. If there is no intervention, the utility of the committee's decision will be $\left(1 + \frac{\delta}{1+\delta}\right) z^*$, and if the Rooney Rule is used, the utility of the committee's decision will be $(1 + \alpha^{1/(1+\delta)})z^*$. Thus, the Rooney Rule produces positive expected change in utility if and only if $\alpha^{1/(1+\delta)} > \frac{\delta}{(1+\delta)}$; that is, if and only if $\alpha > \left(\frac{\delta}{1+\delta}\right)^{1+\delta}$.

In addition to providing a simple closed-form expression for when to use the Rooney Rule in this setting, the condition itself leads to some counter-intuitive consequences. In particular, the closed-form expression for the condition makes it clear that *for every* $\alpha > 0$, there exists a sufficiently small $\delta > 0$ so that when the distribution of applicant potentials is a power law with exponent $1 + \delta$, using the Rooney Rule produces the higher expected utility. In other words, with a power law exponent close to 1, it's a better strategy to commit one of the two offers to the X-candidates, even though they form an extremely small fraction of the population.

This appears to come perilously close to contradicting the following argument. We can arbitrarily divide the Y-candidates into two sets A and B of $n/2$ each; and if $\alpha < 1/2$, each of A and B is larger than the set of all X-candidates. Let a^* be the top candidate in A and b^* be the top candidate in B. Each of a^* and b^* has at least the expected value of the top X-candidate, and moreover, one of them is the top Y-candidate overall. So how can it be that choosing a^* and b^* fails to improve on the result of using the Rooney Rule?

The resolution is to notice that using the Rooney Rule still involves hiring the *top* Y-candidate. So it's not that the Rooney Rule chooses one of a^* or b^* at random, together with the top X-candidate. Rather, it chooses the *better* of a^* and b^*, along with the top

X-candidate. The real point is that power law distributions have so much probability in the tail that the best person among a set of αn can easily have a higher expected value than the second-best person among a set of n, even when α is quite small. This is a key property of power law distributions that helps explain what's happening both in this example and in our analysis.

1.4 A Non-Monotonicity Effect

As noted above, much of the complexity in the analysis arises from working with their expected values conditioned on the outcomes of certain biased comparisons. It turns out that expected values conditional on these types of comparisons can exhibit some fairly counter-intuitive behavior; to familiarize the reader with some of these phenomena – both as preparation for the subsequent sections, but also as an interesting end in itself – we offer the following example.

Much of our analysis involves quantities like $\mathbb{E}[X|X > \beta Y]$ – the conditional expectation of X, given that it exceeds some other random variable Y multiplied by a bias parameter. (We will also be analyzing the version in which the inequality goes in the other direction, but we'll focus on the current expression for now.) If we choose X and Y as independent random variables both drawn from a distribution Z, and then view the conditional expectation as a function just of the bias parameter β, what can we say about the properties of this function $f(\beta) = \mathbb{E}[X|X > \beta Y]$?

Intuitively we'd expect $f(\beta)$ to be monotonically increasing in β – indeed, as β increases, we're putting a stricter lower bound on X, and so this ought to raise the conditional expectation of X.

The surprise is that this is not true in general; we can construct independent random variables X and Y for which $f(\beta)$ is not monotonically increasing. In fact, the random variables are very simple: we can have each of X and Y take values independently and uniformly from the finite set $\{1, 5, 9, 13\}$. Now, the event $X > 2Y$ consists of four possible pairs of (X, Y) values: (5,1), (9,1), (13,1), and (13,5). Thus, $f(2) = \mathbb{E}[X|X > 2Y] = 10$. In contrast, the event $X > 3Y$ consists of three possible pairs of (X, Y) values: (5,1), (9,1), and (13,1). Thus, $f(3) = 9$, which is a smaller value, despite the fact that X is required to be a larger multiple of Y.

The surprising content of this example has a fairly sharp formulation in terms of a story about recruiting. Suppose that two academic departments, Department A and Department B, both engage in hiring each year. In our stylized setting, each interviews one X-candidate and one Y-candidate each year, and hires one of them. Each candidate comes from the uniform distribution on $\{1, 5, 9, 13\}$. Departments A and B are both biased in their hiring: A only hires the X-candidate in a given year if they're more than twice as good as the Y-candidate, while B only hires the X-candidate in a given year if they're more than three times as good as the Y-candidate.

Clearly this bias hurts the average quality of both departments, B more so than A. But you might intuitively expect that at least if you looked at the X-candidates that B has actually hired, they'd be of higher average quality than the X-candidates that A has hired – simply because they had to pass through a higher filter to get hired. In fact, however, this isn't the case: despite the fact that B imposes a higher filter, the calculations for this example performed above show that the average quality of the X-candidates B hires is 9, while the average quality of the X-candidates A hires is 10.

This non-monotonicity property shows that the conditional expectations we work with in the analysis can be pathologically behaved for arbitrary (even relatively simple) distributions.

However, we will see that with power law distributions we are able – with some work – to avoid these difficulties; and part of our analysis will include a set of explicit monotonicity results.

2 Biased Selection with Power Law Distributions

Recall that for a random variable Z, we use $Z_{(\ell:m)}$ to denote the ℓ^{th} order statistic in m draws from Z: the value in position ℓ when we sort m independent draws from Z from lowest to highest. Recall also that when selecting k finalists, the Rooney Rule improves expected utility exactly when

$$\mathbb{E}\left[X_{(\alpha n:\alpha n)} - Y_{(n-k+1:n)} | X_{(\alpha n:\alpha n)} < \beta Y_{(n-k+1:n)}\right] > 0.$$

Using linearity of expectation and the fact that $\Pr[A|B]\Pr[B] = \Pr[A \cdot \mathbb{1}_B]$, this is equivalent to

$$\frac{\mathbb{E}\left[X_{(\alpha n:\alpha n)} \cdot \mathbb{1}_{X_{(\alpha n:\alpha n)} < \beta Y_{(n-k+1:n)}}\right]}{\mathbb{E}\left[Y_{(n-k+1:n)} \cdot \mathbb{1}_{X_{(\alpha n:\alpha n)} < \beta Y_{(n-k+1:n)}}\right]} > 1. \tag{2}$$

We will show an asymptotically tight characterization of the tuples of parameters $(k, \alpha, \beta, \delta)$ for which this condition holds, up to an error term on the order of $O\left(\frac{(\ln n)^2}{n}\right)$. In order to better understand the terms in (2), we begin with some necessary background.

2.1 Preliminaries

▶ **Fact 4.** Let $f_{(p:m)}$ and $F_{(p:m)}$ be the pdf and cdf of the p^{th} order statistic out of m draws from the power law distribution with parameter δ. Using definitions from [8],

$$f_{(p:m)}(x) = (1+\delta)(m-p+1)\binom{m}{p-1}\left(1 - x^{-(1+\delta)}\right)^{p-1}\left(x^{-(1+\delta)}\right)^{m-p+1} x^{-1}$$

and

$$F_{(p:m)}(x) = \sum_{j=p}^{m}\binom{m}{j}\left(1 - x^{-(1+\delta)}\right)^{j}\left(x^{-(1+\delta)}\right)^{m-j}.$$

▶ **Definition 5.**

$$\Gamma(a) = \int_0^\infty t^{a-1}e^{-t}\,dt.$$

$\Gamma(\cdot)$ is considered the continuous relaxation of the factorial, and it satisfies

$$\Gamma(a+1) = a\Gamma(a).$$

If a is a positive integer, $\Gamma(a+1) = a!$. Furthermore, $\Gamma(a) > 1$ for $0 < a < 1$ and $\Gamma(a) < 1$ for $1 < a < 2$.

2.2 The Case where $k = 2$

For simplicity, we begin with the case where we're selecting $k = 2$ finalists. In this section, we will make several approximations, growing tight with large n, that we treat formally in the full version. Our analysis makes use of results from [15, 16, 19]. This section is intended

to demonstrate the techniques needed to understand the condition (2). In the case where $k = 2$, always selecting an X-candidate increases expected utility if and only if

$$\frac{\mathbb{E}\left[X_{(\alpha n:\alpha n)} \cdot \mathbb{1}_{X_{(\alpha n:\alpha n)} < \beta Y_{(n-1:n)}}\right]}{\mathbb{E}\left[Y_{(n-1:n)} \cdot \mathbb{1}_{X_{(\alpha n:\alpha n)} < \beta Y_{(n-1:n)}}\right]} > 1. \tag{3}$$

In the full version, we give tight approximations to these quantities; here, we provide an outline for how to find them. For the sake of exposition, we'll only show this for the denominator in this section, which is slightly simpler to approximate. We begin with

$$\mathbb{E}\left[Y_{(n-1:n)} \cdot \mathbb{1}_{X_{(\alpha n:\alpha n)} < \beta Y_{(n-1:n)}}\right] = \int_1^\infty y f_{(n-1:n)}(y) F_{(\alpha n:\alpha n)}(\beta y)\, dy.$$

Letting $c = \alpha\beta^{-(1+\delta)}$, with some work, we can approximate this by

$$(1+\delta)n(n-1)\int_1^\infty \left(1 - y^{-(1+\delta)}\right)^{n(1+c)-2} \left(y^{-(1+\delta)}\right)^2 dy.$$

Conveniently, the function being integrated is (up to a constant factor) $y \cdot f_{(n(1+c)-1:n(1+c))}(y)$, i.e. y times the pdf of the second-highest order statistic from $n(1+c)$ samples. Since

$$\begin{aligned}\mathbb{E}\left[Z_{(n(1+c)-1:n(1+c))}\right] &= \int_1^\infty z f_{(n(1+c)-1:n(1+c))}(z)\, dz \\ &= (1+\delta)n(1+c)(n(1+c)-1)\int_1^\infty \left(1 - z^{-(1+\delta)}\right)^{n(1+c)-2} \left(z^{-(1+\delta)}\right)^2 dz,\end{aligned}$$

we have

$$\mathbb{E}\left[Y_{(n-1:n)} \cdot \mathbb{1}_{X_{(\alpha n:\alpha n)} < \beta Y_{(n-1:n)}}\right] \approx \frac{1}{(1+c)^2}\mathbb{E}\left[Z_{(n(1+c)-1:n(1+c))}\right].$$

We can show that $\mathbb{E}\left[Z_{(n(1+c)-1:n(1+c))}\right] \approx (1+c)^{1/(1+\delta)}\mathbb{E}\left[Y_{(n-1:n)}\right]$, meaning that

$$\mathbb{E}\left[Y_{(n-1:n)} \cdot \mathbb{1}_{X_{(\alpha n:\alpha n)} < \beta Y_{(n-1:n)}}\right] \approx (1+c)^{-(1+\delta/(1+\delta))}\mathbb{E}\left[Y_{(n-1:n)}\right]. \tag{4}$$

For the numerator of (3), a slightly more involved calculation yields

$$\begin{aligned}&\mathbb{E}\left[X_{(\alpha n:\alpha n)} \cdot \mathbb{1}_{X_{(\alpha n:\alpha n)} < \beta Y_{(n-1:n)}}\right] \\ &\approx \mathbb{E}\left[X_{(\alpha n:\alpha n)}\right]\left[1 - (1+c^{-1})^{-\delta/(1+\delta)}\left[1 + \frac{\delta}{1+\delta}(1+c)^{-1}\right]\right].\end{aligned} \tag{5}$$

We can show that

$$\mathbb{E}\left[X_{(\alpha n:\alpha n)}\right] \approx \Gamma\left(\frac{\delta}{1+\delta}\right)(\alpha n)^{1/(1+\delta)} \quad \text{and} \quad \mathbb{E}\left[Y_{(n-1:n)}\right] \approx \Gamma\left(1 + \frac{\delta}{1+\delta}\right) n^{1/(1+\delta)}.$$

Recall that, up to the approximations we made, the Rooney Rule improves utility in expectation if and only if the ratio between (5) and (4) is larger than 1. Therefore, the following theorem holds:

▶ **Theorem 6.** *For sufficiently large n, the Rooney Rule with $k = 2$ improves utility in expectation if and only if*

$$\frac{\alpha^{1/(1+\delta)}\left[1 - (1+c^{-1})^{-\delta/(1+\delta)}\left[1 + \frac{\delta}{1+\delta}(1+c)^{-1}\right]\right]}{\frac{\delta}{1+\delta}(1+c)^{-1-\delta/(1+\delta)}} > 1. \tag{6}$$

where $c = \alpha\beta^{-(1+\delta)}$.

Note that in the limit as $\beta \to \infty$, $c \to 0$, and the entire expression goes to $\alpha^{1/(1+\delta)}(1+\delta)/\delta$, as noted in Section 1.3. Although this expression is complex, it can be directly evaluated, giving a tight characterization of when the Rule yields increased utility in expectation.

With this result, we could ask for a fixed α and δ how to characterize the set of β such that the condition in (6) holds. In fact, we can show that this expression is monotonically increasing in β.

▶ **Theorem 7.** *The left hand side of* (6) *is decreasing in c and therefore increasing in* β. *Hence for fixed* α *and* δ *there exists* β^* *such that* (6) *holds if and only if* $\beta > \beta^*$.

Non-monotonicity in δ

From Theorem 6, we can gain some intuition for the non-monotonicity in δ shown in Figure 1. For $\alpha < e^{-1}$, we can show that even with infinite bias, the Rooney Rule has a negative effect on utility for sufficiently large δ. Intuitively, this is because the condition for positive change with infinite bias is $\alpha > \left(\frac{\delta}{1+\delta}\right)^{1+\delta}$, which can be written as $\alpha > \left(1 - \frac{1}{d}\right)^d$ for $d = 1 + \delta$. Since this converges to e^{-1} from below, for sufficiently large δ and $\alpha < e^{-1}$, $\alpha < \left(\frac{\delta}{1+\delta}\right)^{1+\delta}$. On the other hand, as $\delta \to 0$, the Rooney Rule has a more negative effect on utility. For instance, $\phi_2(.3, 10, 1) > 1$ but $\phi_2(.3, 10, .5) < 1$. Intuitively, this non-monotonicity arises from the fact that for large δ and small α, the Rooney Rule always has a negative impact on utility, while for very small δ, samples are very far from each other, meaning that the bias has less effect on the ranking.

2.3 The General Case

We can extend these techniques to handle larger values of k. For $k \in [n]$, we define

$$r_k(\alpha, \beta, \delta) = \frac{\mathbb{E}\left[X_{(\alpha n:\alpha n)} | X_{(\alpha n:\alpha n)} < \beta Y_{(n-k+1:n)}\right]}{\mathbb{E}\left[Y_{(n-k+1:n)} | X_{(\alpha n:\alpha n)} < \beta Y_{(n-k+1:n)}\right]} = \frac{\mathbb{E}\left[X_{(\alpha n:\alpha n)} \cdot \mathbb{1}_{X_{(\alpha n:\alpha n)} < \beta Y_{(n-k+1:n)}}\right]}{\mathbb{E}\left[Y_{(n-k+1:n)} \cdot \mathbb{1}_{X_{(\alpha n:\alpha n)} < \beta Y_{(n-k+1:n)}}\right]}.$$

We can see that the Rooney Rule improves expected utility when selecting k candidates if and only if $r_k > 1$. While r_k depends on n, we will show that it is a very weak dependence: for small k, as n increases, r_k converges to a function of $(\alpha, \beta, \delta, k)$ up to a $1 + O((\ln n)^2/n)$ multiplicative factor. To make this precise, we define the following notion of asymptotic equivalence:

▶ **Definition 8.** For nonnegative functions $f(n)$ and $g(n)$, define

$$f(n) \approxeq g(n)$$

if and only if there exist $a > 0$ and $n_0 > 0$ such that

$$\frac{f(n)}{g(n)} \leq 1 + \frac{a(\ln n)^2}{n} \quad \text{and} \quad \frac{g(n)}{f(n)} \leq 1 + \frac{a(\ln n)^2}{n}$$

for all $n \geq n_0$. In other words, $f(n) = g(n)\left(1 \pm O\left(\frac{(\ln n)^2}{n}\right)\right)$. When being explicit about a and n_0, we'll write $f(n) \approxeq_{a;n_0} g(n)$.

The full version contains a series of lemmas establishing how to rigorously manipulate equivalences of this form. Now, we formally define a tight approximation to r_k, which serves as an expanded restatement of Theorem 2 from the introduction.

▶ **Theorem 9.** *For $k \in [n]$, define*

$$\phi_k(\alpha,\beta,\delta) = \frac{\alpha^{1/(1+\delta)}c^{\delta/(1+\delta)}(1+c)^{k-1}}{\binom{k-1-\frac{1}{1+\delta}}{k-1}}\left[(1+c^{-1})^{\delta/(1+\delta)} - \sum_{j=0}^{k-1}\binom{j-\frac{1}{1+\delta}}{j}(1+c)^{-j}\right] \quad (7)$$

where $c = \alpha\beta^{-(1+\delta)}$. Note that ϕ_k does not depend on n. When (α,β,δ) are fixed, we will simply write this as ϕ_k. For $k \leq ((1-c^2)\ln n)/2$, we have

$$r_k \approxeq \phi_k,$$

and therefore the Rooney Rule improves expected utility for sufficiently large n if and only if $\phi_k > 1$.

This condition tightly characterizes when the Rooney Rule improves expected utility, and its asymptotic nature in n becomes accurate even for moderately small n: for example, when $n = 50$, the error between r_k and ϕ_k is around 1%.

Increasing k

Consider the scenario in which we're selecting k candidates, and for the given parameter values, the Rooney Rule improves our expected utility. If we were to instead select $k+1$ candidates, should we still be reserving a spot for an X-candidate? Intuitively, as k increases, the Rule is less likely to change our selections, since we're more likely to have already chosen an X-candidate; however, it is not a priori obvious whether increasing k should make it better for us to use the Rooney Rule (because we have more slots, so we're losing less by reserving one) or worse (because as we take more candidates, we stop needing a reserved slot).

In fact, we can apply Theorem 9 to understand how r_k changes with k. The following theorem, proven in the full version, is an expanded restatement of Theorem 3, showing that if the Rooney Rule yields an improvement in expected quality when selecting k candidates, it will do so when selecting $k+1$ candidates as well.

▶ **Theorem 10.** *For $k \leq ((1-c^2)\ln n)/2$, we have $\phi_{k+1} > \phi_k$, and therefore for sufficiently large n, we have $r_{k+1} > r_k$.*

Finally, using these techniques, we can provide a tight characterization of the probability that the Rooney Rule produces a positive change. Specifically, find the probability that the Rooney Rule has a positive effect conditioned on the event that it changes the outcome.

▶ **Theorem 11.**

$$\Pr\left[X_{(\alpha n:\alpha n)} > Y_{(n-k+1:n)} | X_{(\alpha n:\alpha n)} < \beta Y_{(n-k+1:n)}\right] \approxeq 1 - \left(\frac{1+\alpha\beta^{-(1+\delta)}}{1+\alpha}\right)^k.$$

Note that in the case of infinite bias, this becomes $1-(1+\alpha)^{-k}$, and thus, the Rooney Rule produces positive change with probability at least $1/2$ if and only if $\alpha \geq \sqrt[k]{2}-1$. It is interesting to note that the infinite bias case here is independent of δ, while when considering expectations instead as we did in Section 1.3, the expected change in utility due to the Rooney Rule did depend on δ.

2.4 Maximum Likelihood Estimation of β

The techniques established thus far make it possible to answer other related questions, including the following type of question that we consider in this section: "Given some historical data on past selections, can we estimate the bias present in the data?" For example, suppose that for the last m years, a firm has selected one candidate for each year i out of a pool of $\alpha_i n_i$ X-candidates and n_i Y-candidates. If all applicants are assumed to come from the same underlying distribution, then it is easy to see that the expected number of X-selections should be

$$\sum_{i=1}^{m} \frac{\alpha_i}{1+\alpha_i},$$

regardless of what distribution the applicants come from. However, if there is bias in the selection procedure, then this quantity now depends on the bias model and parameters of the distribution. In particular, in our model, we can use results from the full version to get

$$\Pr\left[X_{(\alpha n:\alpha n)} < \beta Y_{(n:n)}\right] \approxeq \frac{1}{1+\alpha\beta^{-(1+\delta)}}.$$

This gives us the following approximation for the likelihood of the data $D = (M_1, \ldots, M_m)$ given β, where M_i is 1 if an X-candidate was selected in year i and 0 otherwise:

$$\prod_{i=1}^{m} (1-M_i) \cdot \frac{1}{1+\alpha_i\beta^{-(1+\delta)}} + M_i \cdot \frac{\alpha_i\beta^{-(1+\delta)}}{1+\alpha_i\beta^{-(1+\delta)}}.$$

Taking logs, this is

$$\sum_{i:M_i=1} \log(\alpha_i\beta^{-(1+\delta)}) - \sum_{i=1}^{m} \log(1+\alpha_i\beta^{-(1+\delta)}),$$

and maximizing this is equivalent to maximizing

$$\sum_{i:M_i=1} \log(\beta^{-(1+\delta)}) - \sum_{i=1}^{m} \log(1+\alpha_i\beta^{-(1+\delta)}) = N\log(\beta^{-(1+\delta)}) - \sum_{i=1}^{m} \log(1+\alpha_i\beta^{-(1+\delta)})$$

where N is the number of X-candidates selected. Taking the derivative with respect to β, we get

$$-(1+\delta)N\beta^{-1} + (1+\delta)\sum_{i=1}^{m} \frac{\alpha_i\beta^{-(2+\delta)}}{1+\alpha_i\beta^{-(1+\delta)}}.$$

Setting this equal to 0 and canceling common terms, we have

$$\sum_{i=1}^{m} \frac{1}{1+\alpha_i^{-1}\beta^{1+\delta}} = N$$

Since each $1/(1+\alpha_i^{-1}\beta^{1+\delta})$ is strictly monotonically decreasing in β, there is a unique $\hat{\beta}$ for which equality holds, meaning that the likelihood is uniquely maximized by $\hat{\beta}$, up to the $1 \pm O((\ln n)^2/n)$ approximation we made for $\Pr\left[X_{(\alpha n:\alpha n)} < \beta Y_{(n:n)}\right]$. In the special case where $\alpha_i = \alpha$ for $i = 1, \ldots, m$, then the solution is given by

$$\hat{\beta} = \left(\left(\frac{m}{N} - 1\right)\alpha\right)^{1/(1+\delta)}.$$

3 Biased Selection with Bounded Distributions

In this section, we consider a model in which applicants come from from a distribution with bounded support. Qualitatively, one would expect different results here from those with power law distributions because in a model with bounded distributions, we expect that for large n, the top order statistics of any distribution will concentrate around the maximum of that distribution. As a result, when there is even a small amount of bias against one population, for large n the probability that *any* of the samples with the highest perceived quality come from that population goes to 0. This means that the Rooney Rule has an effect with high probability, and the effect is positive if the unconditional expectation of the top X-candidate is larger than the unconditional expectation of the Y-candidate that it replaces.

We focus on the case when $\alpha = 1$, meaning we have equal numbers of applicants from both populations. We use the same order statistic notation as before. While all of our previous results have modeled the bias as a multiplicative factor β, we can in fact show that in the bounded distribution setting, for any model of bias $\tilde{X}_{(k:n)} = b(X_{(k:n)})$ such that $b(x) < T$ for $x \geq 0$, where T is strictly less than the maximum of the distribution, the Rooney Rule increases expected utility. Unlike in the previous section the following theorem and analysis are by no means a tight characterization; instead, this is an existence proof that for bounded distributions, there is always a large enough n such that the Rooney Rule improves utility in expectation. We prove our results for continuous distributions with support $[0, 1]$, but a simple scaling argument shows that this extends to any continuous distribution with bounded nonnegative support – specifically, we scale a distribution such that $\inf_{x:f(x)>0} = 0$ and $\sup_{x:f(x)>0} = 1$.

▶ **Theorem 12.** *If f is continuous probability density function on $[0, 1]$ such that $\sup_{x:f(x)>0} = 1$ and $\tilde{X}_{(n:n)} = b(X_{(n:n)})$ is never more than $T < 1$, then for large enough n,*

$$\mathbb{E}\left[X_{(n:n)} - Y_{(n-1:n)} | b(X_{(n:n)}) < Y_{(n-1:n)}\right] > 0.$$

While we the defer the full proof to the full version, the strategy for the proof is as follows:

1. With high probability, $X_{(n:n)}$ and $Y_{(n-1:n)}$ are both large.
2. Whenever $X_{(n:n)}$ and $Y_{(n-1:n)}$ are large, $X_{(n:n)}$ is significantly larger than $Y_{(n-1:n)}$.
3. The gain from switching from $Y_{(n-1:n)}$ to $X_{(n:n)}$ when $X_{(n:n)}$ and $Y_{(n-1:n)}$ are both large outweighs the loss when at least one of them is not large.

4 Conclusion

In this work we have presented a model for implicit bias in a selection problem motivated by settings including hiring and admissions, and we analyzed the Rooney Rule, which can in fact improve the quality of the resulting choices. For one of the most natural settings of the problem, when candidates are drawn from a power-law distribution, we found a number of counter-intuitive effects at work, which we believe help provide better insight into how we might reason about the effects of implicit bias. Our techniques also provided a natural solution to an inference problem in which we estimate parameters of a biased decision-making process. Finally, we performed a similar type of analysis on general bounded distributions.

Our framework makes it possible to naturally explore extensions in a number of further directions. First, the model can be generalized to include noisy observations, potentially with a different level of noise for each group. It would also be interesting to analyze generalizations of the Rooney Rule; for example, if we were to define the *ℓ^{th}-order Rooney Rule* to be the requirement that at least ℓ of k finalists must be from an underrepresented group, we could

ask which ℓ produces the greatest increase in utility for a given set of parameters. Finally, we could benefit from a deeper undestanding of the function ϕ. For example, while we showed in Theorem 3 that ϕ is monotone in β for $k = 2$, we also demonstrated that ϕ is not monotone in δ. A better understaning of the function ϕ may lead to new insights into our model and into the phenomena it seeks to capture.

Acknowledgements. We thank Eric Parsonnet for his invaluable technical insights.

References

1 Marianne Bertrand and Sendhil Mullainathan. Are Emily and Greg more employable than Lakisha and Jamal? A field experiment on labor market discrimination. *American Economic Review*, 94(4):991–1013, 2004.

2 Iris Bohnet, Alexandra van Geen, and Max Bazerman. When performance trumps gender bias: Joint vs. separate evaluation. *Management Science*, 62(5):1225–1234, 2016.

3 Mark Braverman and Elchanan Mossel. Sorting from noisy information. *arXiv preprint arXiv:0910.1191*, 2009.

4 Marilyn Cavicchia. How to fight implicit bias? With conscious thought, diversity expert tells NABE. *American Bar Association: Bar Leader*, 40(1), 2015.

5 Aaron Clauset, Cosma R. Shalizi, and Mark E. J. Newman. Power-law distributions in empirical data. *SIAM Review*, 51(4):661–703, 2009.

6 Brian W. Collins. Tackling unconscious bias in hiring practices: The plight of the Rooney Rule. *NYU Law Review*, 82(3), 2007.

7 Covington and Burling. Recommendations to Uber, 13 June 2017.

8 H. A. David and H. N. Nagaraja. *Basic Distribution Theory*, pages 9–32. John Wiley & Sons, Inc., 2005.

9 Uriel Feige, Prabhakar Raghavan, David Peleg, and Eli Upfal. Computing with noisy information. *SIAM Journal on Computing*, 23(5):1001–1018, 1994.

10 Qiang Fu and Jingfeng Lu. Micro foundations of multi-prize lottery contests: a perspective of noisy performance ranking. *Social Choice and Welfare*, 38(3):497–517, 2012.

11 Alexander R. Green, Dana R. Carney, Daniel J. Pallin, Long H. Ngo, Kristal L. Raymond, Lisa I. Iezzoni, and Mahzarin R. Banaji. Implicit bias among physicians and its prediction of thrombolysis decisions for black and white patients. *Journal of General Internal Medicine*, 22(9):1231–1238, 2007.

12 Anthony G. Greenwald and Mahzarin R. Banaji. Implicit social cognition: attitudes, self-esteem, and stereotypes. *Psychological Review*, 102(1):4–27, 1995.

13 Anthony G. Greenwald and Linda Hamilton Krieger. Implicit bias: Scientific foundations. *California Law Review*, 94:945–967, 2006.

14 Christine Jolls and Cass R. Sunstein. The law of implicit bias. *California Law Review*, 94:969–996, 2006.

15 Manuel Lopez and James Marengo. An upper bound for the expected difference between order statistics. *Mathematics Magazine*, 84(5):365–369, 2011.

16 Henrick John Malik. Exact moments of order statistics from the pareto distribution. *Scandinavian Actuarial Journal*, 1966(3-4):144–157, 1966.

17 Christina Passariello. Tech firms borrow football play to increase hiring of women. *Wall Street Journal*, 27 September 2016.

18 Hamza Shaban. What is the “Rooney Rule" that Uber just adopted? *Washington Post*, 13 June 2017.

19 Francesco Giacomo Tricomi and Arthur Erdélyi. The asymptotic expansion of a ratio of gamma functions. *Pacific J. Math*, 1(1):133–142, 1951.

20 Eric Luis Uhlmann and Geoffrey L. Cohen. Constructed criteria: Redefining merit to justify discrimination. *Psychological Science*, 16(6):474–480, 2005.

21 Linda van den Bergh, Eddie Denessen, Lisette Hornstra, Marinus Voeten, and Rob W. Holland. The implicit prejudiced attitudes of teachers: Relations to teacher expectations and the ethnic achievement gap. *American Education Research Journal*, 47(2):497–527, 2010.

22 Christine Wenneras and Agnes Wold. Nepotism and sexism in peer-review. *Nature*, 387:341–343, 1997.

Fine-grained I/O Complexity via Reductions: New Lower Bounds, Faster Algorithms, and a Time Hierarchy*

Erik D. Demaine[1], Andrea Lincoln[2], Quanquan C. Liu[3], Jayson Lynch[4], and Virginia Vassilevska Williams[5]

1 **Computer Science and Artificial Intelligence Lab, Massachusetts Institute of Technology, Cambridge, MA, USA**
edemaine@mit.edu
2 **Computer Science and Artificial Intelligence Lab, Massachusetts Institute of Technology, Cambridge, MA, USA**
andreali@mit.edu
3 **Computer Science and Artificial Intelligence Lab, Massachusetts Institute of Technology, Cambridge, MA, USA**
quanquan@mit.edu
4 **Computer Science and Artificial Intelligence Lab, Massachusetts Institute of Technology, Cambridge, MA, USA**
jaysonl@mit.edu
5 **Computer Science and Artificial Intelligence Lab, Massachusetts Institute of Technology, Cambridge, MA, USA**
virgi@mit.edu

Abstract

This paper initiates the study of I/O algorithms (minimizing cache misses) from the perspective of fine-grained complexity (conditional polynomial lower bounds). Specifically, we aim to answer why sparse graph problems are so hard, and why the Longest Common Subsequence problem gets a savings of a factor of the size of cache times the length of a cache line, but no more. We take the reductions and techniques from complexity and fine-grained complexity and apply them to the I/O model to generate new (conditional) lower bounds as well as faster algorithms. We also prove the existence of a time hierarchy for the I/O model, which motivates the fine-grained reductions.

- Using fine-grained reductions, we give an algorithm for distinguishing 2 vs. 3 diameter and radius that runs in $O(|E|^2/(MB))$ cache misses, which for sparse graphs improves over the previous $O(|V|^2/B)$ running time.
- We give new reductions from radius and diameter to Wiener index and median. These reductions are new in both the RAM and I/O models.
- We show meaningful reductions between problems that have linear-time solutions in the RAM model. The reductions use low I/O complexity (typically $O(n/B)$), and thus help to finely capture the relationship between "I/O linear time" $\Theta(n/B)$ and RAM linear time $\Theta(n)$.
- We generate new I/O assumptions based on the difficulty of improving sparse graph problem running times in the I/O model. We create conjectures that the current best known algorithms for Single Source Shortest Paths (SSSP), diameter, and radius are optimal.
- From these I/O-model assumptions, we show that many of the known reductions in the word-RAM model can naturally extend to hold in the I/O model as well (e.g., a lower bound on the I/O complexity of Longest Common Subsequence that matches the best known running time).

* A full version of the paper is available at https://arxiv.org/abs/1711.07960.

9th Innovations in Theoretical Computer Science Conference (ITCS 2018).
Editor: Anna R. Karlin; Article No. 34; pp. 34:1–34:23

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

- We prove an analog of the Time Hierarchy Theorem in the I/O model, further motivating the study of fine-grained algorithmic differences.

1998 ACM Subject Classification B.3.2 Shared memory algorithms

Keywords and phrases IO model, Fine-grained Complexity, Algorithms

Digital Object Identifier 10.4230/LIPIcs.ITCS.2018.34

1 Introduction

The I/O model (or external-memory model) was introduced by Aggarwal and Vitter [5] to model the non-uniform access times of memory in modern processors. The model nicely captures the fact that, in many practical scenarios, cache misses between levels of the memory hierarchy (including disk) are the bottleneck for the program. As a result, the I/O model has become a popular model for developing cache-efficient algorithms.

In the I/O model, the expensive operation is bringing a *cache line* of B contiguous words from the "main memory" (which may alternately represent disk) to the "cache" (local work space). The cache can store up to M words in total, or M/B cache lines. Computation on data in the cache is usually treated as free, and thus the main goal of I/O algorithms is to access memory with locality. That is, when bringing data into cache from main memory in contiguous chunks, we would like to take full advantage of the fetched cache line. This is preferable to, say, randomly accessing noncontiguous words in memory.

When taking good graph algorithms for the RAM model and analyzing them in the I/O model, the running times are often very bad. Take for example Dijkstra's algorithm or the standard BFS algorithm. These algorithms fundamentally look at an adjacency list, and follow pointers to every adjacent node. Once the new node is reached, the process repeats, accessing all the adjacent nodes in priority order that have not previously been visited. This behavior looks almost like random access! Unless one can efficiently predict the order these nodes will be reached, the nodes will likely be stored far apart in memory. Even worse, this optimal order could be very different depending on what node one starts the algorithm at.

Because of this bad behavior, I/O-efficient algorithms for graph problems take a different approach. For dense graphs, one approach is to reduce the problems to matrix equivalent versions. For example, APSP is solved by $(\min, +)$ matrix multiplication [34, 37]. The locality of matrices leads to efficient algorithms for these problems.

Unfortunately, sparse graph problems are not solved efficiently by $(\min, +)$ matrix multiplication. For example, the best algorithms for directed single-source shortest paths in sparse graphs take $O(n)$ time, giving no improvement from the cache line at all [16, 18, 8]. Even in the undirected case, the best algorithm takes $O(n/\sqrt{B})$ time in sparse graphs [32].

The Diameter problem in particular has resisted improvement beyond $O(|V|/\sqrt{B})$ even in undirected unweighted graphs [30], and in directed graphs, the best known algorithms still run in time $\Omega(|V|)$ [17, 8]. For this reason, we use this as a conjecture and build a network of reductions around sparse diameter and other sparse graph problems.

In this paper we seek to explain why these problems, and other problems in the I/O model, are so hard to improve and to get faster algorithms for some of them.

In this paper we use reductions to generate new algorithms, new lower bounds, and a time hierarchy in the I/O model. Specifically, we get new algorithms for computing the diameter and radius in sparse graphs, when the computed radii are small. We generate novel reductions (which work in both the RAM and I/O models) for the Wiener Index problem (a

graph centrality measure). We generate further novel reductions which are meaningful in the I/O model related to sparse graph problems. Finally, we show that an I/O time hierarchy exists, similar to the classic Time Hierarchy Theorem.

1.1 Caching Model and Related Work

Cache behavior has been studied extensively. In 1988, Aggarwal and Vitter [5] developed the *I/O model*, also known as the external-memory model [25], which now serves as a theoretical formalization for modern caching models. A significant amount of work on algorithms and data structures in this model has occurred including items like buffer-trees [7], B-trees [13], permutations and sorting [5], ordered file maintenance [14], $(\min, +)$ matrix multiplication [34], and triangle listing [33]. Frigo, Leiserson, Prokop and Ramachandran [27] proposed the *cache-oblivious model*. In this model, the algorithm is not given access to the cache size M, nor is it given access to the cache-line size B. Thus the algorithm must be oblivious to the cache, despite being judged on its cache performance. Some surveys of the work include [42, 6, 25].

When requesting cache lines from main memory in this paper we will only request the B words starting at location xB for integers x. Another common model (which we do not follow in this paper) allows arbitrary offsets for the cache line pulls. This can be simulated with at most twice as many cache misses and twice as much cache.

1.2 Fine-grained Complexity

A popular area of recent study is fine-grained complexity. The field uses efficient reductions to uncover relationships between problems whose classical algorithms have not been improved substantially in decades. Significant progress has been made in explaining the lack of progress on many important problems [10, 44, 2, 3, 15, 1, 36] such as APSP, othogonal vectors (OV), 3-SUM, longest common subsequence (LCS), edit distance and more. Such results focused on finding reductions from these problems to other (perhaps less well-studied) problems such that an improvement in the upper bound on any of these problems will lead to an improvement in the running time of algorithms for these problems. For example, research around the All-Pairs Shortest Paths problem (APSP) has uncovered that many natural, seemingly simpler graph problems on n node graphs are fine-grained equivalent to APSP, so that an $O(n^{3-\varepsilon})$ time algorithm for $\varepsilon > 0$ for one of the problems implies an $O(n^{3-\varepsilon'})$ time algorithm for some $\varepsilon' > 0$ for all of them.

1.3 History of Upper Bounds

In the I/O model, the design of algorithms for graph problems is difficult. This is demonstrated by the number of algorithms designed for problems like Sparse All Pairs Shortest Paths, Breath First Search, Graph Radius and Graph Diameter where very minor improvements are made (see Table 1 for definitions of problems). Note that dense and sparse qualifiers in the front of problems indicate the problem defined over a dense/sparse graph, respectively.

The Wiener index problem measures the total distance from all points to each other. Intuitively, this measures how close or far points in the graph are from each other. In this respect, Wiener index is similar to the radius, diameter and median measures of graph distance.

The history of improvements to the upper bound of negative triangle in the I/O model is an important example of the difficulty in the design of I/O efficient algorithms for graph

Table 1 Fine-grained problems definitions.

Problem Name	Problem Definition
Orthogonal Vector (OV)	Given two sets U and V of n vectors each with elements $\{0,1\}^d$ where $d = \omega(\log n)$, determine whether there exist vectors $u \in U$ and $v \in V$ such that $\sum_{i=1}^{d} u_i \cdot v_i = 0$.
Longest Common Subsequence (LCS)	Given two strings of n symbols over some alphabet Σ, compute the length of the longest sequence that appears as a subsequence in both input strings.
Edit Distance (ED)	Given two strings s_1 and s_2, determine the minimum number of operations that converts s_1 to s_2.
Sparse Diameter	Given a sparse graph $G = (V, E)$, determine if $\max_{u,v \in V} d(u, v)$ where $d(u, v)$ is the distance between nodes u and v in V.
2 vs. 3 Sparse Diameter	Given a sparse graph $G = (V, E)$, determine if $\max_{u,v \in V} d(u, v) \leq 2$ where $d(u, v)$ is the distance between nodes u and v in V.
Hitting Set (HS)	Given two lists of size n, V and W, where the elements are taken from a universe U, does there exist a set in V that hits (contains an element of) every set in W.
Sparse Radius	Given a sparse graph $G = (V, E)$, determine $\min_{u \in V} (\max_{v \in V} d(u, v))$ where $d(u, v)$ is the distance between nodes u and v in V.
2 vs. 3 Sparse Radius	Given a sparse graph $G = (V, E)$, determine if $\min_{u \in V} (\max_{v \in V} d(u, v)) \leq 2$ where $d(u, v)$ is the distance between nodes u and v in V.
3 vs. 4 Sparse Radius	Given a sparse graph $G = (V, E)$, determine if $\min_{u \in V} (\max_{v \in V} d(u, v)) \leq 3$ where $d(u, v)$ is the distance between nodes u and v in V.
Median	Let $d(u, v)$ be the shortest path distance between nodes u and v in a graph G. The median is the node v that minimizes the sum $\sum_{u \in V} d(v, u)$.
3-SUM	Given a set of n integers, determine whether the set contains three integers a, b, c such that $a + b = c$.
Convolutional 3-SUM	Given three lists A, B and C each consisting of n numbers, return true if $\exists i, j, k \in [0, n-1]$ such that $i + j + k \equiv 0 \textbf{ mod } n$ and $A[i] + B[j] + C[k] = 0$.
0 Triangle	Given a graph G, return true if $\exists a, b, c \in V$ such that $w(a, b) + w(b, c) + w(c, a) = 0$ where $w(u, v)$ is the weight of the edge (u, v).
All-Pairs Shortest Paths (APSP)	Given a directed or undirected graph with integer weights, determine the shortest distance between all pairs of vertices in the graph.
Wiener Index	Let $d(u, v)$ be the shortest paths distance between nodes u and v in a graph G. The Wiener Index of G is $\sum_{u \in V} \sum_{v \in V} d(u, v)$.
Negative Traingle	Given a graph G, return true if $\exists a, b, c \in V$ such that $w(a, b) + w(b, c) + w(c, a) < 0$ where $w(u, v)$ is the weight of the edge (u, v).
$(\min, +)$-Matrix Multiplication	Given two n by n matrices A and B, return $C[i, k] = \min_{j \in [1,n]} (A[i, j] + B[j, k])$.
Sparse Weighted Diameter	Given a sparse, weighted graph $G = (V, E)$, determine $\max_{u,v \in V} d(u, v)$ where $d(u, v)$ is the distance between nodes u and v in V.

Table 2 History of APSP upper bounds. * This is an extension of [29] see e.g. [33].

Dense APSP		Sparse Weighted Diameter		Sparse-△	
$\tilde{O}(n^3)$	[naive]	$\tilde{O}(n^2)$	[naive]	$\tilde{O}(n^{1.5})$	[naive]
$\tilde{O}(n^3/(\sqrt{M}))$	[29]	$\tilde{O}(n^2/\sqrt{B})$	[9]	$\tilde{O}(n+n^{1.5}/B)$	[31]
$\tilde{O}(n^3/(\sqrt{MB}))$	Extension*	$\tilde{O}(n^2/\sqrt{B})$	[19]	$\tilde{O}(n^{1.5}/B)$	[26]
				$\tilde{O}(n^{1.5}/(\sqrt{MB}))$	[33]

problems (see Table 2 for a summary). For a long time, no improvements in terms of M were made to the upper bound for negative triangle. A key was re-interpreting the problem as a repeated scan of lists.

We feel that the history of negative triangle has taught us that upper bounds in the I/O model on graph problems are best achieved by creating efficient reductions from a graph problem to a non-graph problem. Hence the study of fine-grained reductions in the I/O model is crucial to using this approach in solving such graph problems with better I/O efficiency. Graph problems tempt the algorithmic designer into memory access patterns that look like random access, whereas matrix and array problems immediately suggest memory local approaches to these problems. We consider the history of the negative triangle problem to be an instructive parable of why the matrix and array variants are the right way to view I/O problems.

1.4 Our Results

We will now discuss our results in this paper. In all tables in this paper, our results will be in bold. We demonstrate the value of reductions as a tool to further progress in the I/O model through our results.

Our results include improved upper bounds, a new technique for lower bounds in the I/O model and the proof of a computational hierarchy. Notably, in this paper we tie the I/O model into both fine-grained complexity and classical complexity.

1.4.1 Upper Bounds

We get improved upper bounds on two sparse graph problems and have a clarifying note to the community about matrix multiplication algorithms.

For both the sparse 2 vs 3 diameter and sparse 2 vs 3 radius problems, we improve the running time from $O(n^2/B)$ to $O(n^2/(MB))$. We get these results by using an insight from a pre-existing reduction to two very local problems which have trivial $O(n^2/(MB))$ algorithms that solve them. Note that this follows the pattern we note in Section 1.3 in that we produce a reduction from a graph problem to a non-graph problem to obtain better upper bounds in terms of M.

Furthermore, previous work in the I/O model related to matrix multiplication seems to use the naive matrix multiplication n^3 bound, or the Strassen subdivision. However, fast matrix multiplication algorithms which runs in n^ω time imply a nice self-reduction. Thus, we can get better I/O algorithms which run in the most recent fast matrix multiplication time. We want to explicitly add a note in the literature that fast matrix multiplication in the I/O model should run in time $T_{MM}(n, M, B) = O(n^{\omega'}/(M^{\omega'/2-1}B))$ where ω' is the matrix

multiplication exponent, if it is derived using techniques bounding the rank of the matrix multiplication tensor. The current best ω' is $\omega' < 2.373$ [41, 28] giving us the I/O running time of $T_{MM}(n, M, B) = O(n^{2.373}/(M^{0.187}B))$. We give these results in Section 2.3.

1.4.2 I/O model Conjectures

In the I/O model a common way to get upper bounds is to get a self-reduction where a large problem is solvable by a few copies of a smaller problem. We make the small subproblems so small they fit in cache. If the problem is laid out in a memory local fashion in main memory then it will take M/B I/Os to solve a subproblem that fits in memory M.

In Section 2.2, we give an I/O-based Master Theorem which gives the running time for algorithms with recurrences of the form $T(n, M, B) = \alpha T(n/\beta, M, B) + f(n, M, B)$ (like the classic Master Theorem from [24]) and $T(n, M, B) = g^2 T(n/\beta, M, B) + f(n, M, B)$ (self-reduction). The running times generated by these recurrences match the best known running times of All-Pairs Shortest Paths (APSP), 3-SUM, Longest Common Subsequence (LCS), Edit Distance, Orthogonal Vectors (OV), and more. Thus, if we conjecture that a recursive algorithm has a running time that is optimal for a problem, we are able to transfer this bound over to the I/O model using our Master Theorem and self-reduction framework in a natural way.

1.4.2.1 Lower Bounds From Fine-Grained Complexity Assumptions

We demonstrate that many of the reductions in the RAM model between problems of interest and common fine-grained assumptions give lower bounds in the I/O model. We generate reasonable I/O conjectures for these problems and demonstrate that the reductions are I/O-efficient. First, we begin with the conjectures.

▶ **Conjecture 1** (I/O All-Pairs Shortest Paths (APSP) Conjecture). *APSP requires* $\frac{n^{3-o(1)}}{M^{1/2+o(1)}B^{1+o(1)}}$ *I/Os.*

▶ **Conjecture 2** (I/O 3-SUM Conjecture). *3-SUM requires* $\frac{n^{2-o(1)}}{M^{1+o(1)}B^{1+o(1)}}$ *I/Os.*

▶ **Conjecture 3** (I/O Orthogonal Vectors (OV) Conjecture). *OV requires* $\frac{n^{2-o(1)}}{M^{1+o(1)}B^{1+o(1)}}$ *I/Os.*

▶ **Conjecture 4** (I/O Hitting Set (HS) Conjecture). *HS requires* $\frac{n^{2-o(1)}}{M^{1+o(1)}B^{1+o(1)}}$ *I/Os.*

From these conjectures we can generate many lower bounds. Many of our lower bounds are tight to the fastest known algorithms. These reductions have value even if the conjectures are refuted since many of these reductions also give upper bounds for other problems–leading to better algorithms for many problems even if the conjectures are refuted.

1.4.3 Lower Bounds from Sparse Graph Problems

In addition to the upper, lower bounds, and reductions presented in the I/O model for the standard RAM problems listed in Table 3, we introduce novel upper, lower bounds, and reductions between graph problems. The reason for this focus is the fact that, more than in the RAM model, the I/O model has a history of particularly slow algorithms in graphs. In particular, sparse graph problems have very slow algorithms. We make novel reductions between sparse graph problems, many of which apply to the RAM model as well, such that solving one of these problems will solve many other variations of hard sparse graph problems in the I/O model.

Table 3 Previous results and our results on upper and lower bounds of problems. Sparse (Sprs.) means that $|E| = O(|V|)$.

Problem	Upper Bound	UB source	Lower Bound	LB from	LB source
OV	$\tilde{O}(n^2/(MB))$	**Lem 49**	$\tilde{\Omega}(n^2/(MB))$	IO OV Conj	By Def
LCS	$\tilde{O}(n^2/(MB))$	[20]	$\tilde{\Omega}(n^2/(MB))$	IO OV Conj	**Lem 52**
Edit Distance	$\tilde{O}(n^2/(MB))$	[21]	$\tilde{\Omega}(n^2/(MB))$	IO OV Conj	**Lem 51**
Sparse Diameter	$\tilde{O}(n^2/B)$	[9]	$\tilde{\Omega}(n^2/(MB))$	IO OV Conj	**Lem 50**
2 vs. 3 Sprs. Diameter	$\tilde{O}(n^2/(MB))$	**Full Paper**	$\tilde{\Omega}(n^2/(MB))$	IO OV Conj	**Lem 50**
Hitting Set	$\tilde{O}(n^2/(MB))$	**Lem 54**	$\tilde{\Omega}(n^2/(MB))$	IO HS Conj	By Def.
Sparse Radius	$\tilde{O}(n^2/B)$	[9]	$\tilde{\Omega}(n^2/(MB))$	IO HS Conj	**Lem 55**
2 vs. 3 Sparse Radius	$\tilde{O}(n^2/(MB))$	**Full Paper**	$\tilde{\Omega}(n^2/(MB))$	IO HS Conj	**Lem 55**
3 vs. 4 Sparse Radius	$\tilde{O}(n^2/B)$	[9]	$\tilde{\Omega}(n^2/(MB))$	IO HS Conj	**Lem 55**
Sparse Median	$\tilde{O}(n^2/B)^a$	[9]	$\tilde{\Omega}(n^2/(MB))$	IO HS Conj	**Thm 17**
Sparse Median	$\tilde{O}(n^2/B)^b$	[9]	$\tilde{\Omega}(n^2/B)$	3 vs. 4 Sprs. Radius	**Thm 17**
3-SUM	$\tilde{O}(n^2/(MB))$	[12]	$\tilde{\Omega}(n^2/(MB))$	IO 3-SUM Conj	By Def
Conv. 3-SUM	$\tilde{O}(n^2/(MB))^c$	[12]	$\tilde{\Omega}(n^2/(MB))$	IO 3-SUM Conj	**Lem 30**
0 Triangle	$\tilde{O}(n^3/(\sqrt{M}B))$	**Lem 48**	$\tilde{\Omega}(n^2/(MB))$	IO 3-SUM Conj	**Thm 33**
APSP	$\tilde{O}(n^3/(\sqrt{M}B))$	[34]	$\tilde{\Omega}(n^2/(\sqrt{M}B))$	IO APSP Conj	By Def
Wiener Index	$\tilde{O}(n^3/(\sqrt{M}B))^d$	[34]	$\tilde{\Omega}(n^2/(\sqrt{M}B))$	IO APSP Conj	**Thm 18**
0 Triangle	$\tilde{O}(n^3/(\sqrt{M}B))$	**Lem 48**	$\tilde{\Omega}(n^2/(\sqrt{M}B))$	IO APSP Conj	**Thm 47**
− Triangle	$\tilde{O}(n^3/(\sqrt{M}B))$	**Thm 45**	$\tilde{\Omega}(n^2/(\sqrt{M}B))$	IO APSP Conj	**Thm 40**
(min,+) MM	$\tilde{O}(n^3/(\sqrt{M}B))$	[34]	$\tilde{\Omega}(n^2/(\sqrt{M}B))$	IO APSP Conj	**Thm 40**

[a] Upper bound comes from APSP in general graphs.
[b] Upper bound comes from APSP in general graphs.
[c] The upper bound is a trivial extension of the 3-SUM upper bound for explanation see Lemma 34.
[d] This upper bound comes directly from applying the algorithm for APSP, solving APSP, and summing the results.

We provide reductions between problems that currently require $\Omega(n/\sqrt{B})$ time to solve. Thus, these problems specifically require linear time reductions. We show equivalence between the following set of problems for undirected/directed and unweighted graphs: (s,t)-shortest path, finding the girth through an edge, and finding the girth through a vertex.

We additionally generate a new reduction from sparse weighted Diameter to the sparse Wiener Index problem in Section 3.1. This reduction holds in the RAM model as well as the I/O model.

1.4.4 Hierarchy

The time and space hierarchy theorems are fundamental results in computational complexity that tell us there are problems which can be solved on a deterministic Turing Machine with some bounded time or space, which cannot be solved on a deterministic Turing Machine which has access to less time or space. See notably the famous time and space hierarchies [38]. For some classes, for example BPP, no time hierarchy is known to exist (e.g., [43, 11]).

In Section 5.3, we show similar separation hierarchies exist in the I/O model once again using the simulations between the RAM and I/O models and our complexity class $CACHE_{M,B}(t(n))$ defined in Section 5.2 as the set of problems solvable in $O(t(n))$ cache misses.

▶ **Theorem 5.** *If the memory used by the algorithm is referenceable by $O(B)$ words (i.e. the entire input can be brought into cache by bringing in at most $O(B)$ words), then*

$$CACHE_{M,B}(t(n)) \subsetneq CACHE_{M,B}((t(n)\,B)^{1+\varepsilon}).$$

Notably, this theorem applies any time we use a polynomially size memory and our word size is $w = \Omega(\lg n)$, which is the standard case in the RAM model.

This separation is motivation for looking at complexity of specific problems and trying to understand what computational resources are necessary to solve them.

1.4.5 Improved TM Simulations of RAM Imply Better Algorithms

In the full version, we show that improved simulations of RAM machines by Turing Machines would imply better algorithms in the I/O model. Specifically, if we can simulate RAM more efficiently with either multi-tape Turing machines or multi-dimensional Turing machines, then we can show that we can gain some cache locality and thus save by some factor of B, the cache line size.

1.5 Organization

In this paper, we argue that the lens of reductions offer a powerful way to view the I/O model. We show that reductions give novel upper and lower bounds. We also define complexity classes for the I/O model and prove a hierarchy theorem further motivating the analysis of the I/O model using fine-grained complexity.

We begin with faster algorithms obtained through reductions which are collected in Section 2. Section 2.1 develops such algorithms for small diameter and radius. Section 2.2 develops the I/O Master Theorem, which is more broadly a useful tool for analyzing almost all cache-oblivious algorithms. Section 2.3 uses this theorem to show how all recent improvements to matrix multiplication's RAM running time also give efficient cache-oblivious algorithms.

One can get new lower bounds, by using the techniques from fine-grained complexity. Some fine-grained reductions from the RAM model also work in the I/O model, we show examples in Sections 4.1, 4.2, 4.3. We also get new reductions that work in both the RAM and I/O model related to the Wiener Index problem in Section 3.1. Some reductions in the RAM model do not work in the I/O model; thus, in Section 3.1, we give novel reductions between several algorithms which take $O(n^2/B)$ and $O(n^2/(MB))$ time. We also get reductions that are meaningful in the I/O model which are not in the RAM model, notably, between problems whose fastest algorithms are $O(n/\sqrt{B})$ and $O(n)$, respectively, in Section 3.2.

One can also use reductions and simulation arguments to prove a hierarchy theorem for the I/O model, explained in Section 5.

2 Algorithms in the I/O Model

In this section, we discuss our improved algorithms, algorithm analysis tools, and how reductions generate algorithms. As is typical in the I/O model, we assume that all inputs

are stored in disk and any computation done on the inputs are done in cache (after some or all of the inputs are brought into cache). Section 2.1 gives better algorithms for the 2 vs 3 Diameter problem and the 2 vs 3 Radius problem in the I/O model. Section 2.3 gives improved algorithms for Matrix Multiplication in the I/O model.

Self-reductions are commonly used for cache-oblivious algorithms, because dividing until the subproblems are arbitrarily small allows for the problems to always fit in cache. In the RAM model, self-reductions allow for easy analysis via the Master Theorem. Despite the amount of attention to analyzing self-reductions in the I/O model, no one has written down the I/O-based Master Theorem. In Section 2.2, we describe and prove a version of the Master Theorem for the I/O model. We present a proof of this theorem to simplify our analysis and to help future papers avoid redoing this analysis.

Finally, in the full version of the paper we explain how some reductions in the RAM model imply faster algorithms in the I/O model.

2.1 Algorithms for Sparse 2 vs 3 Radius and Diameter

For both the radius and diameter problems on unweighted and undirected graphs, we can show distinguishing between a graph with a diameter or radius of 2 and one with a larger diameter or radius can be solved efficiently. Our algorithm relies on the reinterpretation of the 2 vs 3 problem as a set-disjointness problem. Every node, v, has an associated set, S_v, its adjacency list union itself. If two nodes have disjoint sets S_v and S_u, then they are distance greater than 2 from each other. Our algorithm for 2 vs 3 diameter and radius save an entire factor of M from the previously best known running times.

This is a similar idea to the reduction from 2 vs 3 diameter to OV and from 2 vs 3 radius to Hitting Set in the RAM model. These reductions were introduced by Abboud, Vassilevska-Williams and Wang [4]. While these reductions exist in the RAM model, they don't result in faster algorithms for 2 vs 3 diameter and radius in the I/O model because they use a hashing step that results in BFS being run from $\frac{|E|}{\Delta}$ nodes for some parameter Δ that can be set that gives the orthogonal vectors instance a dimension of Δ^2. In the I/O model, BFS is quite inefficient: we would need to set $\Delta \approxeq M$ to get an efficient algorithm using the approach in [4]. But, with a dimension of M^2 the OV algorithm will run very slowly. Therefore, below we present a solution to the set disjointness problem with no hashing into a smaller dimension.

Below we present the cache-aware algorithm for distinguishing 2 vs 3 diameter in an undirected, unweighted graph which runs in $O\left(\frac{|E|^2}{MB} + sort(|E|)\right)$ time where $sort(|E|)$ is the time to sort the elements in $|E|$ in the I/O model and $sort(|E|) = O\left(\frac{|E|\log|E|}{B}\right)$. We leave the proofs of cache-oblivious 2 vs 3 diameter and radius in the full version. For both 2 vs 3 diameter and radius we get running times of $O\left(\frac{n^2}{MB} + \frac{|E|\lg(|E|)}{B}\right)$. The algorithms and proofs for cache-obliviousness are finicky, but fundamentally are self-reductions of the form $T(n) = 4T(n/2) + n/B$. We leave the proofs to the full version of the paper.

We will start by giving a non-oblivious algorithm which relies on a recursive self-reduction. We will then show how to make this oblivious. It is easier to explain the analysis and algorithm when we can rely on the size of cache, but we can avoid that and get an oblivious algorithm anyway. The previous best algorithm is from Arge, Meyer and Toma which achieves $O(|V|sort(|E|)) = O\left(\frac{|V||E|\log|E|}{B}\right)$ [9]. We get an improvement over the previous algorithm in terms of running time whenever $\frac{|E|}{|V|} = o(M)$.

▶ **Theorem 6.** *Determining if the diameter of an undirected, unweighted graph is 1, 2 or greater than 2 can be done in* $O\left(\frac{|E|^2}{MB} + sort(|E|)\right)$ *time in the I/O-model.*

See full version of the paper for proof.

The proofs of cache-oblivious 2 vs 3 Diameter and 2 vs 3 Radius are included in the full version.

2.2 Master Theorem in the I/O Model

In this section, we formally define our Master Theorem framework for the I/O model and provide bounds on the I/O complexity of problems whose I/O complexity fits the specifications of our framework. In addition, we also describe some example uses of our Master Theorem for the I/O model.

The Master Theorem recurrence in the RAM model looks like $T(n) = aT(n/b) + f(n)$. We will use a similar recurrence but all functions will now be defined over n, M and B. The I/O-Master Theorem function $f(n, M, B)$ includes all costs that are incurred in each layer of the recursive call. This includes the I/O complexity of reading in an input, processing the input, processing the output and writing out the output. In this section, we assume that $f(n, M, B)$ is a monotonically increasing function in terms of n in order to apply our Master Theorem framework. What this means is that for any fixed M and B we want the number of I/Os to increase or stay the same as n increases. Given that $f(n, M, B)$ specifies the I/O complexity of reading in the inputs and writing out the outputs, we prove the following version of the Master Theorem in the I/O model.

▶ **Theorem 7** (I/O Master Theorem)**.** *If* $f(n, M, B)$ *contains the cost of reading in the input (for each subproblem) and writing output (after computation of each subproblem), then the following holds. Given a recurrence of* $T(n, M, B) = \alpha T(n/\beta, M, B) + f(n, M, B)$*, where* $\alpha \geq 1$ *and* $\beta > 1$ *are constants, and a base case of* $T(n/x, M, B) = t(x, M, B)$ *(where* $t(x, M, B) = \Omega(1)$*) for some* $x \leq n$ *and some function* $t(x, M, B)$*. Let* $A(n, M, B) = \left(\frac{n}{x}\right)^{\log_\beta(\alpha)} t(x, M, B)$*,* $C(n, M, B) = \left(\frac{n}{x}\right)^{\log_\beta(\alpha)+\varepsilon_1} t(x, M, B)$ *and* $D(n, M, B) = \left(\frac{n}{x}\right)^{\log_\beta(\alpha)-\varepsilon_2} t(x, M, B)$ *for some* $\varepsilon_1, \varepsilon_2 > 0$*, then we get the following cases:*

Case 1: *If* $f(n, M, B) = O\left(D(n, M, B)\right)$ *then* $T(n) = \Theta\left(A(n, M, B) + \frac{n}{B}\right)$.

Case 2: *If* $C(n, M, B) = O\left(f(n, M, B)\right)$ *and* $\alpha T(n/\beta, M, B) \leq cf(n, M, B)$ *for some constant* $c < 1$ *and all sufficiently large* n*, then* $T(n) = \Theta\left(f(n, M, B) + \frac{n}{B}\right)$.

Case 3: *If* $f(n, M, B) = \Theta(A(n, M, B))$*, then* $T(n, M, B) = \Theta(f(n, M, B) \log\left(\frac{n}{x}\right) + \frac{n}{B})$.

Case 4: *If* $f(n, M, B)$ *has a constant number of terms,* $f(n, M, B) = \Omega\left(\frac{n}{B}\right)$*, and none of the previous cases are satisfied, then* $T(n, M, B) = O\left(A(n, M, B) + f(n, M, B)\left(\frac{n}{x}\right)^{\log_\beta \alpha} + \frac{n}{B}\right)$ *(note that this includes if* A *and* f *are incomparable), with tighter upper bounds provided in our proof (in the full version of the paper) depending on characteristics of the actual function,* $f(n, M, B)$*.*

See full version of the paper for proof.

One-Layer Self-Reductions

We state a relationship between one-layer self-reductions and our Master Theorem framework above. We refer to the process of solving a problem by reducing to several problems of smaller size each of which can be solved in cache and one recursive call is necessary as a *one-layer self-reduction*. Suppose the runtime of an algorithm in the RAM model is $n^{\log_\beta \alpha}$, then by dividing the problems into $\frac{n^{\log_\beta \alpha}}{M}$ subproblems each of which takes M/B I/Os to

process, the I/O complexity of the algorithm is $\Theta\left(\frac{n^{\log_\beta \alpha}}{M^{\log_\beta \alpha - 1}B}\right)$ which is the same result we obtain via our Master Theorem framwork above when $t(x, M, B) = M/B$.

We now prove formally the theorem related to one-layer self-reductions.

▶ **Theorem 8.** *Let P be a problem of size n which can be reduced to $g(n/M)$ sub-problems, each of which takes $T(M, M, B)$ I/Os to process. The runtime of such a one-layer self reduction for the problem P is $T(n, M, B) = g(n/M)T(M, M, B) + f(n, M, B)$ where $f(n, M, B) = \Omega\left(\frac{n}{B}\right)$.*

See full version of the paper for proof.

2.3 Faster I/O Matrix Multiplication via I/O Master Theorem

As we mentioned above, any I/O algorithm that has a self-reduction to one of the forms stated in Section 2.2. Using our I/O Master Theorem, we can show a comparable I/O matrix multiplication bound to the matrix multiplication bound based on finding the rank of the Matrix Multiplication Tensor in the RAM model.

Recent improvements to matrix multiplication's running time also imply faster cache oblivious algorithms. Recent work has improved the bounds on ω where ω is the constant such that for any $0 < \varepsilon < 1$ there is an algorithm for n by n matrix multiplication that runs in $n^{\omega+\varepsilon}$. The most recent improvements on these bounds have been achieved by bounding the rank of the Matrix Multiplication Tensor [41, 28].

The I/O literature does not seem to have kept pace with these improvements. While previous work discusses the efficiency of naive matrix multiplication and Strassen matrix multiplication, it does not discuss the further improvements that have been generated.

We note in this section that the modern techniques to improve matrix multiplication running time, those of bounding the rank of the Matrix Multiplication Tensor, all imply cache-efficient algorithms.

▶ **Theorem 9** (Matrix Multiplication I/O Complexity[23]). *Let $T_{MA}(n) = O\left(\frac{n^2}{B}\right)$ be the time it takes to do matrix addition on matrices of size n by n. If the matrix multiplication tensor's rank is bounded such that the RAM model running time is $n^{\omega'+\varepsilon}$ for any $0 < \varepsilon < 1$ then the following self-reduction exists for some constant α,*

$$T_{MM}(n) = O\left(\alpha^{\omega'+\varepsilon} T_{MM}\left(\frac{n}{\alpha}\right) + \alpha^{\omega'} T_{MA}\left(\frac{n}{\alpha}\right)\right).$$

Self-reductions feed conveniently into cache oblivious algorithms. Notably, when we plug this equation into the I/O Master Theorem from Section 2.2, we obtain the following bound as given in Lemma 10. Recursive structures like this tend to result in cache-oblivious algorithms. After all, regardless of the size of cache, the problems will be broken down until they fit in cache. Then, when a problem and the algorithm's execution fit in memory, the time to answer the query is $O(M/B)$, regardless of the size of M and B.

▶ **Lemma 10.** *If the matrix multiplication tensor's rank is bounded such that the RAM model running time is $n^{\omega'+\varepsilon}$ for any $0 < \varepsilon < 1$ and Theorem 9 holds with base case $T_{MM}(\sqrt{M}) = \frac{M}{B}$, then the running time for cache-oblivious matrix multiplication in the I/O model is at most $O\left(\frac{n^{\omega'+\varepsilon}}{M^{\frac{\omega'+\varepsilon}{2}-1}B} + \frac{n^2}{B}\right)$ for any $0 < \varepsilon < 1$.*

See full version of the paper for proof.

3 Novel Reductions

In this section we cover reductions related to Wiener Index, Median, Single Source Shortest Paths, and s-t Shortest Paths. We first cover our super linear lower bounds, then cover the linear lower bounds.

3.1 Super Linear Lower Bounds

We present reductions in the I/O model which yield new lower bounds. We have as corollaries of these same reductions related lower bounds in the RAM model. Many of these reductions relate to the problem of finding the Wiener Index of the graph.

We show diameter reduces to Wiener Index, APSP reduces to Wiener Index, and we show 3 vs 4 radius reduces to median.

▶ **Definition 11** (Wiener Index)**.** Given a graph $G = (V, E)$ let $D[i, j]$ be the shortest path distance between node i and node j in the graph G, where $D[i, i] = 0$), $n = |V|$, and $m = |E|$. The Wiener index of the graph is $\sum_{\forall i \in V} \sum_{\forall j \in V} D[i, j]$.

▶ **Lemma 12.** *If (directed/undirected) Wiener index in a (weighted/unweighted) graph, $G = (V, E)$, is solvable in $T(n, m, M, B)$ time then for any choice of sets $X \subset V$ and $T \subset V$ the sum $\sum_{x \in X} \sum_{t \in T} \delta(x, t)$ is computable in $O\left(T(n, m, M, B) + \frac{m}{B}\right)$ time.*

See full version of the paper for proof.

▶ **Lemma 13.** *If undirected Wiener index in an unweighted graph, $G = (V, E)$, is solvable in $T(n, m, M, B)$ time then for any choice of sets $X \subset V$ and $T \subset V$ the sum $\sum_{x \in X} \sum_{t \in T} \max\{\min\{\delta(u, v), k+1\}, k\}$ is computable in $O\left(T(kn, km, M, B) + \frac{k|E|}{B}\right)$ time.*

See full version of the paper for proof.

▶ **Theorem 14.** *If undirected Wiener index in an unweighted graph is solvable in $T(n, m, M, B)$ time then counting the number of pairs $x \in X$ and $t \in T$ in a directed/undirected, unweighted graph where $\delta(x, t) = k$ and computing the number of pairs where $\delta(x, t) \geq k$ is computable in $O\left(T(kn, km, M, B) + \frac{km}{B}\right)$ time.*

See full version of the paper for proof.

We now show that Wiener index, in sparse graphs, can efficiently return small diameters. Notably, this means that improvements to the sparse Wiener index algorithm will imply faster algorithms for the sparse diameter problem than exist right now.

▶ **Corollary 15.** *If Wiener index is solvable in $T(n, m, M, B)$ time then returning $\min\{\text{diameter}, k\}$ is solvable in $O\left(\log(k) T(kn, km, kM, kB) + \frac{km}{B}\right)$ time.*

See full version of the paper for proof.

▶ **Corollary 16.** *If Wiener index is solvable in $T(n, m, M, B)$ time then returning the number of nodes at distances in $[1, k]$ from each other can be done in $O\left(kT(kn, kM, kB) + \frac{k^2 m}{B}\right)$ time.*

See full version of the paper for proof.

Next, we prove that improvements to median finding in sparse graphs improve the radius algorithm, using a novel reduction. Notably, in the I/O model 3 vs 4 radius is slower than 2 vs 3 radius; whereas, in the RAM-model, these two problems both run in n^2 time. The gap in the I/O model of a factor of M is what allows us to make these statements meaningful.

▶ **Theorem 17.** *If median is solvable in $T(n, m, M, B)$ time then 3 vs 4 radius is solvable in $O\left(T(n, m, M, B) + \frac{n^2}{MB} + \frac{E \log(E)}{B}\right)$ time.*

See full version of the paper for proof.

It has previously been shown that Wiener Index is equivalent to APSP in the RAM model. Here we show this also holds in the I/O-model.

▶ **Theorem 18.** *If Wiener Index is solvable in $n^{3-\varepsilon}$ time in a dense graph then APSP is solvable in $\tilde{O}(n^{3-\varepsilon} + n^2/B)$ time.*

See full version of the paper for proof.

3.2 Linear-Time Reductions

In fine-grained complexity, it often does not make sense to reduce linear-time problems to one another because problems often have a trivial lower bound of $\Omega(n)$ needed to read in the entire problem. However, in the I/O model, truly linear time—the time needed to read in the input—is $\Theta(n/B)$. Despite significant effort, many problems do not achieve this full factor of B in savings, and thus linear lower bounds of $\Omega(n)$ are actually interesting. We can use techniques from fine-grained complexity to try to understand some of this difficulty.

In the remainder of this section, we cover reductions between linear-time graph problems whose best known algorithms take longer than $O(|E|/B)$ time. This covers many of even the most basic problems, like the s-t shortest path problem, which asks for the distance between two specified nodes s and t in a graph G. The *sparse* s-t shortest paths problem has resisted improvement beyond $O(|V|/\sqrt{B})$ even in undirected unweighted graphs [30], and in directed graphs, the best known algorithms still run in time $\Omega(|V|)$ [17, 8].

Notably, the undirected unweighted s-t shortest path problem is solved by Single Source Shortest Paths (SSSP) and Breadth First Search (BFS). Further note that for directed graphs the best known algorithms for SSSP, BFS, and Depth First Search (DFS) in sparse, when $|E| = O(|V|)$, directed graphs take $O(|V|)$ time. Which is a cache miss for every constant number of operations, giving no speed up at all. SSSP, BFS, and DFS solve many other basic problems like graph connectivity.

By noting these reductions we want to show that improvements in one problem propagate to others. We also seek to explain why improvements are so difficult on these problems. Because, improving one of these problems would improve many others, any problem which requires new techniques to improve implies the others must also need these new techniques. Furthermore, any lower bound proved for one problem will imply lower bounds for the other problems reduced to it. We hope that improvements will be made to algorithms or lower bounds and propagated accordingly.

We show reductions between the following three problems in weighted and unweighted as well as directed and undirected graphs.

▶ **Definition 19** (s-t-shortest-path(G, s, t))**.** Given a graph G and pointers to two verticies s and t, return the length of the shortest path between s and t.

▶ **Definition 20** (Girth-Containing-Edge(G, e))**.** Given a graph G and a pointer to an edge e, return the length of the shortest cycle in G which contains e.

▶ **Definition 21** (Girth-Containing-Vertex(G, v))**.** Given a graph G and a pointer to a vertex v, return the length of the shortest cycle in G which contains v.

We now begin showing that efficient reductions exist between these hard to solve linear problems.

▶ **Theorem 22.** *Given an algorithm that solves (undirected/directed) s-t-shortest-path in $f(n, |E|, M, B)$ time (undirected/directed) Girth-Containing-Edge can be solved in $O(f(n, |E|, M, B) + O(1))$ time.*

See full version of the paper for proof.

▶ **Theorem 23.** *Given an algorithm that solves (undirected/directed) Girth-Containing-Edge in $f(n, |E|, M, B)$ time (undirected/directed) s-t-shortest-path can be solved in $O(f(n, |E|, M, B) + O(1))$ time.*

See full version of the paper for proof.

▶ **Theorem 24.** *Given an algorithm that solves (undirected/directed) Girth-Containing-Vertex in $f(n, |E|, M, B)$ time (undirected/directed) s-t-shortest-path can be solved in $O(f(n, |E|, M, B) + O(1))$ time.*

See full version of the paper for proof.

▶ **Theorem 25.** *Given an algorithm that solves (undirected/directed) Girth-Containing-Vertex in $f(n, |E|, M, B)$ time (undirected/directed) Girth-Containing-Edge can be solved in $O(f(n, |E|, M, B) + O(1))$ time.*

See full version of the paper for proof.

▶ **Theorem 26.** *Given an algorithm that solves directed Girth-Containing-Edge in $f(n, |E|, M, B)$ time then directed Girth-Containing-Vertex is solvable in $O(f(n, |E|, M, B) + n/B)$ time.*

See full version of the paper for proof.

▶ **Theorem 27.** *Given an algorithm that solves directed s-t-shortest-path in $f(n, |E|, M, B)$ time then directed Girth-Containing-Vertex is solvable in $O(f(n, |E|, M, B) + n/B)$ time.*

See full version of the paper for proof.

When solving Girth-Containing-Vertex in the directed case, we know which direction the path must follow the edges and can perform this decomposition. Unfortunately this no longer works in the undirected case and a more complex algorithm is needed, giving slightly weaker results.

▶ **Theorem 28.** *Given an algorithm that solves undirected s-t-shortest-path in $f(n, |E|, M, B)$ time then undirected Girth-Containing-Vertex is solvable in $O((f(n, |E|, M, B) + n/B) \lg(n))$ time.*

See full version of the paper for proof.

▶ **Theorem 29.** *Given an algorithm that solves undirected Girth-Containing-Edge in $f(n, |E|, M, B)$ time then undirected Girth-Containing-Vertex is solvable in $O((f(n, |E|, M, B) + n/B) \lg(n))$ time.*

See full version of the paper for proof.

4 Lower Bounds from Fine-Grained Reductions

The fundamental problems in the fine-grained complexity world are good starting points for assumptions in the I/O model because these problems are so well understood in the RAM model. Additionally, both APSP and 3-SUM have been studied in the I/O model [9, 34, 35]. These reductions allow us to propagate believed lower bounds from one problem to others, as well as propagate any potential future algorithmic improvements.

4.1 Reductions to 3-SUM

We will show that 3-SUM is reducible to both convolution 3-SUM and 0 triangle in the I/O-model.

▶ **Lemma 30.** *If convolution 3-SUM is solved in $f(n, M, B)$ time then 3-SUM is solved in $O(g^3 f(n/g, M, B)) + n^2/(gMB))$ time for all $g \in [1, n]$.*

See full version of the paper for proof.

▶ **Corollary 31.** *If convolution 3-SUM is solved in time $O(n^{2-\varepsilon}/(MB))$ or $O(n^2/(M^{1+\varepsilon}B))$ or $O(n^2/(MB^{1+\varepsilon}))$ then 3-SUM is solved in $O(n^{2-\varepsilon'}/(MB))$ or $O(n^2/(M^{1+\varepsilon'}B))$ or $O(n^2/(MB^{1+\varepsilon'}))$ time, violating the I/O 3-SUM conjecture.*

See full version of the paper for proof.

▶ **Lemma 32.** *If 0 triangle is solved in $f(n, M, B)$ time then convolution 3-SUM is solved in $O(\sqrt{n} f(\sqrt{n}, M, B) + n^{1.5} \lg_{M/B}(n)/B)$ time.*

See full version of the paper for proof.

▶ **Theorem 33.** *If 0 triangle is solved in time $O(n^{3-\varepsilon}/(MB))$ or $O(n^3/(M^{1+\varepsilon}B))$ or $O(n^3/(MB^{1+\varepsilon}))$ then 3-SUM is solved in $O(n^{2-\varepsilon'}/(MB))$ or $O(n^2/(M^{1+\varepsilon'}B))$ or $O(n^2/(MB^{1+\varepsilon'}))$ time, violating the I/O 3-SUM conjecture.*

See full version of the paper for proof.

▶ **Lemma 34.** *If 3-SUM is solved in $f(n, M, B)$ I/Os then Convolution 3-SUM is solvable in $O(f(n, M, B) + n/B)$ I/Os.*

See full version of the paper for proof.

▶ **Corollary 35.** *Convolution 3-SUM is solvable in $O(n^2/(MB) + n/B)$*

Proof. Given Lemma 34 ◀

4.2 APSP Reductions in the IO-Model

▶ **Definition 36** (All-Pairs-Shortest-Path(G))**.** Given a fully connected graph G with large edge weights (weights between $-n^c$ and n^c for some constant c) return the path lengeths between all pairs of nodes in a matrix D where $D[i][j]$ = the length of the shortest path from node i to node j).

Another related version of APSP requires us to return all the shortest paths in addition to the distances. To represent this information efficiently, one is required to return an n by n matrix P where the $P[i][j]$ is the next node after i on the shortest path from i to j. The matrix P allows one to extract the shortest path between two points by following the path through the matrix P. This problem is also called APSP.

▶ **Definition 37** (Three-Layer-APSP(G))**.** Solve APSP on G where G is promised to be a bipartite graph G which has partitions A, B, and C, such that there are no edges within A,B or C and no edges between A and C.

▶ **Definition 38** (Negative-triangle-detection(G))**.** Given a graph G, retuning true if there is a negative triangle and false if there is no negative triangle. This problem is also called $-\triangle$ detection.

▶ **Definition 39** ($(\min,+)$-Matrix-Multiplication(A,B))**.** This problem is a variant on matrix multiplication.Given an n by n matrix A and an n by n matrix B return an n by n matrix C such that $C[i][j] = \min(\{A[i][k] + B[k][j] | \forall k \in [1,n]\})$.

The motivation for showing I/O equivalences between these problems is two fold. First, just as in the RAM model, these reductions can provide a shared explanation for why some problems have seen no improvement in their I/O complexity for years.

The set of reductions

▶ **Theorem 40.** *If $(\min,+)$ runs in time $f(n,M,B)$, then APSP runs in $O(\lg(n) f(n,M,B))$.*

See full version of the paper for proof.

▶ **Theorem 41.** *If All Pairs Shortest Paths runs in time $f(n,M,B)$, then negative weight triangle detection in a tripartite graph runs in $O(f(n,M,B))$ cache misses.*

See full version of the paper for proof.

▶ **Theorem 42.** *If negative weight triangle detection in a tripartite graph runs in time $f(n,M,B)$, then three layer APSP with weights in the range $[-W,W]$ runs in $O\left(\lg(W) n^2 f\left(n^{1/3},M,B\right)\right)$ cache misses.*

See full version of the paper for proof.

▶ **Corollary 43.** *If negative weight triangle detection in a tripartite graph runs in time $f(n,M,B)$, then three layer APSP over weights in the range $[-poly(n), poly(n)]$ runs in $O\left(\lg(n) n^2 f\left(n^{1/3},M,B\right)\right)$ cache misses.*

See full version of the paper for proof.

▶ **Lemma 44.** *If All Pairs Shortest Paths runs in time $f(n,M,B)$, then three layer APSP runs in $O(f(n,M,B))$ cache misses.*

See full version of the paper for proof.

The equivalences

▶ **Theorem 45.** *The following problems run in time $\widetilde{O}(\frac{N^3}{\sqrt{M}B} + \frac{n^2}{B})$ cache misses:*

1. *All Pairs Shortest Paths*
2. $(\min,+)$ *matrix multiplication*
3. *Negative triangle detection in a tripartite graph*
4. *Three layer APSP*

See full version of the paper for proof.

▶ **Corollary 46.** *The following solve APSP faster.*

1. *If* $(\min,+)$ *matrix multiplication is solvable in* $f(n)$ *time then APSP is solvable in* $O(lg(n)f(n,M,B))$ *time.*
2. *If negative triangle detection in a tripartite graph is solvable in* $f(n)$ *time then APSP is solvable in* $O(lg^(n)n^2 f(n^{1/3},M,B))$ *time.*

See full version of the paper for proof.

▶ **Lemma 47.** *If 0 triangle is solved in* $f(n,M,B)$ *time then* $-\triangle$ *over numbers in the range of* $[-W,W]$ *is solved in* $O(lg(W)f(n,M,B)+lg(W)n^2/B)$.

See full version of the paper for proof.

▶ **Lemma 48.** *Zero triangle is solvable in* $O(n^3/(\sqrt{M}B)+n^2/B)$ *I/Os.*

See full version of the paper for proof.

4.3 Orthogonal Vectors (OV)

▶ **Lemma 49.** *OV is solvable in* $O(n^2/(MB)+n/B)$ *I/Os cache obliviously.*

See full version of the paper for proof.

▶ **Lemma 50.** *If sparse diameter is solvable in* $f(n,M,B)$ *I/Os then OV is solvable in* $\tilde{O}(f(n,M,B)+n/B)$ *I/Os.*

See full version of the paper for proof.

▶ **Lemma 51.** *If Edit Distance is solvable in* $f(n,M,B)$ *time then OV is solvable in* $\tilde{O}(f(n,M,B)+n/B)$ *I/Os.*

See full version of the paper for proof.

▶ **Lemma 52.** *If Longest Common Subsequence is solvable in* $f(n,M,B)$ *time then OV is solvable in* $\tilde{O}(f(n,M,B)+n/B)$ *I/Os.*

See full version of the paper for proof.

4.4 Hitting Set

Abboud, Vassilevska-Williams and Wang define a new problem, called Hitting Set [4].

▶ **Definition 53** (Hitting Set)**.** Given an input of a list A and B of d dimensional 1 and 0 vectors return $True$ if there $\exists a \in A$ such that $\forall b \in B\ a \cdot b > 0$.

▶ **Lemma 54.** *Hitting Set is solvable in* $\tilde{O}(n^2/(MB)+n/B)$ *I/Os.*

See full version of the paper for proof.

▶ **Lemma 55.** *If sparse radius is solvable in* $f(n,M,B)$ *I/Os then Hitting Set is solvable in* $\tilde{O}(f(n,M,B)+n/B)$ *I/Os.*

See full version of the paper for proof.

5 I/O Model Complexity Classes

In this section we examine the I/O model from a complexity theoretic perspective. Section 5.1 provides some necessary background information. In Section 5.2 we define the classes $PCACHE_{M,B}$ and $CACHE_{M,B}(t(n))$ describing the problems solvable in a polynomial number of cache misses and $O(t(n))$ cache misses respectively. We then demonstrate that $PCACHE_{M,B}$ lies between P and $PSPACE$ for reasonable choices of cache size. In the full version we provide simulations between the I/O model and both the RAM and Turing machine models. In Section 5.3 we prove the existence of a time hierarchy in $CACHE_{M,B}(t(n))$. The existence of a time hierarchy in the I/O model grounds the study of fine-grained complexity by showing that such increases in running time do provably allow more problems to be solved. The techniques to achieve many of the results in this section also follow in the same theme of reductions, although the focus of the problems examined is quite different.

5.1 Hierarchy Preliminaries: Oracle Model

Oracles are used to prove several results, most notably in the time-hierarchy proof. The oracle model was introduced by Turing in 1938 [40]. The definition we use here comes from Soare [39] and similar definitions can be found in computational complexity textbooks.

In the Turing machine oracle model of computation we add a second tape and corresponding tape head. This oracle tape and its oracle tape head can do everything the original tape can: reading, writing and moving left and right. This oracle tape head has two additional states ASK and $RESPONSE$. After writing to the oracle tape, the tape head can go into the ASK state. In the ASK state the oracle computation is done on the input written to the oracle tape and then the tape head is changed to the $RESPONSE$ state. All of this is done in one computational step. If the oracle is the function language $L: \{0,1\}^n \to \{0,1\}^*$, then the output is written on the tape for the input i is $L(i)$. This allows the Turing machine to make $O(1)$ cost black box calls to the oracle language and get strings as output.

The notation A^B describes a computational class of the languages decidable by an oracle Turing machine of A with a B oracle. The oracle language will be a language decidable in the function version of B. The oracle machine will then be resource limited as A is resource limited.

In this paper we will also talk about RAM machine oracles. This is a simple extension of the typical Turing machine oracle setup. The RAM machine will have two randomly accessible memories. One will be the standard RAM memory. The other memory will be the oracle memory, the RAM can read and write words to this memory and can additionally enter the ASK state. One time step after entering the ASK state the RAM will be returned to the $RESPONSE$ state and the contents of the oracle memory will contain the oracle language output, $L(i)$.

5.2 $PCACHE_{M,B}$ and its relationship with P and PSPACE

First we define the class of problems solvable given some function, $t(n)$, the number of cache misses, up to constant factors.

▶ **Definition 56.** Let $CACHE_{M,B}(t(n))$ be the set of problems solvable in $O(t(n))$ cache misses on a IO-model machine with a cache of size $O(M)$ and a cache line size of size $O(B)$.

Next, we consider the class of problems solvable by any polynomial number of cache misses.

▶ **Definition 57.** Let $PCACHE_{M,B}$ be the set of problems solvable in polynomial numbers of cache misses on a IO-model machine with a cache of size $O(M)$ and a cache lines of size $O(B)$.

$$PCACHE_{M,B} = \bigcup_{i=1}^{\infty} CACHE_{M,B}(n^i)$$

First, let us note that the CACHE class can simulate the RAM class. The IO-model is basically a RAM model with extra power.

▶ **Lemma 58.** $RAM(t(n)) \subseteq CACHE_{M,B}(t(n))$

See full version of the paper for proof.

Now we introduce a complexity class *MEM*. Note that this class is very similar to $SPACE$.

▶ **Definition 59.** We define the class $MEM(s(n))$ to be the set of problems solvable in $SPACE(s(n))$ when the input is of size $O(s(n))$.

Why *MEM* and not $SPACE$? We want to use the *MEM* class as an oracle which will model computation doable on a cache machine in one cache miss. When $t(n) = \Omega(n)$ then $MEM(t(n)) = SPACE(t(n))$; however, these classes differ when we have a small work space. A $SPACE(o(n))$ machine is given a read-only tape of size n and compute space $o(n)$. This extra read-only tape gives the $SPACE$ machine too much power when compared with the cache. Notably, we can scan through the entire input with one step with a $SPACE(\lg(n))$ oracle. A cache would require n/B time to scan this input.

▶ **Lemma 60.** $MEM(Mw) \subseteq CACHE_{M,B}(M/B)$.

See full version of the paper for proof.

Now we prove that a RAM machine with oracle access to our *MEM* oracle can be simulated by a cache machine. We simulate the MEM machine and RAM machine together efficiently in cache.

▶ **Corollary 61.** $RAM(t(n))^{MEM(Mw)} \subseteq CACHE_{M+3,B}(t(n))$.

See full version of the paper for proof.

The class $PCACHE_{M,B}$ is equivalent to a polynomial time algorithm with oracle access to a MEM oracle. Intuitively, in both cases we get to use a similarly powerful object (the cache or the MEM oracle) a polynomial number of times.

▶ **Theorem 62.** *If* $Bw = O(poly(n))$, *then* $PCACHE_{M,B} = P^{MEM(Mw)}$

See full version of the paper for proof.

We then note that in many cases MEM and SPACE are equivalent.

▶ **Corollary 63.** *If* $Bw = O(poly(n))$ *and* $Mw = \Omega(n)$, *then* $PCACHE_{M,B} = P^{SPACE(Mw)}$.

See full version of the paper for proof.

Finally, we note that P is a subset of $PCACHE_{M,B}$.

▶ **Lemma 64.** *If* $Mw = \Theta(poly(n))$, *then* $PCACHE_{M,B} \subseteq PSPACE$

See full version of the paper for proof.

▶ **Lemma 65.** $\bigcup_{c=1}^{\infty} PCACHE_{n^c,B} = PSPACE$

See full version of the paper for proof.

5.3 $CACHE_{M,B}$ Hierarchy

In this section we prove that a hierarchy exists in the IO-model. The separation in the CACHE hierarchy is B times the separation for the RAM hierarchy. We know that RAM machines given polynomially more time can solve more problems than those given polynomially less.

▶ **Theorem 66.** *For $\varepsilon \geq 0$,*

$$RAM^O(t(n)) \subsetneq RAM^O((t(n))^{1+\varepsilon}).$$ *[22]*

Let $s(n)$ be the space usage of the algorithm running on the RAM machine. Let $\alpha = \frac{B+\lceil \lg(s(n))/w \rceil}{B}$, which is the number of cache lines needed to represent both a cache line and its memory address. Note, in the case where one word is large enough to address all of the memory used by the algorithm (a standard assumption) $\alpha = 1 + 1/B \leq 2$. We now give a simulation of a CACHE machine by a RAM machine with MEM oracle.

▶ **Lemma 67.** $CACHE_{M,B}(t(n)) \subseteq RAM\left(t(n)\left(B+\alpha\right)\right)^{MEM(Mw\alpha)}$

See full version of the paper for proof.

Plugging our simulation into the RAM hierarchy gives a separation result for the CACHE complexity classes.

▶ **Theorem 68.** *For all $\varepsilon > 0$*

$$CACHE_{M,B}(t(n)) \subsetneq CACHE_{M\alpha,B}((\alpha t(n) B)^{1+\varepsilon}).$$

See full version of the paper for proof.

Under reasonable assumptions about the values of input and word sizes, we can construct a cleaner version of the above theorem.

▶ **Corollary 69.** *When $s(n) = 2^{O(wB)}$, in other words the memory used by the algorithm is referenceable by $O(B)$ words,*

$$CACHE_{M,B}(t(n)) \subsetneq CACHE_{M,B}((t(n) B)^{1+\varepsilon}).$$

Proof. Note $\alpha = 1 + 1/B = O(1)$, and thus is a constant with respect to the time and size of memory which are defined asymptotically. Thus this factor disappears. ◀

Acknowledgements. We thank the anonymous reviewers for their helpful suggestions.

References

1 Amir Abboud, Arturs Backurs, and Virginia Vassilevska Williams. If the current clique algorithms are optimal, so is Valiant's parser. In *Proceedings of the IEEE 56th Annual Symposium on Foundations of Computer Science*, FOCS 2015, pages 98–117, Berkeley, CA, October 2015.

2 Amir Abboud, Fabrizio Grandoni, and Virginia Vassilevska Williams. Subcubic equivalences between graph centrality problems, APSP and diameter. In *Proceedings of the 26th Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA 2015, pages 1681–1697, San Diego, CA, January 2015.

3 Amir Abboud and Virginia Vassilevska Williams. Popular conjectures imply strong lower bounds for dynamic problems. In *Foundations of Computer Science (FOCS), 2014 IEEE 55th Annual Symposium on*, pages 434–443, 2014.

4 Amir Abboud, Virginia Vassilevska Williams, and Joshua Wang. Approximation and fixed parameter subquadratic algorithms for radius and diameter in sparse graphs. In *Proceedings of the 27th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 377–391, 2016.

5 Alok Aggarwal and S. Vitter, Jeffrey. The input/output complexity of sorting and related problems. *Communications of the ACM*, 31(9):1116–1127, 1988.

6 Lars Arge. External-memory algorithms with applications in gis. In *Algorithmic Foundations of Geographic Information Systems, This Book Originated from the CISM Advanced School on the Algorithmic Foundations of Geographic Information Systems*, pages 213–254, 1997. URL: `http://dl.acm.org/citation.cfm?id=648260.753061`.

7 Lars Arge. The buffer tree: A technique for designing batched external data structures. *Algorithmica*, 37(1):1–24, 2003. `doi:10.1007/s00453-003-1021-x`.

8 Lars Arge, Michael A. Bender, Erik D. Demaine, Bryan Holland-Minkley, and J. Ian Munro. An optimal cache-oblivious priority queue and its application to graph algorithms. *SIAM J. Comput.*, 36(6):1672–1695, 2007. `doi:10.1137/S0097539703428324`.

9 Lars Arge, Ulrich Meyer, and Laura Toma. External memory algorithms for diameter and all-pairs shortest-paths on sparse graphs. In Josep Díaz, Juhani Karhumäki, Arto Lepistö, and Donald Sannella, editors, *Automata, Languages and Programming: 31st International Colloquium, ICALP 2004, Turku, Finland, July 12-16, 2004. Proceedings*, volume 3142 of *Lecture Notes in Computer Science*, pages 146–157. Springer, 2004. `doi:10.1007/978-3-540-27836-8_15`.

10 Arturs Backurs and Piotr Indyk. Edit distance cannot be computed in strongly subquadratic time (unless seth is false). In *Proceedings of the 47th Annual ACM on Symposium on Theory of Computing*, pages 51–58, 2015.

11 Boaz Barak. A probabilistic-time hierarchy theorem for "slightly non-uniform" algorithms. In *Proceedings of the 6th International Workshop on Randomization and Approximation Techniques*, RANDOM 2002, pages 194–208, Cambridge, MA, September 2002.

12 Ilya Baran, Erik D Demaine, and Mihai Pătraşcu. Subquadratic algorithms for 3sum. In *Workshop on Algorithms and Data Structures*, pages 409–421, 2005.

13 R. Bayer and E. McCreight. Organization and maintenance of large ordered indices. In *Proceedings of the 1970 ACM SIGFIDET (Now SIGMOD) Workshop on Data Description, Access and Control*, SIGFIDET '70, pages 107–141, 1970. `doi:10.1145/1734663.1734671`.

14 Michael A. Bender, Richard Cole, Erik D. Demaine, Martin Farach-Colton, and Jack Zito. Two simplified algorithms for maintaining order in a list. In *Proceedings of the 10th Annual European Symposium on Algorithms*, ESA 2002, pages 152–164, 2002. URL: `http://dl.acm.org/citation.cfm?id=647912.740822`.

15 Karl Bringmann. Why walking the dog takes time: Frechet distance has no strongly subquadratic algorithms unless SETH fails. In *Proceedings of the 55th IEEE Annual Symposium on Foundations of Computer Science*, FOCS 2014, pages 661–670, Philadelphia, PA, October 2014.

16 Gerth Stølting Brodal. Cache-oblivious algorithms and data structures. In Torben Hagerup and Jyrki Katajainen, editors, *Algorithm Theory - SWAT 2004, 9th Scandinavian Workshop on Algorithm Theory, Humlebaek, Denmark, July 8-10, 2004, Proceedings*, volume 3111 of *Lecture Notes in Computer Science*, pages 3–13. Springer, 2004. `doi:10.1007/978-3-540-27810-8_2`.

17 Yi-Jen Chiang, Michael T Goodrich, Edward F Grove, Roberto Tamassia, Darren Erik Vengroff, and Jeffrey Scott Vitter. External-memory graph algorithms. In *SODA*, volume 95, pages 139–149, 1995.

18 Rezaul Alam Chowdhury and Vijaya Ramachandran. Cache-oblivious shortest paths in graphs using buffer heap. In Phillip B. Gibbons and Micah Adler, editors, *SPAA 2004: Proceedings of the Sixteenth Annual ACM Symposium on Parallelism in Algorithms and*

Architectures, June 27-30, 2004, Barcelona, Spain, pages 245–254. ACM, 2004. doi:10.1145/1007912.1007949.

19 Rezaul Alam Chowdhury and Vijaya Ramachandran. External-memory exact and approximate all-pairs shortest-paths in undirected graphs. In *Proceedings of the 16th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 735–744, 2005.

20 Rezaul Alam Chowdhury and Vijaya Ramachandran. Cache-oblivious dynamic programming. In *Proceedings of the 17th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 591–600, 2006.

21 Rezaul Alam Chowdhury and Vijaya Ramachandran. Cache-oblivious dynamic programming. In *Proceedings of the 17th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 591–600, 2006.

22 Stephen A. Cook and Robert A. Reckhow. Time-bounded random access machines. In *Proceedings of the 4th Annual ACM Symposium on Theory of Computing*, pages 73–80, 1972.

23 Don Coppersmith and Shmuel Winograd. Matrix multiplication via arithmetic progressions. *J. Symb. Comput.*, 9(3):251–280, 1990. doi:10.1016/S0747-7171(08)80013-2.

24 Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms, Third Edition*. The MIT Press, 3rd edition, 2009.

25 Erik D. Demaine. Cache-oblivious algorithms and data structures. Lecture Notes from the EEF Summer School on Massive Data Sets, 2002.

26 Roman Dementiev. *Algorithm engineering for large data sets.* PhD thesis, Saarland University, 2006.

27 Matteo Frigo, Charles E. Leiserson, Harald Prokop, and Sridhar Ramachandran. Cache-oblivious algorithms. In *Proceedings of the 40th Annual Symposium on the Foundations of Computer Science (FOCS)*, pages 285–298, 1999.

28 François Le Gall. Powers of tensors and fast matrix multiplication. In *International Symposium on Symbolic and Algebraic Computation, ISSAC 2014, Kobe, Japan, July 23-25, 2014*, pages 296–303, 2014.

29 Jia-Wei Hong and H. T. Kung. I/O complexity: The red-blue pebble game. In *Proceedings of the 13th Annual ACM Symposium on the Theory of Computation (STOC)*, pages 326–333, 1981.

30 Kurt Mehlhorn and Ulrich Meyer. External-memory breadth-first search with sublinear I/O. In Rolf H. Möhring and Rajeev Raman, editors, *Algorithms - ESA 2002, 10th Annual European Symposium, Rome, Italy, September 17-21, 2002, Proceedings*, volume 2461 of *Lecture Notes in Computer Science*, pages 723–735. Springer, 2002. doi:10.1007/3-540-45749-6_63.

31 Bruno Menegola. An external memory algorithm for listing triangles. Bachelor's thesis, Universidade Federal do Rio Grande do Sul, 2010. URL: http://hdl.handle.net/10183/26335.

32 Ulrich Meyer and Norbert Zeh. I/o-efficient undirected shortest paths. In Giuseppe Di Battista and Uri Zwick, editors, *Algorithms - ESA 2003, 11th Annual European Symposium, Budapest, Hungary, September 16-19, 2003, Proceedings*, volume 2832 of *Lecture Notes in Computer Science*, pages 434–445. Springer, 2003. doi:10.1007/978-3-540-39658-1_40.

33 Rasmus Pagh and Francesco Silvestri. The input/output complexity of triangle enumeration. In *Proceedings of the 33rd ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, pages 224–233, 2014.

34 Rasmus Pagh and Morten Stöckel. The input/output complexity of sparse matrix multiplication. In Andreas S. Schulz and Dorothea Wagner, editors, *Algorithms - ESA 2014*

- *22th Annual European Symposium, Wroclaw, Poland, September 8-10, 2014. Proceedings*, volume 8737 of *Lecture Notes in Computer Science*, pages 750–761. Springer, 2014. doi:10.1007/978-3-662-44777-2_62.

35 Mihai Pătraşcu. Towards polynomial lower bounds for dynamic problems. In *Proceedings of the 42nd ACM Symposium on Theory of Computing*, pages 603–610, 2010.

36 Liam Roditty and Virginia Vassilevska Williams. Fast approximation algorithms for the diameter and radius of sparse graphs. In *Proceedings of the 45th Symposium on Theory of Computing Conference*, STOC 2013, pages 515–524, Palo Alto, CA, June 2013.

37 Raimund Seidel. On the all-pairs-shortest-path problem. In S. Rao Kosaraju, Mike Fellows, Avi Wigderson, and John A. Ellis, editors, *Proceedings of the 24th Annual ACM Symposium on Theory of Computing, May 4-6, 1992, Victoria, British Columbia, Canada*, pages 745–749. ACM, 1992. doi:10.1145/129712.129784.

38 Michael Sipser. *Introduction to the Theory of Computation*, volume 2. Thomson Course Technology Boston, 2006.

39 Robert I. Soare. *Recursively enumerable sets and degrees: A study of computable functions and computably generated sets*. Springer Science & Business Media, 1999.

40 A. M. Turing. Systems of logic based on ordinals. *Proceedings of the London Mathematical Society*, s2-45(1):161–228, 1939.

41 Virginia Vassilevska Williams. Multiplying matrices faster than coppersmith-winograd. In *STOC*, pages 887–898, 2012.

42 Jeffrey Scott Vitter. External memory algorithms and data structures. *ACM Comput. Surv.*, 33(2):209–271, 2001. doi:10.1145/384192.384193.

43 Ryan Williams. Hierarchy for BPP vs derandomization. Theoretical Computer Science Stack Exchange. URL: http://cstheory.stackexchange.com/q/6769.

44 Virginia Vassilevska Williams and Ryan Williams. Subcubic equivalences between path, matrix and triangle problems. In *Foundations of Computer Science (FOCS), 2010 51st Annual IEEE Symposium on*, pages 645–654, 2010.

Fast and Deterministic Constant Factor Approximation Algorithms for LCS Imply New Circuit Lower Bounds

Amir Abboud[1] **and Aviad Rubinstein**[*2]

1 **IBM Almaden Research Center, San Jose, CA, USA**
amir.abboud@ibm.com

2 **Department of Computer Science, Harvard University, Cambridge, MA, USA**
aviad@seas.harvard.edu

Abstract

The Longest Common Subsequence (LCS) is one of the most basic similarity measures and it captures important applications in bioinformatics and text analysis. Following the SETH-based nearly-quadratic time lower bounds for LCS from recent years [4, 22, 5, 3], it is a major open problem to understand the complexity of approximate LCS. In the last ITCS [2] drew an interesting connection between this problem and the area of circuit complexity: they proved that approximation algorithms for LCS in deterministic truly-subquadratic time imply new circuit lower bounds (E^{NP} does not have non-uniform linear-size Valiant Series Parallel circuits).

In this work, we strengthen this connection between approximate LCS and circuit complexity by applying the *Distributed PCP* framework of [6]. We obtain a reduction that holds against much larger approximation factors (super-constant, as opposed to $1 + o(1)$ in [2]), yields a lower bound for a larger class of circuits (linear-size NC^1), and is also easier to analyze.

1998 ACM Subject Classification F.2 Analysis of Algorithms and Problem Complexity

Keywords and phrases Distributed PCP, Longest Common Subsequence, Fine-Grained Complexity, Circuit Lower Bounds, Strong Exponential Time Hypothesis

Digital Object Identifier 10.4230/LIPIcs.ITCS.2018.35

1 Introduction

The Longest Common Substring (LCS) is a fundamental similarity measure between two strings with many important applications to data comparison. It is an elegant abstraction of the core task in sequence alignment tasks in bioinformatics. Given two strings of length N, there is a classical dynamic programming algorithm that computes the LCS in time $O(N^2)$. The quadratic time requirement is prohibitive for very long strings (e.g. genomic sequences), and obtaining a substantially faster algorithm is a longstanding and central open question. In practice, biologists use heuristics such as BLAST to solve it in near-linear time but without any guarantees on the optimality of the solution [8]. Interesting results from recent years [10, 4, 22, 5] showed that under certain complexity assumptions such as "SETH"[1], there are no truly subquadratic algorithms for this problem.

* Research supported by a Rabin Postdoctoral Fellowship.

[1] The Strong Exponential Time Hypothesis (SETH) [35] postulates that we cannot solve k-SAT in $O((2-\varepsilon)^n)$ time, for some $\varepsilon > 0$ independent of and for all constant k.

For many applications, we would be happy to settle for an approximate solution if it could be found in truly subquadratic (ideally near-linear) time. In contrast to the exact variant, the complexity of approximating the LCS is poorly understood. On the algorithmic side, for strings over alphabet Σ there is a trivial $1/|\Sigma|$-approximation algorithm (using only the symbol that appears the most in both strings). On the complexity sides, the known reductions from SETH tell us very little about hardness of approximation. There are two main obstacles to obtaining SETH-based hardness of approximation:

The PCP blowup. The PCP Theorem, which is a crucial step in most NP-hardness of approximation results, can be seen as reducing the satisfiability of a CNF φ to approximate satisfiability of a new CNF φ'. If φ has n variables, the most efficient PCP constructions construct φ' with $n' = n \cdot \mathsf{polylog}(n)$ variables [28]. Obtaining a linear dependence is a major open problem (e.g. [17, 29, 41, 40, 24]).

In contrast, the known reductions from SETH to LCS begin with a CNF over n variables, and transform it to a hard instance of LCS of string length $N \approx 2^{n/2}$. Now solving LCS faster than $N^2 \approx 2^n$ time implies new algorithms for SAT. Even if we had a fantastic PCP of blowup $n' = 10n$, if we begin the reduction with the hard-to-approximate CNF φ', we would get strings of length $N' \approx 2^{n'/2} \approx 2^{5n}$.

The PCP blowup obstacle is common to almost all known reductions from SETH. For several other problems, this obstacle was addressed by the *distributed PCP*[2] framework in [6]. However, this technique does not yet suffice for hardness of approximation of LCS, in part due to the "contribution of unsatisfying assignments" obstacle described next.

Contribution of unsatisfying assignments. The second obstacle is more specific to LCS (and a few other string similarity measures like Edit Distance and Dynamic Time Warping Distance). The reduction from SETH proceeds by concatenating approximately $N \approx 2^{n/2}$ gadgets, one for each partial assignment to half of the variables. On a yes instance, matching the two gadgets that correspond to the satisfying assignment gives a higher contribution to the LCS than matching any two non-satisfying partial assignments. How much more can the satisfying pair contribute to the LCS compared to a non-satisfying pair? The largest gap we can hope to construct is a $|\Sigma|$-factor, because of the trivial $1/|\Sigma|$-approximation algorithm. If we want to keep the alphabet size small ($|\Sigma| = N^{o(1)}$), this is still negligible compared to the contribution from approximately N (disjoint) pairs of non-satisfying partial assignments.

In short, an inherent limitation of all known reduction techniques is that the multiplicative approximation of the LCS in the resulting instances can be obtained from an additive approximation of the fraction of satisfying assignments. The latter can be computed easily by sampling a small number of uniformly random assignments.

In the last ITCS, [2] drew an interesting connection between this problem and classical questions on circuit lower bounds. The authors observed that designing a *deterministic* algorithm that can approximate the number of satisfying assignment to formulas (that are slightly more complex than CNFs) is a challenging task with connections to pseudorandomness, and so this barrier can be circumvented, in a certain sense, if we restrict the attention to deterministic approximation algorithms. In particular, they show that if LCS admits a *deterministic* truly-subquadratic approximation algorithm, then certain long sought-after circuit lower bounds would be implied. However, to obtain such consequences, one would need to design a very good $(1 + o(1))$ approximation algorithms for LCS.

[2] The term *distributed PCP* was first used by Drucker [30], but in a completely different context.

Given the above two, it is natural to ask whether we can combine the techniques of [2] and [6] to obtain stronger inapproximability for LCS in deterministic truly subquadratic time. In this paper we show that this is indeed the case, and in a strong sense: using the distributed PCP framework from [6], we can replace the not-even-1.001 hardness of approximation factor from [2] with superconstant factors.

Additionally, using distributed PCPs also allows us to obtain stronger circuit lower bounds than [2]. The circuit complexity consequence of deterministic and fast approximate LCS algorithms established by [2] was that E^{NP} does not have non-uniform linear-size Valiant Series Parallel (VSP) circuits. In the 1970's Valiant defined the VSP property and argued that it is common in algorithmic circuits. This consequence is still out of reach of current techniques and is typically reported as a circuit lower bound from refuting SETH [37]. Here, we show that the VSP restriction can be replaced by the more natural restriction that the circuits have logarithmic depth (NC^1). Intuitively, the latter are more expressive. Proving the following statement would be a major breakthrough in complexity theory.

▶ **Consequence 1.1.** *The class E^{NP} does not have non-uniform circuits of size $O(n)$ and depth $O(\log n)$, nor VSP circuits of size $O(n)$.*

We are now ready to present our main theorem:

▶ **Theorem 1.2** (Main Theorem)**.** *If there is an algorithm that, given two length-N strings x, y over alphabet Σ, where $|\Sigma| = N^{o(1)}$, approximates the LCS of x and y to within any constant factor in deterministic, $O(N^{2-\varepsilon})$ time, then Consequence 1.1 follows.*

1.1 A succinct discussion of techniques

A sequence of previous works [34, 53, 50, 18, 2] reduces the task of proving Consequence 1.1 to designing deterministic algorithms for the following problem: given an OR with fan-in $2^{o(n)}$ over k-CNFs, for $k = O(n^{0.1})$, approximate the fraction of satisfying assignments (Lemma 2.4) in $\mathsf{DTIME}(2^n/n^{\omega(1)})$. The outer OR is easy to implement, and so we focus on any given CNF.

For CNFs with constant clause width, a common first step is to use the Sparsification Lemma of [36] which reduces the number of clauses to $m = O(n)$. However, the O-notation in this lemma hides a blowup which is doubly exponential in the width, so in our case ($k \approx n^{0.1}$) we are better off sticking with the trivial bound on the number of clauses: $m \lessapprox \binom{n}{n^{0.1}} \approx 2^{n^{0.1}}$.

As we mentioned earlier, the reduction constructs a gadget for each possible assignment to the first (resp. last) half of the variables. We want the LCS of the two gadgets to implement a verifier that receives the two assignments and verifies that they indeed satisfy the CNF. A key observation in [6] is that this task reduces to solving a Set Disjointness problem over the universe of clauses $[m]$: Given partial assignment $\alpha \in \{0,1\}^{n/2}$ to the first half of the variables, Alice locally constructs the set $S_\alpha \subseteq [m]$ of clauses that are not satisfied by α (but she still hopes those clauses are satisfied by the assignment to the remaining variables). Similarly, Bob locally constructs a set $T_\beta \subseteq [m]$ of clauses that he cannot guarantee are satisfied by his partial assignment, β. Now the joint assignment (α, β) satisfies the CNF if and only if S_α, T_β are disjoint.

Observe that two sets are disjoint iff their representation as binary vectors (in $\{0,1\}^m$) are orthogonal (over the reals). Indeed, so far our reduction looks like the classical reduction to the Orthogonal Vectors problem [52]: given two sets $A, B \in \{0,1\}^m$, is there a pair $a \in A, b \in B$ that is orthogonal?

Set Disjointness is a rather difficult problem in Communication Complexity: the randomized and even non-deterministic complexities of Set Disjointness are linear. Fortunatley,

there is an $\tilde{O}(\sqrt{m})$ MA-communication protocol due to [1]. In [6] m was linear, so we could enumerate over all protocols in subexponential time. Here, since $m \approx 2^{n^{0.1}}$, the quadratic saving of the MA-protocol does not help. Instead, we use an IP-protocol for Set Disjointness (also due to [1]) which uses only $\tilde{O}(\log m) \approx n^{0.1}$ communication.

We abstract the first part of the reduction (up to and including the IP-protocol) via a new problem a-la Orthogonal Vectors, that we call Tropical Tensors. Given two lists of tensors $A, B \in \{0,1\}^{[d_1]\times[d_2]\times\cdots\times[d_t]}$, we want to find a pair $a \in A, b \in B$ that maximizes a similarity measure $s(a,b)$ which is defined via a chain of alternating $+$ and max operations, where at the base we take the product of a_i and b_i; we call this the *Tropical Similarity* [3] of a and b.

Similar to Orthogonal Vectors, our new problem allows us to abstract out the PCP-like construction on one hand, and the LCS-specific gadgets on the other hand. While its definition is somewhat more involved than the original Orthogonal Vectors, the extra expressive power allows us to prove a stronger hardness of approximation result: Consequence 1.1 is implied by any truly subquadratic deterministic algorithm that can distinguish between the case where almost all pairs have almost maximum Tropical Similarity, and the case where the Tropical Similarity of every pair is tiny (see Theorem 3.2 for details). We hope that the Tropical Tensors problem will find further applications; see Remark 3.4 for some suggestions.

Once we establish the hardness of Tropical Tensors, we reduce it (Section 4) to LCS using gadgets that implement $+$ and max operations. Our reduction to LCS is particularly simple because the gap we obtain from Tropical Tensors is so large, that we do not need to pad our gadgets to enforce well-behaved solutions.

1.2 Related work

Algorithms for LCS and related problems

Even though many ideas and heuristics for LCS were designed [25, 19, 27, 26] (see also [43, 20] for surveys), none has proven sufficient to compute a better than $|\Sigma|$ approximation in strongly subquadratic time.

Many ingenious approximation algorithms were discovered for the related Edit Distance problem. A linear time $\sqrt{n}$-approximation follows from the exact algorithm that computes the Edit Distance in time $O(n+d^2)$ where $d = ED(S,T)$ [39]. Subsequently, this approximation factor has been improved to $n^{3/7}$ by Bar-Yossef et al. [12], then to $n^{1/3+o(1)}$ by Batu et al. [13]. Building on the breakthrough embedding of Edit Distance by Ostrovsky and Rabani [44], Andoni and Onak obtained the first near-linear time algorithm with a *subpolynomial* approximation factor of $2^{\tilde{O}(\sqrt{\log n})}$. Most recently, Andoni, Krauthgamer, and Onak [9] significantly improved the approximation to polylogarithmic obtaining an algorithm that runs in time $n^{1+\varepsilon}$ and gives $(\log n)^{O(1/\varepsilon)}$ approximation for every fixed $\varepsilon > 0$. There are many works on approximate Edit Distance in various computational models, see e.g. [43, 9, 23] and the references therein. It remains a huge open question whether Edit Distance can be approximated to within a constant factor in near-linear time.

A general tool for speeding up dynamic programming algorithms through a controlled relaxation of the optimality constraint is highly desirable. Encouraging positive results along these lines were recently obtained by Saha [47, 48, 49] for problems related to parsing

[3] The name is inspired by Tropical Algebras, which support $+$ and min operations.

context-free languages. However, we are still far from understanding, more generally, when and how such algorithms are possible.

Fine-grained complexity of LCS

Many hardness results have been recently shown for LCS. Shortly after the $N^{2-o(1)}$ SETH-based lower bound of Backurs and Indyk [10] for the related problem of Edit Distance, it was proven that LCS has a similar lower bound [4, 22]. Bringmann and Kunnemann [22] proved that the SETH lower bound holds even when the strings are binary, and [4] prove that LCS on k strings has an $N^{k-o(1)}$ lower bound. Very recently, [3] prove that the time complexity of computing the LCS between strings of length N that are compressed down to size n (using any of the standard grammar compressions such as Lempel-Ziv) is lower bounded by $(Nn)^{1-o(1)}$ under SETH, and a matching upper bound is known [31].

[5] proved quadratic lower bounds for LCS under safer versions of SETH where CNF is replaced with NC circuits, and connected LCS to circuit lower bounds for the first time. [5] also showed that even mildly subquadratic algorithms for LCS, e.g. $O(N^2/\log^{50} N)$ would imply breakthrough circuit lower bounds similar to Consequence 1.1. This connection to circuit lower bounds was exploited in the work of [2] who showed that such consequences can follow even from approximation algorithms.

The only SETH-based hardness of approximation results for LCS are for variants of the classical problem. For instance, approximate "closest pair" under the LCS similarity requires nearly quadratic time even for $2^{(\log N)^{1-o(1)}}$ approximation factors, under SETH [6].

Distributed PCP

As mentioned above, when viewing the PCP Theorem as a reduction from CNF to hard-to-approximate CNF, all known constructions suffer from blowup in the number of variables, which is prohibitive for fine-grained reductions. Another (in fact, the original) way to view the PCP Theorem is as a *probabilistically checkable proof*: given an assignment $x \in \{0,1\}^n$, we want to write a proof $\pi(x)$ asserting that x satisfies a known CNF φ. *Probabilistically checkable* means that the verifier should be able to query $\pi(x)$ at a small number of random locations to be convinced that φ has a satisfying assignment. Recall that most reductions from SETH to quadratic time problems construct a gadget for each partial assignment to φ. A key observation in [6] is that if we could construct a "partial PCP" for each partial assignment, the total number of gadgets remains approximately $2^{n/2}$, even if each gadget is now a little bit larger.

Thus, in the *Distributed PCP* challenge, we have two parties (Alice and Bob) who hold partial assignments $\alpha, \beta \in \{0,1\}^{n/2}$ to disjoint subsets of the variables, and want to prove to a verifier that their joint assignment satisfies the public CNF φ. A second key observation in [6] is that this challenge is equivalent to computing Set Disjointness over subsets of the clauses of φ. [6] solved this Set Disjointness problem using (a variant of) the MA-communication protocol of [1]. Here, we need the more efficient IP-communication protocol.

Other PCPs in non-standard models

Different models of "non-traditional" PCPs, such as interactive PCPs [38] and interactive oracle proofs (IOP) [16, 46] have been considered and found "positive" applications in cryptography (e.g. [32, 33, 16]). In particular, [15] obtain a linear-size IOP. It is an open question whether these interactive variants can imply interesting hardness of approximation results [15].

SETH and communication complexity

The connection between the SETH and communication complexity goes back at least to [45] who proved that a computationally efficient sublinear protocol for the 3-party Number-on-Forehead Set Disjointness problem would refute SETH.

2 Preliminaries

2.1 Derandomization and Circuit Lower Bounds

This result will utilize the connection between derandomization and circuit lower bounds which originates in the works of Impagliazzo, Kabanets, and Wigderson [34] and has been optimized significantly by the work of Williams [53], Santhanam and Williams [50], and more recently by Ben-Sasson and Viola [18]. These connections rely on "Succinct PCP" theorems [42, 18], and the recent optimized construction of Ben-Sasson and Viola [18] is essential for our results. Our starting point is the following theorem.

▶ **Theorem 2.1** (Theorem 1.4 in [18]). *Let F_n be a set of function from $\{0,1\}^n$ to $\{0,1\}$ that are efficiently closed under projections (see [18] or Definition 10 in [2]).*

If the acceptance probability of a function of the form

- *AND of fan-in $n^{O(1)}$ of*
- *OR's of fan-in 3 of*
- *functions from $F_{n+O(\log n)}$*

can be distinguished from being $= 1$ or $\leq 1/n^{10}$ in $DTIME(2^n/n^{\omega(1)})$, then there is a function f in E^{NP} on n variables such that $f \notin F_n$.

We apply this theorem where the class F_n is the class of circuits on n variables of size cn, for an arbitrarily large constant $c > 0$, and depth upper bounded by $c \log n$. By applying the deMorgan rule, and then noticing that linear-size log-depth circuits are closed under negations and OR's, we can restate the above theorem as follows (see [2] for a more detailed argument).

▶ **Lemma 2.2.** *To prove that E^{NP} does not have non-uniform circuits of size cn and depth $c \log n$ on n input variables, it is enough to show a deterministic algorithm for the following problem that runs in $2^n/n^{\omega(1)}$ time. Given a circuit over n input variables of the form:*

- *OR of fan-in $n^{O(1)}$ of*
- *circuits of size $3cn$ and depth $c \log n + 2$,*

distinguish between the case where no assignments satisfy it, versus the case in which at least $a \geq 1 - 1/n^{10}$ fraction of the assignments satisfy it.

2.2 Valiant's depth reduction

We will use the classical depth-reduction theorem of Valiant [51] to convert linear-size NC^1 circuits into an OR of CNF's, on which we will apply our distributed PCP techniques. The elegant proof is often given in courses, see e.g. [14].

▶ **Theorem 2.3** (Depth reduction [51]). *For all $\varepsilon > 0$ and $c \geq 1$, we can convert any circuit on n variables of size cn and depth $c \log n$, into an equivalent formula which is OR of $2^{f(c,\varepsilon)\cdot(n/\log\log n)}$ k-CNF's on the same n variables, where $k = O(n^{\varepsilon})$. The reduction runs in $2^{O(n/\log\log n)}$ time.*

We remark that if the circuit is assumed to have the additional "series parallel" property [51], then we can get a stronger depth reduction result where the clause size in the CNF's is constant. This was crucial to the results of [2], but our techniques here allow us to handle much larger CNF's.

Combining Valiant's depth reduction with Lemma 2.2 we conclude that to prove our complexity consequence, it is enough to distinguish unsatisfiable from $> 99\%$ satisfiable on circuits of the form: OR of CNF's with clause size n^{ε}.

▶ **Lemma 2.4.** *To prove Consequence 1.1, it is enough to show a deterministic algorithm for the following problem that runs in $2^n/n^{\omega(1)}$ time. Given a circuit over n input variables of the form:*

- *OR of fan-in $2^{O(n/\log\log n)}$ of*
- *k-CNF's where $k = O(n^{0.1})$,*

distinguish between the case where no assignments satisfy it, versus the case in which at least $a \geq 1 - 1/n^{10}$ fraction of the assignments satisfy it.

2.3 Communication complexity

We use the following IP-communication protocol due to Aaronson and Wigderson [1].

▶ **Theorem 2.5** (Essentially [1, Section 7])**.** *There exists a computationally efficient* [4] IP*-protocol for Set Disjointness over domain $[m]$ in which:*

1. *Merlin and Alice exchange $O(\log m \log\log m)$ bits;*
2. *Bob learns the outcome of $O(\log m \log\log m)$ coins tossed by Alice during the protocol;*
3. *Bob sends Alice $O(\log m)$ bits.*
4. *Alice returns* Accept *or* Reject.

If the sets are disjoint, Alice always accepts; otherwise, Alice rejects with probability at least $1/2$.

3 A surrogate problem: Tropical Tensors

In this section we introduce a new problem a-la Orthogonal Vectors, and show that approximating it with a truly subquadratic determinstic algorithm would be enough to prove the breakthrough Consequence 1.1.

▶ **Definition 3.1** (Tropical Tensors)**.** Our similarity measure s is defined with respect to parameters t and $\ell_1, \ldots, \ell_t$. For two tensors $u, v \in \{0,1\}^{d_1 \times \cdots \times d_t}$, we define their *Tropical Similarity* score with an alternating sequence of E (expecation) and max operators:

$$s(u,v) \triangleq \mathsf{E}_{i_1 \in d_1}\Big[\max_{i_2 \in d_2}\Big\{\mathsf{E}_{i_3 \in d_3}\Big[\cdots \max_{i_t \in d_t}\{u_i \cdot v_i\}\cdots\Big]\Big\}\Big].$$

Given two sets of tensors $A, B \in \{0,1\}^{d_1 \times \cdots \times d_t}$, the Tropical Tensors problem asks to find a pair $a \in A, b \in B$ that maximizes the Tropical Similarity $s(a,b)$.

▶ **Theorem 3.2.** *Let $d_1, \ldots, d_t$ be such that $d_1 d_2 \cdots d_t = N^{o(1)}$. To prove Consequence 1.1 it is enough to design a deterministic $O(N^{2-\varepsilon})$-time algorithm that, given two sets of tensors $A, B \in \{0,1\}^{d_1 \times \cdots \times d_t}$, distinguishes between the following:*

[4] Although [1] do not explicitly consider computational efficiency, it is not hard to make their protocol computationally efficient.

Completeness: A $(1 - 1/\log^{10} N)$*-fraction of the pairs* a, b *have a perfect Tropical Similarity score,* $s(a, b) = 1$.
Soundness: Every pair has low Tropical Similarity score, $s(a, b) = o(1)$.

▶ **Remark 3.3.** Our hardness of approximation for Tropical Tensors continues to hold even in the special case where we only take max's with respect to coordinates of A-tensors. In other words, we could redefine

$$s'(a, b) \triangleq \mathsf{E}_{i_1 \in d_1}\Big[\max_{j_2 \in d_2}\Big\{\mathsf{E}_{i_3 \in d_3}\Big[\cdots \max_{j_t \in d_t}\{a_{i,j} \cdot b_i\} \cdots\Big]\Big\}\Big].$$

(Note that now b is of smaller dimension.)

▶ **Remark 3.4.** Another interesting approximation variant of Tropical Tensors is the challenge of distinguishing between the sets A, B containing at least one pair with perfect Tropical Similarity ($s(a, b) = 1$) versus every pair having subconstant Tropical Similarity ($s(a, b) = o(1)$). Following the same proof outline, one could prove an analog of Theorem 3.2, whereby no $O(N^{2-\varepsilon})$-time algorithms (deterministic or randomized) solve the above problem, assuming SETH for circuits of linear size and logarithmic depth.

Thus we can obtain variants of hardness of approximation results from [6] for LCS Closest Pair, Approximate Regular Expression Matching, and Diameter in Product Metric, based on the latter assumption, which is safer than the standard SETH (i.e. SETH of k-CNF for every constant k).

Proof. Lemma 2.4 tells us that in order to prove Consequence 1.1, it is enough solve the derandomization problem on circuits of the form: an OR over $2^{O(n/\log\log n)}$ CNFs, with clause width $O(n^{0.1})$. In particular, each CNF has at most $m \triangleq 2^{\tilde{O}(n^{0.1})}$ clauses. The focus of this proof will be on reducing a single such CNF to Tropical Tensors. To reduce the OR over $2^{O(n/\log\log n)}$ CNFs to Tropical Tensors, we simply take the max over the Tropical Similarity scores constructed for each CNF.

We do the following for each CNF in the OR. The set A will contain a tensor a^α for each half-assignment $\alpha \in \{0,1\}^{n/2}$ to the CNF. Given half assignment α, let $S_\alpha \subset [m]$ be the set of clauses that it does *not* satisfy, i.e. all the literals determined by α are false. Define B, β, and T_β analogously. Observe that the CNF is satisfied by a pair (α, β) iff the sets S_α, T_β are disjoint.

Recall the IP-communication protocol for Set Disjointness (Theorem 2.5). To obtain subconstant soundness, amplify the soundness of the protocol by repeating a small superconstant number (e.g. $\log\log m$) of times.

We construct the tensors a^α and b^β recursively, using the IP protocol. Each dimension of the tensors corresponds to a message from one of the parties or a coin toss. Each entry corresponds to an entire transcript. Notice that since the total communication complexity is $\tilde{O}(\log m) = \tilde{O}(n^\varepsilon)$, the total number of possible transcripts is at most $d_1 d_2 \cdots d_t = 2^{\tilde{O}(n^\varepsilon)} = N^{o(1)}$.

At the end of the protocol, Bob sends a message. Let $[d_t]$ enumerate over all of Bob's potential messages. For each $i_{-t} \in d_1 \times d_2 \times \cdots \times d_{t-1}$, we set ${a^\alpha}_i \triangleq 1$ iff Alice accepts on message i_t from Bob at the end of the protocol (otherwise, ${a^\alpha}_i \triangleq 0$). Similarly, we set ${b^\beta}_i \triangleq 1$ iff i_t is the message that Bob sends. Hence the contribution to the Tropical Similarity is one iff Alice accepts Bob's message at the end of the protocol.

For the rest of the coordinates we take E over random coin tosses, and max over Merlin's potential messages. Hence the Tropical Similarity $s(a^\alpha, b^\beta)$ is exactly equal to the probability that Alice accepts at the end of the IP protocol given for input sets S_α, T_β.

Hence there is a one-to-one correspondence between satisfying assignments and pairs with perfect Tropical Similarity score; similarly there is a one-to-one correspondence between unsatisfying assignments and pairs with subconstant Tropical Similarity score.

Finally, we add another outside max to account for the large OR over $2^{O(n/\log\log n)}$ CNFs. ◀

4 LCS

In this section we provide a gap-preserving reduction from Tropical Tensors to Longest Common Subsequence. Together with Theorem 3.2 this completes our proof of Theorem 1.2.

Proof of Theorem 1.2. We begin with the hard instance of Tropical Tensors from Theorem 3.2. We encode each of the tensors as a string-gadget over alphabet Σ, and then concatenate all the gadgets (in arbitrary order). Unlike previous fine-grained reductions for LCS and related problems (e.g. [7, 10, 4, 22, 5, 11, 2, 21]), we do not need any padding between gadgets, since the gap we obtained for Tropical Tensors is so large.

Bit gadgets

We construct the gadget for each tensor recursively. At the base of our recursion, we use the following encoding for each bit. For each coordinate $i \in [d_1] \times \cdots \times [d_t]$, we reserve a special symbol $i \in \Sigma$. We will also have two special symbols $\perp^A, \perp^B \in \Sigma$. Thus in total $|\Sigma| = d_1 d_2 \cdots d_t + 2 = N^{o(1)}$. Finally, we are ready to define the bit-gadgets:

$$x_i(a) \triangleq \begin{cases} i & a_i = 1 \\ \perp^A & a_i = 0 \end{cases},$$

and

$$y_i(b^\beta) \triangleq \begin{cases} i & b_i = 1 \\ \perp^B & b_i = 0 \end{cases}.$$

Observe that now $LCS\left(x_i(a), y_i(b)\right) = a_i \cdot b_i$.

Tensor gadgets

We now recursively combine gadgets to implement the max and E operators. In order to implement max operators, we concatenate the corresponding x-gadgets, and concatenate the respective y-gadgets in reverse order. For example, for any fixed choice of $i_{-t} = (i_1, \ldots, i_{t-1})$, we combine bit-gadgets across the last dimension as follows:

$$x_{i_{-t}}(a) \triangleq x_{i_{-t},1}(a) \circ x_{i_{-t},2}(a) \circ \cdots \circ x_{i_{-t},d_t}(a)$$
$$y_{i_{-t}}(b) \triangleq y_{i_{-t},d_t}(b) \circ \cdots \circ y_{i_{-t},2}(b) \circ y_{i_{-t},1}(b).$$

Notice that we now have that $LCS\left(x_{i_{-t}}(a), y_{i_{-t}}(b)\right) = \max_{i_t \in [d_t]} LCS\left(x_{i_{-t},i_t}(a), y_{i_{-t},i_t}(b)\right)$.

We implement summations (E), we concatenate both the x and the y gadgets in the same order. For example, for the first dimension, we define:

$$x(a) \triangleq x_1(a) \circ x_2(a) \circ \cdots \circ x_{d_1}(a)$$
$$y(b) \triangleq y_1(b) \circ y_2(b) \circ \cdots \circ y_{d_1}(b).$$

Notice that we now have that $LCS\left(x(a), b(a)\right) = d_1 \cdot \mathsf{E}_{i_1 \in [d_1]}\left[LCS\left(x_{i_1}(a), y_{i_1}(b)\right)\right]$.

Therefore, by induction, we have that

$$\begin{aligned}\frac{1}{D} LCS\left(x(a), y(b)\right) &= \mathsf{E}_{i_1 \in d_1}\left[\max_{i_2 \in d_2}\left\{\mathsf{E}_{i_3 \in d_3}\left[\cdots \max_{i_t \in d_t}\{LCS\left(x_i(a), y_i(b)\right)\} \cdots\right]\right\}\right] \\ &= \mathsf{E}_{i_1 \in d_1}\left[\max_{i_2 \in d_2}\left\{\mathsf{E}_{i_3 \in d_3}\left[\cdots \max_{i_t \in d_t}\{a_i \cdot b_i\} \cdots\right]\right\}\right] \\ &= s(a, b),\end{aligned}$$

where $D \triangleq d_1 d_3 d_5 \cdots d_{t-1}$ is the normalization factor.

The final strings

Finally, we construct the strings x, y by concatenating the $2^{n/2}$ tensor gadgets. We call a pair of tensors a, b "good" if $s(a, b) = 1$ and "bad" if $s(a, b) = o(1)$.

Completeness

Assume that there are at least $(1 - 1/\log^{10} n) \cdot n^2$ good pairs of tensors. We consider a set of $2n - 1$ alignments between x and y: For each shift $k \in [2n - 1]$ define the alignment $\mathcal{A}_k$ that matches the tensor gadget of tensor $a_i \in A$ to the tensor gadget of $b_j \in B$ *optimally* where $j = i + k - n$, and if $j \notin [n]$ then we do not match the gadget of a_i at all. Since the alignments of gadgets to each other are made optimally, their contribution is exactly $LCS(x(a_i), y(b_j)) = D \cdot s(a_i, b_j)$. Observe that for all $i, j \in [n]$ there is exactly one k such that the gadgets of a_i and b_j are matched in $\mathcal{A}_k$, and so the total LCS score of all of these $2n - 1$ alignments is at least

$$(1 - 1/\log^{10} n) \cdot n^2 \cdot D \cdot 1.$$

Therefore at least one of these alignments has score more than $D \cdot n/2$.

Soundness

Assume that all pairs of tensors are bad. In this case, any alignment between two tensor gadgets has score at most $LCS(x(a_i), y(b_j)) = o(1) \cdot D$. We can upper bound the score of any alignment between x and y by upper bounding the number of tensor gadgets participating in the alignment. We say that a pair is participating in the alignment if any of their letters are matched to each other. Due to the non-crossing nature of alignments, we can model all pairs participating in an alignment as the edges in a bipartite *planar* graph, and it follows that there can be at most $2n$ such edges. Therefore, the score of any alignment is upper bounded by $o(1) \cdot D \cdot 2n$. ◀

Acknowledgements. We thank Scott Aaronson, Mika Goos, Elad Haramaty, and Ryan Williams for helpful discussions and suggestions.

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 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 11:1–11:26. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2017. doi:10.4230/LIPIcs.ITCS.2017.11.

3 Amir Abboud, Arturs Backurs, Karl Bringmann, and Marvin Künnemann. Fine-grained complexity of analyzing compressed data: Quantifying improvements over decompress-and-solve. In Chris Umans, editor, *58th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2017, Berkeley, CA, USA, October 15-17, 2017*, pages 192–203. IEEE Computer Society, 2017. doi:10.1109/FOCS.2017.26.

4 Amir Abboud, Arturs Backurs, and Virginia Vassilevska Williams. Tight hardness results for LCS and other sequence similarity measures. In *Proc. of the 56th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 59–78, 2015.

5 Amir Abboud, Thomas Dueholm Hansen, Virginia Vassilevska Williams, and Ryan Williams. Simulating branching programs with edit distance and friends: or: a polylog shaved is a lower bound made. In *Proc. of the 48th STOC*, pages 375–388, 2016.

6 Amir Abboud, Aviad Rubinstein, 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.

7 Amir Abboud, Virginia Vassilevska Williams, and Oren Weimann. Consequences of faster alignment of sequences. In *Proc. of the 41st International Colloquium on Automata, Languages, and Programming (ICALP)*, pages 39–51, 2014.

8 Stephen F Altschul, Thomas L Madden, Alejandro A Schäffer, Jinghui Zhang, Zheng Zhang, Webb Miller, and David J Lipman. Gapped blast and psi-blast: a new generation of protein database search programs. *Nucleic acids research*, 25(17):3389–3402, 1997.

9 Alexandr Andoni, Robert Krauthgamer, and Krzysztof Onak. Polylogarithmic approximation for edit distance and the asymmetric query complexity. In *FOCS*, pages 377–386, 2010.

10 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.

11 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.

12 Ziv Bar-Yossef, TS Jayram, Robert Krauthgamer, and Ravi Kumar. Approximating edit distance efficiently. In *Foundations of Computer Science, 2004. Proceedings. 45th Annual IEEE Symposium on*, pages 550–559. IEEE, 2004.

13 Tuğkan Batu, Funda Ergun, and Cenk Sahinalp. Oblivious string embeddings and edit distance approximations. In *Proceedings of the seventeenth annual ACM-SIAM symposium on Discrete algorithm*, pages 792–801. Society for Industrial and Applied Mathematics, 2006.

14 Paul Beame. Lecture notes on circuit reducibility, depth reduction, and parallel arithmetic, April 2008.

15 Eli Ben-Sasson, Alessandro Chiesa, Ariel Gabizon, Michael Riabzev, and Nicholas Spooner. Short interactive oracle proofs with constant query complexity, via composition and sumcheck. *IACR Cryptology ePrint Archive*, 2016:324, 2016. URL: http://eprint.iacr.org/2016/324.

16 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.

17 Eli Ben-Sasson, Yohay Kaplan, Swastik Kopparty, Or Meir, and Henning Stichtenoth. Constant rate pcps for circuit-sat with sublinear query complexity. *J. ACM*, 63(4):32:1–32:57, 2016. doi:10.1145/2901294.

18 Eli Ben-Sasson and Emanuele Viola. Short pcps with projection queries. In *ICALP, Part I*, pages 163–173, 2014.

19 Lasse Bergroth, Harri Hakonen, and Timo Raita. New approximation algorithms for longest common subsequences. In *String Processing and Information Retrieval: A South American Symposium, 1998. Proceedings*, pages 32–40. IEEE, 1998.

20 Lasse Bergroth, Harri Hakonen, and Timo Raita. A survey of longest common subsequence algorithms. In *String Processing and Information Retrieval, 2000. SPIRE 2000. Proceedings. Seventh International Symposium on*, pages 39–48. IEEE, 2000.

21 Karl Bringmann, Allan Grønlund, and Kasper Green Larsen. A dichotomy for regular expression membership testing. In Chris Umans, editor, *58th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2017, Berkeley, CA, USA, October 15-17, 2017*, pages 307–318. IEEE Computer Society, 2017. doi:10.1109/FOCS.2017.36.

22 Karl Bringmann and Marvin Kunnemann. Quadratic conditional lower bounds for string problems and dynamic time warping. In *Proc. of the 56th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 79–97, 2015.

23 Diptarka Chakraborty, Elazar Goldenberg, and Michal Koucký. Streaming algorithms for embedding and computing edit distance in the low distance regime. 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 712–725. ACM, 2016. doi:10.1145/2897518.2897577.

24 Parinya Chalermsook, Marek Cygan, Guy Kortsarz, Bundit Laekhanukit, Pasin Manurangsi, Danupon Nanongkai, and Luca Trevisan. From gap-eth to fpt-inapproximability: Clique, dominating set, and more. In Chris Umans, editor, *58th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2017, Berkeley, CA, USA, October 15-17, 2017*, pages 743–754. IEEE Computer Society, 2017. doi:10.1109/FOCS.2017.74.

25 F Chin and Chung Keung Poon. Performance analysis of some simple heuristics for computing longest common subsequences. *Algorithmica*, 12(4-5):293–311, 1994.

26 Maxime Crochemore, Costas S Iliopoulos, Yoan J Pinzon, and James F Reid. A fast and practical bit-vector algorithm for the longest common subsequence problem. *Information Processing Letters*, 80(6):279–285, 2001.

27 J Boutet de Monvel. Extensive simulations for longest common subsequences. *The European Physical Journal B-Condensed Matter and Complex Systems*, 7(2):293–308, 1999.

28 Irit Dinur. The PCP theorem by gap amplification. *J. ACM*, 54(3):12, 2007. doi:10.1145/1236457.1236459.

29 Irit Dinur. Mildly exponential reduction from gap 3sat to polynomial-gap label-cover. *Electronic Colloquium on Computational Complexity (ECCC)*, 23:128, 2016. URL: http://eccc.hpi-web.de/report/2016/128.

30 Andrew Drucker. PCPs for Arthur-Merlin Games and Communication Protocols. Master's thesis, Massachusetts Institute of Technology, 2010. URL: http://people.csail.mit.edu/andyd/Drucker_SM_thesis.pdf.

31 Paweł Gawrychowski. Faster algorithm for computing the edit distance between slp-compressed strings. In *International Symposium on String Processing and Information Retrieval*, pages 229–236. Springer, 2012.

32 Shafi Goldwasser, Yael Tauman Kalai, and Guy N. Rothblum. Delegating computation: Interactive proofs for muggles. *J. ACM*, 62(4):27:1–27:64, 2015. doi:10.1145/2699436.

33 Vipul Goyal, Yuval Ishai, Mohammad Mahmoody, and Amit Sahai. Interactive locking, zero-knowledge pcps, and unconditional cryptography. In Tal Rabin, editor, *Advances in Cryptology - CRYPTO 2010, 30th Annual Cryptology Conference, Santa Barbara, CA, USA, August 15-19, 2010. Proceedings*, volume 6223 of *Lecture Notes in Computer Science*, pages 173–190. Springer, 2010. doi:10.1007/978-3-642-14623-7_10.

34 Russell Impagliazzo, Valentine Kabanets, and Avi Wigderson. In search of an easy witness: Exponential time vs. probabilistic polynomial time. *Journal of Computer and System Sciences*, 65(4):672–694, 2002.

35 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.

36 Russell Impagliazzo, Ramamohan Paturi, and Francis Zane. Which problems have strongly exponential complexity? *J. Comput. Syst. Sci.*, 63(4):512–530, 2001. doi:10.1006/jcss.2001.1774.

37 Hamid Jahanjou, Eric Miles, and Emanuele Viola. Local reductions. In *International Colloquium on Automata, Languages, and Programming*, pages 749–760. Springer, 2015.

38 Yael Tauman Kalai and Ran Raz. Interactive PCP. In Luca Aceto, Ivan Damgård, Leslie Ann Goldberg, Magnús M. Halldórsson, Anna Ingólfsdóttir, and Igor Walukiewicz, editors, *Automata, Languages and Programming, 35th International Colloquium, ICALP 2008, Reykjavik, Iceland, July 7-11, 2008, Proceedings, Part II - Track B: Logic, Semantics, and Theory of Programming & Track C: Security and Cryptography Foundations*, volume 5126 of *Lecture Notes in Computer Science*, pages 536–547. Springer, 2008. doi:10.1007/978-3-540-70583-3_44.

39 Gad M Landau, Eugene W Myers, and Jeanette P Schmidt. Incremental string comparison. *SIAM Journal on Computing*, 27(2):557–582, 1998.

40 Pasin Manurangsi. Almost-polynomial ratio eth-hardness of approximating densest k-subgraph. 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 954–961. ACM, 2017. doi:10.1145/3055399.3055412.

41 Pasin Manurangsi and Prasad Raghavendra. A birthday repetition theorem and complexity of approximating dense csps. 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 78:1–78:15. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2017. doi:10.4230/LIPIcs.ICALP.2017.78.

42 Thilo Mie. Short pcpps verifiable in polylogarithmic time with o (1) queries. *Annals of Mathematics and Artificial Intelligence*, 56(3-4):313–338, 2009.

43 Gonzalo Navarro. A guided tour to approximate string matching. *ACM computing surveys (CSUR)*, 33(1):31–88, 2001.

44 Rafail Ostrovsky and Yuval Rabani. Low distortion embeddings for edit distance. *Journal of the ACM (JACM)*, 54(5):23, 2007.

45 Mihai Patrascu and Ryan Williams. On the possibility of faster SAT algorithms. In Moses Charikar, editor, *Proceedings of the Twenty-First Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2010, Austin, Texas, USA, January 17-19, 2010*, pages 1065–1075. SIAM, 2010. doi:10.1137/1.9781611973075.86.

46 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 Comput-*

ing, STOC 2016, Cambridge, MA, USA, June 18-21, 2016, pages 49–62. ACM, 2016. doi:10.1145/2897518.2897652.

47 Balaram Saha. The dyck language edit distance problem in near-linear time. In *Foundations of Computer Science (FOCS), 2014 IEEE 55th Annual Symposium on*, pages 611–620. IEEE, 2014.

48 Barna Saha. Language edit distance and maximum likelihood parsing of stochastic grammars: Faster algorithms and connection to fundamental graph problems. In *Foundations of Computer Science (FOCS), 2015 IEEE 56th Annual Symposium on*, pages 118–135. IEEE, 2015.

49 Barna Saha. Fast & space-efficient approximations of language edit distance and RNA folding: An amnesic dynamic programming approach. In Chris Umans, editor, *58th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2017, Berkeley, CA, USA, October 15-17, 2017*, pages 295–306. IEEE Computer Society, 2017. doi:10.1109/FOCS.2017.35.

50 Rajesh Santhanam and Ross Williams. On medium-uniformity and circuit lower bounds. In *Computational Complexity (CCC), 2013 IEEE Conference on*, pages 15–23. IEEE, 2013.

51 Leslie G Valiant. *Graph-theoretic arguments in low-level complexity*. Springer, 1977.

52 R. Ryan Williams. A new algorithm for optimal 2-constraint satisfaction and its implications. *Theoretical Computer Science*, 348(2–3):357–365, 2005.

53 Ryan Williams. Improving exhaustive search implies superpolynomial lower bounds. *SIAM Journal on Computing*, 42(3):1218–1244, 2013.

ETH-Hardness of Approximating 2-CSPs and Directed Steiner Network*

Irit Dinur[1] and Pasin Manurangsi[2]

1 **Weizmann Institute of Science, Rehovot, Israel**
irit.dinur@weizmann.ac.il
2 **University of California, Berkeley, USA**
pasin@berkeley.edu

Abstract

We study 2-ary constraint satisfaction problems (2-CSPs), which can be stated as follows: given a constraint graph $G = (V, E)$, an alphabet set Σ and, for each edge $\{u, v\} \in E$, a constraint $C_{uv} \subseteq \Sigma \times \Sigma$, the goal is to find an assignment $\sigma : V \to \Sigma$ that satisfies as many constraints as possible, where a constraint C_{uv} is said to be satisfied by σ if $(\sigma(u), \sigma(v)) \in C_{uv}$.

While the approximability of 2-CSPs is quite well understood when the alphabet size $|\Sigma|$ is constant (see e.g. [37]), many problems are still open when $|\Sigma|$ becomes super constant. One open problem that has received significant attention in the literature is whether it is hard to approximate 2-CSPs to within a polynomial factor of both $|\Sigma|$ and $|V|$ (i.e. $(|\Sigma||V|)^{\Omega(1)}$ factor). As a special case of the so-called Sliding Scale Conjecture, Bellare et al. [5] suggested that the answer to this question might be positive. Alas, despite many efforts by researchers to resolve this conjecture (e.g. [39, 4, 20, 21, 35]), it still remains open to this day.

In this work, we separate $|V|$ and $|\Sigma|$ and ask a closely related but weaker question: is it hard to approximate 2-CSPs to within a polynomial factor of $|V|$ (while $|\Sigma|$ may be super-polynomial in $|V|$)? Assuming the exponential time hypothesis (ETH), we answer this question positively: unless ETH fails, no polynomial time algorithm can approximate 2-CSPs to within a factor of $|V|^{1-1/\log^{\beta}|V|}$ for some $\beta > 0$. Note that our ratio is not only polynomial but also almost linear. This is almost optimal since a trivial algorithm yields an $O(|V|)$-approximation for 2-CSPs.

Thanks to a known reduction [25, 16] from 2-CSPs to the Directed Steiner Network (DSN) problem, our result implies an inapproximability result for the latter with polynomial ratio in terms of the number of demand pairs. Specifically, assuming ETH, no polynomial time algorithm can approximate DSN to within a factor of $k^{1/4-o(1)}$ where k is the number of demand pairs. The ratio is roughly the square root of the approximation ratios achieved by best known polynomial time algorithms [15, 26], which yield $O(k^{1/2+\varepsilon})$-approximation for every constant $\varepsilon > 0$.

Additionally, under Gap-ETH, our reduction for 2-CSPs not only rules out polynomial time algorithms, but also fixed parameter tractable (FPT) algorithms parameterized by the number of variables $|V|$. These are algorithms with running time $g(|V|) \cdot |\Sigma|^{O(1)}$ for some function g. Similar improvements apply for DSN parameterized by the number of demand pairs k.

1998 ACM Subject Classification F.2.2 Nonnumerical Algorithms and Problems

Keywords and phrases Hardness of Approximation, Constraint Satisfaction Problems, Directed Steiner Network, Parameterized Complexity

Digital Object Identifier 10.4230/LIPIcs.ITCS.2018.36

* This work was supported by BSF grant 2014371 and NSF grants CCF 1540685, CCF 1655215.

1 Introduction

We study 2-ary constraint satisfaction problems (2-CSPs): given a constraint graph $G = (V, E)$, an alphabet Σ and, for each edge $\{u, v\} \in E$, a constraint $C_{uv} \subseteq \Sigma \times \Sigma$, the goal is to find an assignment $\sigma : V \to \Sigma$ that satisfies as many constraints as possible, where a constraint C_{uv} is satisfied by σ if $(\sigma(u), \sigma(v)) \in C_{uv}$. Throughout the paper, we use k to denote the number of variables $|V|$, n to denote the alphabet size $|\Sigma|$, and N to denote nk.

Constraint satisfaction problems (CSPs) and their inapproximability have been studied extensively since the proof of the PCP theorem in the early 90's [3, 2]. Most of the effort has been directed towards understanding the approximability of CSPs with constant arity and constant alphabet size, leading to a reasonable if yet incomplete understanding of the landscape [27, 31, 37, 12]. When the alphabet size grows, the Sliding Scale Conjecture (SSC) [5] predicts that the hardness of approximation ratio will grow as well, and be at least a constant power of the alphabet size n. This has been confirmed for values of n up to $2^{(\log N)^{1-\delta}}$ (see [39, 4, 20]). Proving the same for n that is itself a constant power of N is the so-called polynomial SSC and is still open. Before we proceed, let us note that the results of [39, 4, 20] work only for arity larger than two and, hence, do not imply inapproximability for 2-CSPs. We will discuss the special case of 2-CSPs in more details below.

The polynomial SSC has been approached from different angles. In [21] the authors try to find the smallest arity and alphabet size such that the hardness factor is a constant power of n, and in [19] the conjecture is shown to follow (in some weaker sense) from the Gap-ETH hypothesis, which we discuss in more details later. In this work we focus on yet another angle, which is to separate n and k and ask whether it is hard to approximate constant arity CSPs to within a factor that is a constant power of k (but possibly not a constant power of n). Observe here that obtaining NP-hardness of $k^{\Omega(1)}$ factor is likely to be as hard as obtaining one with $N^{\Omega(1)}$; this is because CSPs can be solved exactly in time $n^{O(k)}$, which means that, unless $\mathsf{NP} \nsubseteq \bigcap_{\varepsilon>0} \mathsf{DTIME}(2^{n^\varepsilon})$, NP-hard instances of CSPs must have $k = \mathrm{poly}(N)$.

This motivates us to look for hardness from assumptions stronger than $\mathsf{P} \neq \mathsf{NP}$. Specifically, our result will be based on the Exponential Time Hypothesis (ETH), which states that no subexponential time algorithm can solve 3-SAT (Conjecture 5). We show that, assuming ETH, no polynomial time algorithm can approximate 2-CSPs to within an almost linear ratio in k, as stated below. This is almost optimal since there is a straightforward $(k/2)$-approximation for 2-CSPs, by simply satisfying all constraints that touch a variable with highest degree.

▶ **Theorem 1** (Main Theorem). *Assuming ETH, there exists a constant $\beta > 0$ such that, no algorithm can, given a 2-CSP instance Γ with alphabet size n and k variables such that the constraint graph is complete, distinguish between the following two cases in polynomial time:*

- *(Completeness)* $\mathrm{val}(\Gamma) = 1$*, and,*
- *(Soundness)* $\mathrm{val}(\Gamma) < 2^{(\log k)^{1-\beta}}/k$.

Here $\mathrm{val}(\Gamma)$ *denotes the maximum fraction of edges satisfied by any assignment.*

To paint a full picture of how our result stands in comparison to previous results, let us state what is known about the approximability of 2-CSPs; due to the vast literature regarding 2-CSPs, we will focus only the regime of large alphabets which is most relevant to our setting. In terms of NP-hardness, the best known inapproximability ratio is $(\log N)^c$ for every constant $c > 0$; this follows from Moshkovitz-Raz PCP [36] and the Parallel Repetition Theorem for the low value regime [24]. Assuming a slightly stronger assumption that NP is not contained in quasipolynomial time (i.e. $\mathsf{NP} \nsubseteq \bigcup_{c>0} \mathsf{DTIME}(n^{(\log n)^c})$), 2-CSP is hard to

approximate to within a factor of $2^{(\log N)^{1-\delta}}$ for every constant $\delta > 0$; this can be proved by applying Raz's original Parallel Repetition Theorem [38] to the PCP Theorem. In [19], the author observed that running time for parallel repetition can be reduced by looking at unordered sets instead of ordered tuples. This observation implies that[1], assuming ETH, no polynomial time $N^{1/(\log\log\log N)^c}$-approximation algorithm exists for 2-CSPs for some constant $c > 0$. Moreover, under Gap-ETH (which will be stated shortly), it was shown that, for every sufficiently small $\varepsilon > 0$, any N^ε-approximation algorithm must run in time $N^{\Omega(\exp(1/\varepsilon))}$. Note that, while this latest result comes close to the polynomial sliding scale conjecture, it does not quite resolve the conjecture yet. In particular, even the weak form of the conjecture which postulates that there exists $\delta > 0$ for which no polynomial time algorithm can approximate 2-CSPs to within N^δ factor of the optimum does not follow from [19]. Nevertheless, the Gap-ETH-hardness of [19] does imply that, for any function $f = o(1)$, no polynomial time algorithm can approximate 2-CSPs to within a factor of $N^{f(N)}$.

In all results mentioned above, the constructions give 2-CSP instances in which the alphabet size n is smaller than the number of variables k. In other words, even if we aim for an inapproximability ratio in terms of k instead of N, we still get the same ratios as stated above. Thus, our result is the first hardness of approximation for 2-CSPs with $k^{\Omega(1)}$ factor. Note again that our result rules out any polynomial time algorithm and not just $N^{O(\exp(1/\varepsilon))}$-time algorithm ruled out by [19]. Moreover, our ratio is almost linear in k whereas the result of [19] only holds for ε that is sufficiently small depending on the parameters of Gap-ETH.

An interesting feature of our reduction is that it produces 2-CSP instances with the alphabet size n that is much larger than k. This is reminiscent of the setting of 2-CSPs parameterized by the number of variables k. In this setting, the algorithm's running time is allowed to depend not only polynomially on N but also on any function of k (i.e. $g(k)\cdot\mathrm{poly}(N)$ running time for some function g); such algorithm is called a *fixed parameter tractable (FPT)* algorithm parameterized by k. The question here is whether this added running time can help us approximate the problem beyond the $O(k)$ factor achieved by the straightforward algorithm. We show that, even in this parameterized setting, the trivial algorithm is still essentially optimal (up to lower order terms). This result holds under the Gap Exponential Time Hypothesis (Gap-ETH), a strengthening of ETH which states that, for some $\varepsilon > 0$, even distinguishing between a satisfiable 3-CNF formula and one which is not even $(1-\varepsilon)$-satisfiable cannot be done in subexponential time (see Conjecture 7). Moreover, under this stronger assumption, we improve the lower order term in our inapproximability ratio from $2^{(\log k)^{1-\beta}}$ for some $\beta > 0$ to $2^{(\log k)^{1/2+\rho}}$ for any $\rho > 0$. This result is stated formally below.

▶ **Theorem 2.** *Assuming Gap-ETH, for any constant $\rho > 0$ and any function g, no algorithm can, given a 2-CSP instance Γ with alphabet size n and k variables such that the constraint graph is complete, distinguish between the following two cases in $g(k)\cdot(nk)^{O(1)}$ time:*

- *(Completeness)* $\mathrm{val}(\Gamma) = 1$*, and,*
- *(Soundness)* $\mathrm{val}(\Gamma) < 2^{(\log k)^{1/2+\rho}}/k$*.*

To the best of our knowledge, the only previous inapproximability result for parameterized 2-CSPs is from [16]. There the authors showed that, assuming Gap-ETH, no $k^{o(1)}$-approximation $g(k)\cdot(nk)^{O(1)}$-time algorithm exists; this is shown via a simple reduction from parameterized inapproximbability of Densest-k Subgraph from [11] (which is in turn based on a construction from [33]). Our result is a direct improvement over this result.

[1] In [19], only the Gap-ETH-hardness result is stated. However, the ETH-hardness result follows easily by invoking a PCP theorem (Theorem 6 below) to get a gap instance.

We end our discussion on 2-CSPs by noting that, while our results suggest that the trivial algorithm achieves an essentially optimal ratio in terms of k, non-trivial approximation is possible when we measure the ratio in terms of N instead of k: in particular, a polynomial time $O(N^{1/3})$-approximation algorithm is known for the problem [14].

Direct Steiner Network

As a corollary of our hardness of approximation results for 2-CSPs, we obtain an inapproximability result for Directed Steiner Network with polynomial ratio in terms of the number of demand pairs. In the Directed Steiner Network (DSN) problem (sometimes referred to as the Directed Steiner Forest problem [26, 17]), we are given an edge-weighed directed graph G and a set $\mathcal{D}$ of k demand pairs $(s_1, t_1), \ldots, (s_k, t_k) \in V \times V$ and the goal is to find a subgraph H of G with minimum weight such that there is a path in H from s_i to t_i for every $i \in [k]$. DSN was first studied in the approximation algorithms context by Charikar et al. [13] who gave a polynomial time $\tilde{O}(k^{2/3})$-approximation algorithm for the problem. This ratio was later improved to $O(k^{1/2+\varepsilon})$ for every $\varepsilon > 0$ by Chekuri et al. [15]. Later, a different algorithm with similar approximation ratio was proposed by Feldman et al. [26].

Algorithms with approximation ratios in terms of the number of vertices n have also been devised [26, 9, 17, 1]. In this case, the best known algorithm is that of Berman et al. [9], which yields an $O(n^{2/3+\varepsilon})$-approximation for every constant $\varepsilon > 0$ in polynomial time.

On the hardness side, there exists a known reduction from 2-CSP to DSN that preserves approximation ratio to within polynomial factor[2] [25]. Hence, known hardness of approximation of 2-CSPs translate immediately to that of DSN: it is NP-hard to approximate to within any polylogarithmic ratio, it is hard to approximate to within $2^{\log^{1-\varepsilon} n}$ factor unless $\mathsf{NP} \subseteq \mathsf{QP}$, and it is Gap-ETH-hard to approximate to within $n^{o(1)}$ factor. Note that, since k is always bounded above by n^2, these hardness results also hold when n is replaced by k in the ratios. Recently, this reduction was also used by Chitnis et al. [16] to rule out $k^{o(1)}$-FPT-approximation algorithm for DSN parameterized by k assuming Gap-ETH. Alas, none of these results achieve ratios that are polynomial in either n or k and it remains open whether DSN is hard to approximate to within a factor that is polynomial in n or in k.

By plugging our hardness results for 2-CSPs into the reduction, we immediately get (Gap)-ETH-hardness of approximating DSN to within a factor of $k^{1/4-o(1)}$ as stated below.

▶ **Corollary 3.** *Assuming ETH, there exists a constant $\beta' > 0$ such that, there is no polynomial time $\frac{k^{1/4}}{2^{(\log k)^{1-\beta'}}}$-approximation algorithm for DSN.*

▶ **Corollary 4.** *Assuming Gap-ETH, for any constant $\rho' > 0$ and any function g, there is no $g(k) \cdot (nk)^{O(1)}$-time $\frac{k^{1/4}}{2^{(\log k)^{1/2+\rho'}}}$-approximation algorithm for DSN.*

In other words, if one wants a polynomial time approximation algorithm with ratio depending only on k and not n, then the approximation ratios from the algorithms of [15, 26] are roughly within a square of the best possible approximation ratio. To the best of our knowledge, these are the first inapproximability results of DSN whose ratios are polynomial in k.

[2] That is, for any non-decreasing function ρ , if DSN admits $\rho(nk)$-approximation in polynomial time, then 2-CSP also admits $\rho(nk)^c$-approximation polynomial time for some absolute constant c.

Agreement tests

Our main result is proved through an agreement testing argument. In agreement testing there is a universe V, a collection of subsets $S_1, \ldots, S_k \subseteq V$, and for each subset S_i we are given a local function $\sigma_{S_i} : S_i \to \{0,1\}$. A pair of subsets are said to *agree* if their local functions agree on every element in the intersection. The goal is, given a non-negligible fraction of agreeing pairs, to deduce the existence of a global function $g : V \to \{0,1\}$ that coincides with many of the local functions. For a more complete description see [22].

Agreement tests capture a natural local to global statement and are present in essentially all PCPs, for example they appear explicitly in the line vs. line and plane vs. plane low degree tests [40, 4, 39]. Our reduction is based on a combinatorial agreement test, where the universe is $[n]$ and the subsets $S_1, \ldots, S_k$ have $\Omega(n)$ elements each and are "in general position", namely they behave like subsets chosen independently at random. A convenient feature about this setting is that every pair of subsets intersect.

Since we are aiming for a large gap, the agreement test must work (i.e., yield a global function) with a very small fraction of agreeing pairs, which in our case is close to $1/k$.

In this small agreement regime the idea, as pioneered in the work of Raz-Safra citeRazS97, is to zero in on a sub-collection of subsets that is (almost) perfectly consistent. From this sub-collection it is easy to recover a global function and show that it coincides almost perfectly with the local functions in the sub-collection. A major difference between our combinatorial setting and the algebraic setting of Raz-Safra is the lack of "distance" in our case: we can not assume that two distinct local functions differ on many points (in contrast, this is a key feature of low degree polynomials). We overcome this by considering different "strengths" of agreement, depending on the fraction of points on which the two subsets agree. This notion too is present in several previous works on combinatorial agreement tests [28, 23].

Hardness of Approximation through Subexponential Time Reductions

Our result is one of the many results in recent years that show hardness of approximation via subexponential time reductions. These results are often based on ETH and its variants. Proposed by Impagliazzo and Paturi [29], ETH can be formally stated as follows:

▶ **Conjecture 5** (Exponential Time Hypothesis (ETH) [29]). *There exist constants $\delta > 0$ such that no algorithm can decide whether any given 3-CNF formula is satisfiable in time $O(2^{\delta m})$ where m denotes the number of clauses*[3].

A crucial ingredient in most reductions in this line of work is a nearly-linear size PCP Theorem. For the purpose of our work, the PCP Theorem can be viewed as a polynomial time transformation of a 3-SAT instance $\widetilde{\Phi}$ to another 3-SAT instance Φ that creates a gap between the YES and NO cases. Specifically, if $\widetilde{\Phi}$ is satisfiable, Φ remains satisfiable. On the other hand, if $\widetilde{\Phi}$ is unsatisfiable, then Φ is not only unsatisfiable but it is also not even $(1-\varepsilon)$-satisfiable for some constant $\varepsilon > 0$ (i.e. no assignment satisfies $(1-\varepsilon)$ fraction of clauses). The "nearly-linear size" part refers to the size of the new instance Φ compared to that of $\widetilde{\Phi}$. Currently, the best known dependency in this form of the PCP Theorem between the two sizes is quasi-linear (i.e. with a polylogarithmic blow-up), as stated below.

[3] The original conjecture states the lower bound as exponential in terms of the number of variables not clauses. However, thanks to the sparsification lemma [30], these two versions are equivalent.

▶ **Theorem 6** (Quasi-Linear Size PCP [8, 18]). *For some constants $\varepsilon, \Delta, c > 0$, there is a polynomial time algorithm that, given any 3-CNF formula $\widetilde{\Phi}$ with m clauses, produces another 3-CNF formula Φ with $O(m \log^c m)$ clauses such that*

- *(Completeness) if* $\mathrm{val}(\widetilde{\Phi}) = 1$*, then* $\mathrm{val}(\Phi) = 1$*, and,*
- *(Soundness) if* $\mathrm{val}(\widetilde{\Phi}) < 1$*, then* $\mathrm{val}(\Phi) < 1 - \varepsilon$*, and,*
- *(Bounded Degree) each variable in Φ appears in at most Δ clauses.*

ETH-hardness of approximation proofs usually proceed in two steps. First, the PCP Theorem is invoked to reduce a 3-SAT instance $\widetilde{\Phi}$ of size m to an instance of the gap version of 3-SAT Φ of size $m' = O(m \log^c m)$. Second, the gap version of 3-SAT is reduced in subexponential time to the problem at hand. As long as the reduction takes time $2^{o(m'/\log^c m')} = 2^{o(m)}$, we can obtain hardness of approximation result for the latter problem. This is in contrast to proving NP-hardness of approximation for which a polynomial time reduction is required.

Another related but stronger version of ETH that we will also employ is Gap-ETH, which states that even the gap version of 3-SAT cannot be solved in subexponential time:

▶ **Conjecture 7** (Gap Exponential Time Hypothesis (Gap-ETH) [19, 34]). *There exist constants $\delta, \varepsilon, \Delta > 0$ such that no algorithm can, given any 3-CNF formula Φ such that each of its variable appears in at most Δ clauses*[4]*, distinguish between the following two cases in time $O(2^{\delta m})$ time where m denotes the number of clauses:*

- *(Completeness)* $\mathrm{val}(\Phi) = 1$.
- *(Soundness)* $\mathrm{val}(\Phi) < 1 - \varepsilon$.

By starting with Gap-ETH instead of ETH, there is no need to apply the PCP Theorem and hence a polylogarithmic loss in the size of the 3-SAT instance does not occur. As demonstrated in previous works, this allows one to improve the ratio in hardness of approximation results [19, 34, 33] and, more importantly, prove inapproximability results for some parameterized problems [10, 11, 16], which are not known to be hard to approximate under ETH. Specifically, for many parameterized problems, the reduction from the gap version of 3-SAT to the problem has size $2^{m'/f(k)}$ for some function f that grows to infinity with k, where m' is the number of clauses in the 3-CNF formula and k is the parameter of the problem. For simplicity, let us focus on the case where $f(k) = k$. If one wishes to derive a meaningful result starting form ETH, $2^{m'/k}$ must be subexponential in terms of m, the number of clauses in the original (no-gap) 3-CNF formula. This means that the term k must dominate the $\log^c m$ factor blow-up from the PCP Theorem. However, since FPT algorithms are allowed to have running time of the form $g(k)$ for any function g, we can pick g to be 2^{2^k}. In this case, the algorithm runs in $2^{\omega(m)}$ time and we cannot deduce anything regarding the algorithm. On the other hand, if we start from Gap-ETH, we can pick k to be a large constant independent of m, which indeed yields hardness of the form claimed in Theorem 2 and Corollary 4.

Finally, we remark that Gap-ETH would follow from ETH if a linear-size (constant-query) PCP exists. While constructing short PCPs has long been an active area of research [6, 8, 18, 36, 7], no such PCP is yet known. For a more in-depth discussion, please refer to [19].

Organization of the Paper. In the next section, we describe our reduction and give an overview of the proof. After defining additional notations in Section 3, we proceed to provide

[4] This bounded degree assumption can be assumed without loss of generality; see [34] for more details.

the soundness analysis of our construction in Section 4. In Section 5, we briefly discuss the setting of parameters that give the desired inapproximability results for 2-CSPs and DSN. Finally, we conclude our work with some discussions and open questions in Section 6.

2 Proof Overview

Like other (Gap-)ETH-hardness of approximation results, our proof is based on a subexponential time reduction from the gap version of 3-SAT to our problem of interest, 2-CSPs. Before we describe our reduction, let us define more notations for 2-CSPs and 3-SAT.

2-CSPs. For notational convenience, we will modify the definition of 2-CSPs slightly so that each variable is allowed to have different alphabets; this definition is clearly equivalent to the more common definition used above. Specifically, an instance Γ of 2-CSP now consists of (1) a constraint graph $G = (V, E)$, (2) for each vertex (or variable) $v \in V$, an alphabet set Σ_v, and, (3) for each edge $\{u, v\} \in E$, a constraint $C_{uv} \subseteq \Sigma_u \times \Sigma_v$. Additionally, to avoid confusion with 3-SAT, we refrain from using the word *assignment* for 2-CSPs and instead use *labeling*, i.e., a labeling of Γ is a tuple $\sigma = (\sigma_v)_{v \in V}$ such that $\sigma_v \in \Sigma_v$ for all $v \in V$. An edge $\{u, v\} \in E$ is said to be *satisfied* by a labeling σ if $(\sigma_u, \sigma_v) \in \Sigma_u \times \Sigma_v$. The value of a labeling σ, denoted by $\mathrm{val}(\sigma)$, is defined as the fraction of edges that it satisfies, i.e., $|\{\{u, v\} \in E \mid (\sigma_u, \sigma_v) \in C_{uv}\}|/|E|$. The goal of 2-CSPs is to find σ with maximum value; we denote the such optimal value by $\mathrm{val}(\Gamma)$, i.e., $\mathrm{val}(\Gamma) = \max_\sigma \mathrm{val}(\sigma)$.

3-SAT. An instance Φ of 3-SAT consists of a variable set X and a clause set $\mathcal{C}$ where each clause is a disjunction of at most three literals. For any assignment $\psi : \mathrm{X} \to \{0, 1\}$, $\mathrm{val}(\psi)$ denotes the fraction of clauses satisfied by ψ. The goal is to find an assignment ψ that satisfies as many clauses as possible; let $\mathrm{val}(\Phi) = \max_\psi \mathrm{val}(\psi)$ denote the fraction of clauses satisfied by such assignment. For each $C \in \mathcal{C}$, we use $\mathrm{var}(C)$ to denote the set of variables whose literals appear in C and, for each $S \subseteq \mathcal{C}$, we use $\mathrm{var}(S)$ to denote $\bigcup_{C \in S} \mathrm{var}(C)$.

Our Construction

Before we state our reduction, let us again reiterate the objective of our reduction. Given a 3-SAT instance $\Phi = (\mathrm{X}, \mathcal{C})$, we would like to produce a 2-CSP instance Γ_Φ such that

- (Completeness) If $\mathrm{val}(\Phi) = 1$, then $\mathrm{val}(\Gamma_\Phi) = 1$,
- (Soundness) If $\mathrm{val}(\Phi) < 1 - \varepsilon$, $\mathrm{val}(\Gamma_\Phi) < k^{o(1)}/k$ where k is number of variables of Γ_Φ,
- (Reduction Time) The time it takes to produce Γ_Φ should be $2^{o(m)}$ where $m = |\mathcal{C}|$,

where $\varepsilon > 0$ is some absolute constant.

Observe that, when plugging a reduction with these properties to Gap-ETH, we directly arrive at the claimed $k^{1-o(1)}$ inapproximability for 2-CSPs. However, for ETH, since we start with a decision version of 3-SAT without any gap, we have to first invoke the PCP theorem to produce an instance of the gap version of 3-SAT before we can apply our reduction. Since the shortest known PCP has a polylogarithmic blow-up in the size, the running time lower bound for gap 3-SAT will not be exponential anymore, rather it will be of the form $2^{\Omega(m/\mathrm{polylog} m)}$ instead. Hence, our reduction will need to produce Γ_Φ in $2^{o(m/\mathrm{polylog} m)}$ time. As we shall see below, this will also be possible with appropriate settings of parameters.

We now move on to state our reduction. In addition to a 3-CNF formula Φ, the reduction also takes in a collection S of subsets of clauses of Φ. For now, the readers should think of the subsets in S as random subsets of $\mathcal{C}$ where each element is included in each subset independently at random with probability α, which will be specified later. As we will see

below, we only need two simple properties that the subsets in S are "well-behaved" enough and we will later give a deterministic construction of such well-behaved subsets. With this in mind, our reduction can be formally described as follows.

▶ **Definition 8** (The Reduction)**.** Given a 3-CNF formula $\Phi = (\mathrm{X}, \mathcal{C})$ and a collection S of subsets of $\mathcal{C}$, we define a 2-CSP instance $\Gamma_{\Phi,\mathsf{S}} = (G = (V, E), \Sigma, \{C_{uv}\}_{\{u,v\}\in E})$ as follows:

- The graph G is the complete graph where the vertex set is S, i.e., $V = \mathsf{S}$ and $E = \binom{\mathsf{S}}{2}$.
- For each $S \in \mathsf{S}$, the alphabet set Σ_{S} is the set of all partial assignments to $\mathrm{var}(S)$ that satisfies every clause in S, i.e., $\Sigma_S = \{\psi_S : \mathrm{var}(S) \to \{0,1\} \mid \ \forall C \in S, \psi_S \text{ satisfies } C\}$.
- For every $S_1 \neq S_2 \in \mathsf{S}$, (ψ_{S_1}, ψ_{S_2}) is included in $C_{S_1S_2}$ if and only if there are consistent, i.e., $C_{S_1S_2} = \{(\psi_{S_1}, \psi_{S_2}) \in \Sigma_{S_1} \times \Sigma_{S_2} \mid \ \forall x \in \mathrm{var}(S_1) \cap \mathrm{var}(S_2), \psi_{S_1}(x) = \psi_{S_2}(x)\}$.

Let us now examine the properties of the reduction. The number of vertices in $\Gamma_{\Phi,\mathsf{S}}$ is $k = |\mathsf{S}|$. For this proof overview, α should be thought of as $1/\mathrm{polylog}(m)$ whereas k should be thought of as much larger than $\exp(1/\alpha)$ (e.g. $k = \exp(1/\alpha^2)$). For such value of k, all random sets in S have size $O(\alpha m)$ w.h.p., meaning that the reduction time is $2^{m/\mathrm{polylog}m}$.

Moreover, when Φ is satisfiable, it is not hard to see that $\mathrm{val}(\Gamma_{\Phi,\mathsf{S}}) = 1$; specifically, if $\psi : \mathrm{X} \to \{0,1\}$ is the assignment that satisfies every clause of Φ, then we can label each vertex $S \in \mathsf{S}$ of $\Gamma_{\Phi,\mathsf{S}}$ by $\psi|_{\mathrm{var}(S)}$, the restriction of ψ on $\mathrm{var}(S)$. Since ψ satisfies every clause, $\psi|_{\mathrm{var}(S)}$ satisfies all clauses in S and this is a valid labeling. Moreover, since these are restrictions of the same global assignment ψ, they are consistent, i.e., every edge is satisfied.

Hence, we are only left to show that, if $\mathrm{val}(\Phi) < 1 - \varepsilon$, then $\mathrm{val}(\Gamma_{\Phi,\mathsf{S}}) < k^{o(1)}/k$; this is indeed our main contribution. We will show this contrapositively: assuming that $\mathrm{val}(\Gamma_{\Phi,\mathsf{S}}) \geqslant k^{o(1)}/k$, we will "decode" back an assignment to Φ that satisfies at least $1 - \varepsilon$ fraction of clauses.

We remark that our task at hand can be viewed as agreement testing. Informally, in agreement testing, the input is a collection $\{f_T\}_T$ of local functions $f_T : T \to \{0,1\}$ where T is a subset of some universe $\mathcal{U}$ such that, for many pairs T_1 and T_2, f_{T_1} and f_{T_2} agree, i.e., $f_{T_1}(u) = f_{T_2}(u)$ for all $u \in T_1 \cap T_2$. An agreement theorem says that there must be a global function $f : \mathcal{U} \to \{0,1\}$ that coincides (exactly or approximately) with many of the local functions, and thus explains the pairwise "local" agreements. (See e.g. Section 1.1 [22] for a formal definition.) In our case, a labeling $\sigma = \{\sigma_S\}_{S\in\mathsf{S}}$ with high value is exactly a collection of functions $\sigma_S : S \to \{0,1\}$ such that, for many pairs of S_1 and S_2, σ_{S_1} and σ_{S_2} agrees. Our proof of soundness indeed recovers a global function $\psi : \mathrm{X} \to \{0,1\}$ that coincides with many of the local functions σ_S's and thus satisfies $1 - \varepsilon$ fraction of clauses of Φ.

A Simplified Proof: $k^{1/2-o(1)}$ Ratio Inapproximability

Before we describe how we can decode an assignment for Φ when $\mathrm{val}(\Gamma_{\Phi,\mathsf{S}}) \geqslant k^{o(1)}/k$, let us sketch the proof assuming a stronger assumption that $\mathrm{val}(\Gamma_{\Phi,\mathsf{S}}) \geqslant \Theta(1/\alpha)/k^{1/2}$. Since $1/\alpha = k^{o(1)}$, this already implies a $k^{1/2-o(1)}$ factor ETH-hardness of approximating 2-CSPs. In the next subsection, we will refine the arguments and arrive at the desired $k^{1-o(1)}$ factor.

Let D be a large constant to be chosen later. Recall that $\mathrm{val}(\Gamma_{\Phi,\mathsf{S}}) \geqslant (D/\alpha)/k^{1/2}$ implies that there is a labeling $\sigma = \{\sigma_S\}_{S\in\mathsf{S}}$ that satisfies $(D/\alpha)/k^{1/2} \cdot \binom{k}{2} \geqslant \left(\frac{D}{4\alpha}\right) k^{3/2}$ edges.

Let us consider the *consistency graph* of $\Gamma_{\Phi,\mathsf{S}}$ with respect to σ. This is the graph G^σ whose vertex set is S and there is an edge between S_1 and S_2 iff σ_{S_1} and σ_{S_2} are consistent. Note that the number of edges in G^σ is equal to the number of edges satisfied by σ.

Previous works on agreement testers exploit particular structures of the consistency graph to decode a global function. One such property that is relevant to our proof is the notion of *almost transitivity* defined by Raz and Safra in the analysis of their test [39]. More specifically,

a graph $G = (V, E)$ is said to be q-transitive for some $q > 0$ if, for every non-edge $\{u, v\}$ (i.e. $\{u, v\} \in \binom{V}{2} \notin E$), u and v can share at most q common neighbors[5]. Raz and Safra showed that their consistency graph is $(k^{1-\Omega(1)})$-transitive where k denote the number of vertices of the graph. They then proved a generic theorem regarding $(k^{1-\Omega(1)})$-transitive graphs: its vertex set can be partitioned so that the subgraph induced by each partition is a clique and the number of edges between different partitions is small. Since a sufficiently large clique corresponds to a global function in their setting, they immediately arrive at their result.

Observe that, in our setting, a large clique also corresponds to an assignment that satisfies almost all clauses of Φ. In particular, suppose that there exists $\mathsf{S}' \subseteq \mathsf{S}$ of size sufficiently large size such that S induces a clique in G^σ. Since σ_S are perfectly consistent among all $S \in \mathsf{S}'$, these local functions induce a function $\psi : \mathrm{var}(\bigcup_{S\in\mathsf{S}'} S) \to \{0, 1\}$ that satisfies all clauses in $\bigcup_{S\in\mathsf{S}'} S$. If S is larger than $\Omega(1/(\varepsilon\alpha))$, then, with high probability, $\bigcup_{S\in\mathsf{S}'} S$ contains all but ε fraction of clauses, which means that ψ satisfies $1 - \varepsilon$ fraction of clauses as desired. Hence, if we could show that our consistency graph G^σ is $(k^{1-\Omega(1)})$-transitive, then we could use the same argument as Raz and Safra's to deduce our desired result. Alas, our graph G^σ does not necessarily satisfy this transitivity property; for instance, consider any two sets $S_1, S_2 \in \mathsf{S}$ and let $\sigma_{S_1}, \sigma_{S_2}$ be such that they disagree on only one variable, i.e., there is a unique $x \in S_1 \cap S_2$ such that $\sigma_{S_1}(x) \neq \sigma_{S_2}(x)$. It is possible that, for every $S \in \mathsf{S}$ that does not contain x, σ_S agrees with both σ_{S_1} and σ_{S_2}; in other words, every such S can be a common neighbor of S_1 and S_2. Since each variable x appears roughly in only $\Theta(\alpha)$ fraction of the sets, there can be as many as $(1 - \Theta(\alpha))k = (1 - o(1))k$ common neighbors of S_1 and S_2 even when there is no edge between S_1 and S_2!

Fortunately for us, a weaker statement holds. If σ_{S_1} and σ_{S_2} disagree on ζn variables (instead of just one variable as above) where n denotes the number of variables in the 3-CNF formula, then we say that they *strongly* disagree. In this case, S_1 and S_2 can have at most $O(\ln(1/\zeta)/\alpha)$ common neighbors in G^σ. Here ζ should be thought of as α^2 times a small constant which will be specified later. To see why this statement holds, observe that, since every $S \in \mathsf{S}$ is a random subset that includes each clause $C \in \mathcal{C}$ with probability α, Chernoff bound implies that, for every subcollection $\widetilde{\mathsf{S}} \subseteq \mathsf{S}$ of size $\Omega(\ln(1/\zeta)/\alpha)$, $\bigcup_{S\in\widetilde{\mathsf{S}}} S$ contains all but $O(\zeta)$ fraction of clauses. Let $\widetilde{\mathsf{S}}_{S_1,S_2} \subseteq \mathsf{S}$ denote the set of common neighbors of S_1 and S_2. It is easy to see that S_1 and S_2 can only disagree on variables that do not appear in $\mathrm{var}(\bigcup_{S\in\widetilde{\mathsf{S}}_{S_1,S_2}} S)$. If $\widetilde{\mathsf{S}}_{S_1,S_2}$ is of size $\Omega(\ln(1/\zeta)/\alpha)$, then $\bigcup_{S\in\widetilde{\mathsf{S}}_{S_1,S_2}} S$ contains all but $O(\zeta)$ fraction of clauses. Hence, assuming that each variable appears in bounded number of clauses, $\mathrm{var}(\bigcup_{S\in\widetilde{\mathsf{S}}_{S_1,S_2}} S)$ also contains all but $O(\zeta)$ fraction of variables. This means that S_1 and S_2 disagrees only on $O(\zeta)$ fraction of variables. By selecting the constant appropriately inside $O(\cdot)$, we arrive at the claim statement.

In other words, while the transitive property does not hold for every edge, it holds for the edges $\{S_1, S_2\}$ where σ_{S_1} and σ_{S_2} strongly disagree. This motives us to define a two-level consistency graph, where the edges with strong disagreement are referred to as *red* edges whereas the original edges in G^σ are now referred to as *blue* edges, as formalized below.

▶ **Definition 9** (Two-Level Consistency Graph)**.** A red/blue graph is an undirected graph $G = (V, E = E_r \cup E_b)$ where its edge set E is partitioned into two sets E_r, the set of red edges, and E_b, the set of blue edges. We use prefixes "blue-" and "red-" to refer to quantities of (V, E_b) and (V, E_r) respectively. (E.g. u is a blue-neighbor of v if $\{u, v\} \in E_b$).

[5] In [39], the parameter q denotes the *fraction* of vertices that are neighbors of both u and v rather than the *number* of such vertices. However, we use the latter notion as it is more convenient for us.

Given a labeling σ of $\Gamma_{\Phi,\mathsf{S}}$ and a real number $0 < \zeta < 1$, the two-level consistency graph $G^{\sigma,\zeta} = (V^{\sigma,\zeta}, E_r^{\sigma,\zeta} \cup E_b^{\sigma,\zeta})$ is a red/blue graph defined as follows.

- The vertex set $V^{\sigma,\zeta}$ is simply S.
- The blue edges are the pairs $\{S_1, S_2\}$ satisfied by σ, i.e., $E_b = \{\{S_1, S_2\} \in \binom{\mathsf{S}}{2} \mid \mathrm{disagr}(\sigma_{S_1}, \sigma_{S_2}) = 0\}$.
- The red edges are the pairs $\{S_1, S_2\}$ whose the assignments to the two endpoints disagree on more than ζn variables, i.e., $E_r = \{\{S_1, S_2\} \in \binom{\mathsf{S}}{2} \mid \mathrm{disagr}(\sigma_{S_1}, \sigma_{S_2}) > \zeta n\}$.

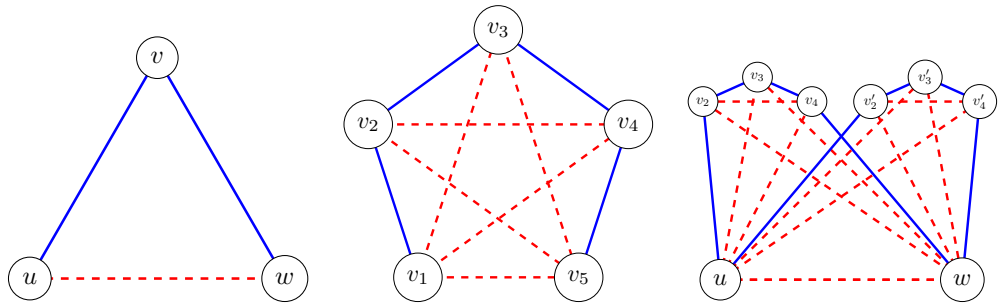
Here $\mathrm{disagr}(\sigma_{S_1}, \sigma_{S_2})$ denotes the number of variables that $\sigma_{S_1}, \sigma_{S_2}$ disagree on, i.e., $\mathrm{disagr}(\sigma_{S_1}, \sigma_{S_2}) = |\{x \in \mathrm{var}(S_1) \cap \mathrm{var}(S_2) \mid \sigma_{S_1}(x) \neq \sigma_{S_2}(x)\}|$.

Note that, when $\mathrm{disagr}(\sigma_{S_1}, \sigma_{S_2}) \in [1, \zeta n]$, S_1, S_2 constitute neither a blue nor a red edge.

Now, the transitivity property above can be stated as follows: for every red edge $\{S_1, S_2\}$ of $G^{\sigma,\zeta}$, there are at most $O(\ln(1/\zeta)/\alpha)$ different S's such that both $\{S, S_1\}$ and $\{S, S_2\}$ are blue edges. For brevity, let us call any red/blue graph $G = (V, E_r \cup E_b)$ *q-red/blue-transitive* if, for every red edge $\{u, v\} \in E_r$, u and v have at most q common blue-neighbors. We will now argue that in any q-red/blue-transitive of average blue-degree d, there exists a subset $U \subseteq V$ of size $\Omega(d)$ such that, only $O(qk/d^2)$ fraction of pairs of vertices in U form red edges.

Before we prove this, let us state why this is useful for decoding a good assignment for the 3-CNF formula Φ. Observe that such a subset U of vertices in the two-level consistency graph translates to a subcollection $\mathsf{S}' \subseteq \mathsf{S}$ such that, for all but $O(qk/d^2)$ fraction of pairs of sets $S_1, S_2 \subseteq \mathsf{S}'$, $\{S_1, S_2\}$ does not form a red edge, meaning that σ_{S_1} and σ_{S_2} disagrees on at most ζn variables. In other words, S' is similar to a clique in the (not two-level) consistency graph, except that (1) $O(qk/d^2)$ fraction of pairs $\{S_1, S_2\}$ are allowed to disagree on as many variables as they like, and (2) even for the rest of pairs, the guarantee now is that they agree on all but at most ζn variables, instead of total agreement as in the previous case of clique. Fortunately, when S' satisfies a certain uniformity condition (which random subsets satisfy w.h.p.), this still suffices to find an assignment to Φ that satisfies $1 - O(qk/d^2) - O(\zeta/\alpha^2)$ fraction of the clauses. One way construct a good assignment is to simply assign each variable $x \in \mathrm{X}$ according to the majority of $\sigma_S(x)$ for all $S \in \mathsf{S}'$ such that $x \in \mathrm{var}(S)$. Our actual proof proceeds slightly differently for technical reasons. Note that in our case $q = O(\ln(1/\zeta)/\alpha)$ and $d = \Omega(Dk^{1/2}/\alpha)$; if we pick $\zeta \ll \varepsilon\alpha^2$ and $D \gg 1/\sqrt{\varepsilon}$, we indeed get an assignment that satisfies $1 - \varepsilon$ fraction of clauses.

We now move on to sketch how one can find such a clique-like subgraph. For simplicity, let us assume that every vertex has the same blue-degree (i.e. (V, E_b) is d-regular). Let us count the number of *red-blue-blue triangle* (or *rbb triangle*), which is a 3-tuple (u, v, w) of vertices in V such that $\{u, v\}, \{v, w\}$ are blue edges whereas $\{u, w\}$ is a red edge. An illustration of a rbb triangle can be found in Figure 1a. The red/blue transitivity can be used to bound the number of rbb triangles as follows. For each $(u^*, w^*) \in V^2$, since the graph is q-red/blue-transitive there are at most q rbb triangle with $u = u^*$ and $w = w^*$. Hence, in total, there can be at most qk^2 rbb triangles. As a result, there exists $v^* \in V$ such that the number of rbb triangles (u, v, w) such that $v = v^*$ is at most qk. Let us now consider the set $U = N_b(v^*)$ that consists of all blue-neighbors of v^*. There can be at most qk red edges with both endpoints in $N_b(v^*)$ because each such edge corresponds to a rbb triangle with $v = v^*$. From our assumption that every vertex has blue degree d, we indeed have that $|U| = d$ and that the fraction of pairs of vertices in U that are linked by red edges is $O(qk/d^2)$ as desired. This completes our overview for $k^{1/2-o(1)}$ factor inapproximability result for 2-CSPs.



(a) a red-blue-blue triangle **(b)** a red-filled 4-walk **(c)** disjoint red-filled 4-walks

Figure 1 Illustrations of red-filled walks. Figures 1a and 1b demonstrate a red-filled 2 walk (rbb triangle) and a red-filled 4-walk respectively. Figure 1c shows two disjoint red-filled 4-walks.

Towards Nearly Linear Ratio Inapproximability

To improve our inapproximability ratio from $k^{1/2-o(1)}$ to $k^{1-o(1)}$, we need to first understand why the approach above fails to work beyond the $k^{1/2}$ ratio regime. To do so, note that the above proof sketch can be summarized into three main steps as follows:

(1) Show that the two-level consistency graph $G^{\sigma,\zeta}$ is q-red/blue-transitive for some $q = k^{o(1)}$.
(2) Use red/blue transitivity to find a large subgraph of $G^{\sigma,\zeta}$ with few induced red edges.
(3) Decode a good assignment to Φ from such "clique-like" subgraph.

The reason that we need $d \geqslant k^{1/2}$ lies in Step 2. Although not stated as such above, our argument in this step can be described as follows. Consider all length-2 blue-walks, i.e., all $(u, v, w) \in V^3$ such that $\{u, v\}$ and $\{v, w\}$ are both blue edges. Using the red/blue transitivity of the graph, we argue that, for almost of all these walks, $\{u, w\}$ is not a red edge (i.e. (u, v, w) is not a rbb triangle), which then allows us to find the "clique-like" subgraph. For this argument to work, we need the number of length-2 blue-walks to exceed the number of rbb triangles. The former is kd^2 whereas the latter is bounded by k^2q in q-red/blue-transitive graphs. This means that we need $kd^2 \geqslant k^2q$, which implies that $d \geqslant k^{1/2}$.

To overcome this limitation, we will instead consider length-ℓ blue-walks for $\ell > 2$ and define a "rbb-triangle-like" structure on these walks. Our goal is again to show that this structure appears rarely in random length-ℓ blue-walks and then use this to find a subgraph that allows us to decode a good assignment for Φ. Observe that the number of length-ℓ blue walks is kd^ℓ. We hope that the number of "rbb-triangle-like" structures is still small; in particular, we will still get a similar bound $k^{2+o(1)}$ for such generalized structure, similar to our previous bound for the red-blue-blue triangles. When this is the case, we need $kd^\ell \geqslant k^{2+o(1)}$, meaning that when $\ell = \omega(1)$ it suffices to select $d = k^{o(1)}$, which yields $k^{1-o(1)}$ factor inapproximability as desired. To facilitate our discussion, let us define notations for ℓ-walks here.

▶ **Definition 10** (ℓ-Walks)**.** For any red/blue graph $G = (V, E_r \cup E_b)$ and any integer $\ell \geqslant 2$, an *ℓ-blue-walk*, abbreviated as an *ℓ-walk*, in G is an $(\ell+1)$-tuple of vertices $(v_1, v_2, \ldots, v_{\ell+1}) \in V^{\ell+1}$ such that every pair of consecutive vertices is joined by a blue edge, i.e., $\{v_i, v_{i+1}\} \in E_b$ for every $i \in [\ell]$. We use $\mathcal{W}_\ell^G$ to denote the set of all ℓ-walks in G.

The structure we will consider is a natural generalization of rbb triangles to an ℓ-walk in which every pair of non-consecutive vertices of the walk must be joined by a red edge. We call such a walk a *red-filled ℓ-walk* (see Figure 1b):

▶ **Definition 11** (Red-Filled ℓ-Walks)**.** For any red/blue graph $G = (V, E_r \cup E_b)$, a *red-filled ℓ-walk* is an ℓ-walk $(v_1, v_2, \ldots, v_{\ell+1})$ such that every pair of non-consecutive vertices is joined by a red edge, i.e., $\{v_i, v_j\} \in E_r$ for every $i, j \in [\ell+1]$ such that $j > i+1$. Let $\widehat{\mathcal{W}}_\ell^G$ denote the set of all red-filled ℓ-walks in G. Moreover, for every $u, v \in V$, let $\widehat{\mathcal{W}}_\ell^G(u, v)$ denote the set of all red-filled ℓ-walks from u to v, i.e., $\mathcal{W}_\ell^G(u, v) = \{(v_1, \ldots, v_{\ell+1}) \in \widehat{\mathcal{W}}_\ell^G \mid v_1 = u \wedge v_{\ell+1} = v\}$.

As mentioned earlier, we will need a generalized transitivity property for our new structure. This can be defined analogously to q-red/blue transitivity as follows.

▶ **Definition 12** ((q, ℓ)-Red/Blue Transitivity)**.** For any $q, \ell \in \mathbb{N}$, a red/blue graph $G = (V, E_r \cup E_b)$ is said to be (q, ℓ)*-red/blue-transitive* if, for every pair of $u, v \in V$ that is joined by a red edge, there are at most q red-filled ℓ-walks from u to v, i.e., $|\widehat{\mathcal{W}}_\ell^G(u, v)| \leqslant q$.

Similar to before, we can argue that, when S consists of random subsets where each element is included in a subset w.p. α, the two-level agreement graph is (q, ℓ)-red/blue transitive for some parameter q that is a function of only α and ℓ. When $1/\alpha$ and ℓ are both small enough in terms of k, q can made to be $k^{o(1)}$. The details of the proof can be found in Section 4.1.

Once this is proved, it is not hard (using a similar argument as before) to show that, when $d \gg (kq)^{1/\ell}$, most ℓ-walks are not red-filled, i.e., $|\mathcal{W}_\ell^G| \gg |\widehat{\mathcal{W}}_\ell^G|$. Even with this, it is still unclear how we can get back a "clique-like" subgraph; in the case of $\ell = 2$ above, this implies that a blue-neighborhood induces few red edges, but the argument does not seem to generalize to larger ℓ. Fortunately, it is still quite easy to find a large subgraph that a non-trivial fraction of pairs of vertices do *not* form red edges; specifically, we will find two subsets $U_1, U_2 \subseteq V$ each of size d such that for at least $1/\ell^2$ fraction of $(u_1, u_2) \in U_1 \times U_2$, $\{u_1, u_2\}$ is not a red edge. To find such sets, observe that, if $|\mathcal{W}_\ell^G| \geqslant 2|\widehat{\mathcal{W}}_\ell^G|$, then for a random $(v_1, \ldots, v_{\ell+1}) \in \mathcal{W}_\ell^G$ the probability that there exists non-consecutive vertex v_i, v_j in the walk that are joined by a red edge is at least $1/2$. Since there are less than $\ell^2/2$ such i, j, union bound implies that there must be non-consecutive i^*, j^* such that the probability that v_{i^*}, v_{j^*} are not joined by a red edge is at least $1/\ell^2$. Let us assume without loss of generality that $i^* < j^*$; since they are not consecutive, we have $i^* + 1 < j^*$.

Let us consider v_{i^*+1}, v_{j^*-1}. There must be u^* and w^* such that, conditioning on $v_{i^*+1} = u^*$ and $v_{j^*+1} = w^*$, the probability that $\{v_{i^*}, v_{j^*}\} \notin E_r$ is at least $1/\ell^2$. However, this conditional probability is exactly equal to the fraction of $(u_1, u_2) \in N_b(u^*) \times N_b(w^*)$ such that u_1, u_2 are not joined by a red edge. (Recall that $N_b(v)$ is the set of all blue-neighbors of v.) As a result, $U_1 = N_b(u^*)$ and $U_2 = N_b(w^*)$ are the sets with desired property.

We are still not done yet since we have to use these sets to decode back a good assignment for Φ. This is still not obvious: the guarantee we have for our sets U_1, U_2 are rather weak since we only know that at least $1/\ell^2$ of the pairs of vertices from the two sets do not form red edges. This is in contrast to the $\ell = 2$ case where we have a subgraph such that almost all induced edges are *not* red. Fortunately, there is a well-known fact in combinatorics called the Kővári-Sós-Turán Theorem [32] which roughly states that every bipartite graph that is not too sparse has a reasonably large biclique (a complete bipartite subgraph). We apply this theorem on the bipartite graph between U_1 and U_2 where there is an edge between $u_1 \in U_1$ and $u_2 \in U_2$ iff $\{u_1, u_2\}$ is not a red edge. This gives us $V_1 \subseteq U_1$, $V_2 \subseteq U_2$ of reasonably large sizes such that for all $(u_1, u_2) \in V_1 \times V_2$, u_1 and u_2 are not joined by a red edge.

Once we have such a "non-red biclique", we can decode a good assignment of Φ by taking the majority assignment on one side of the biclique. A simple counting argument again shows that, when V_1 and V_2 are "sufficiently uniform", this majority assignment cannot violate too many clauses of Φ. This wraps up our proof overview.

3 Preliminaries

We next define two properties of collections of subsets, which will be useful in our analysis. First, recall that, in our proof overview for the weaker $k^{1/2-o(1)}$ factor hardness, we need the following to show the red/blue transitivity of the consistency graph: for any r subsets from the collection, their union must contain almost all clauses. Here r is a positive integer that effects the red/blue transitivity parameter. Collections with this property are sometimes called *dispersers*. For walks with larger lengths, we need a stronger property that any union of r intersections of ℓ subsets are large. We call such collections *intersection dispersers*:

▶ **Definition 13** (Intersection Disperser). Given a universe $\mathcal{U}$, a collection S of subsets of $\mathcal{U}$ is an (r, ℓ, η)-*intersection disperser* if, for any r disjoint subcollections $\mathsf{S}^1, \ldots, \mathsf{S}^r \subseteq \mathsf{S}$ each of size at most ℓ, we have $\left|\bigcup_{i=1}^{r} \left(\bigcap_{S \in \mathsf{S}^i} S\right)\right| \geqslant (1-\eta)|\mathcal{U}|$.

Another property we need is that any sufficiently large subcollection $\widetilde{\mathsf{S}}$ of S is "sufficiently uniform". This is used when we decode a good assignment from a non-red biclique. More specifically, the uniformity condition requires that almost all clauses appear in not too small number of subsets in $\widetilde{\mathsf{S}}$, as formalized below.

▶ **Definition 14** (Uniformity). For a universe $\mathcal{U}$, a collection $\widetilde{\mathsf{S}}$ of subsets of $\mathcal{U}$ is (γ, μ)-uniform if, for at least $(1-\mu)$ fraction of elements $u \in \mathcal{U}$, u appears in at least γ fraction of the subsets in $\widetilde{\mathsf{S}}$. In other words, $\widetilde{\mathsf{S}}$ is (γ, μ)-uniform iff $|\{u \in \mathcal{U} \mid |\{S \in \widetilde{\mathsf{S}} \mid u \in S\}| \geqslant \gamma|\widetilde{\mathsf{S}}|\}| \geqslant (1-\mu)|\mathcal{U}|$.

Using standard concentration bounds, it is not hard to show that, when m is sufficiently large, a collection of random subsets where each element is included in each subset independently with probability α is an $(O(\alpha^\ell), \ell, O(1))$-disperser and every subcollection of size $\Omega(1/\alpha)$ is $(\alpha, O(1))$-uniform. The exact parameter dependencies are shown in the lemma below.

▶ **Lemma 15** (Deterministic Construction of Well-Behaved Set). *For any $0 < \alpha, \mu, \eta < 1$ and any $k, \ell \in \mathbb{N}$, let m_0 be $1000(\log k \log(1/\mu)/(\alpha\mu^2) + \ell \log(1/\eta) \log k/(\alpha^\ell \eta) + 1/\alpha + 1)$. For any integer $m \geqslant m_0$ and any m-element universe $\mathcal{U}$, there exists a collection S of subsets of $\mathcal{U}$ with the following properties.*

- *(Size) Every subset in S has size at most $2\alpha m$.*
- *(Intersection Disperser) S is a $(\lceil \ln(2/\eta)/\alpha^\ell \rceil, \ell, \eta)$-disperser.*
- *(Uniformity) Any subcollection $\widetilde{\mathsf{S}} \subseteq \mathsf{S}$ of size $\lceil 8 \ln(2/\mu)/\alpha \rceil$ is $(\alpha/2, \mu)$-uniform.*

Moreover, such a collection S can be deterministically constructed in time $\mathrm{poly}(m) 2^{O((m_0)^3)}$.

The deterministic construction is via a standard technique of using random variables with limited independence instead of total independence; we defer the full proof of Lemma 15 to the full version of the paper.

4 Soundness Analysis

Let us now turn our focus back to the soundness analysis, which is our main technical contribution. As stated in the proof overview, our main goal is to show that, if a 3-CNF Φ has small value, then, for any well-behaved collection S of subsets of clauses of Φ, the value of $\Gamma_{\Phi,\mathsf{S}}$ must be small. More precisely, the main theorem of this section is stated below.

▶ **Theorem 16.** *For any $\Delta \in \mathbb{N}$, let Φ be any 3-CNF formula with variable set X and clause set $\mathcal{C}$ such that each variable appears in at most Δ clauses. Moreover, for any $0 < \eta, \zeta, \gamma, \mu < 1$ and $r, \ell, k, h \in \mathbb{N}$ such that $\ell \geqslant 2$ and $h \leqslant \log k/(\ell \log(4\ell^2))$, let S be any collection of k*

subsets of $\mathcal{C}$ *such that* S *is* $(r, \ell, \zeta/(3\Delta))$*-intersection disperser and every subcollection* $\widetilde{\mathsf{S}} \subseteq \mathsf{S}$ *of size* h *is* (γ, μ)*-uniform (with respect to the universe* $\mathcal{C}$*). If* $\mathrm{val}(\Phi) < 1 - 2\mu - 6\Delta\zeta/\gamma^2$*, then* $\mathrm{val}(\Gamma_{\Phi,\mathsf{S}}) < \frac{32k^{1/\ell}(r\ell)^2}{k}$.

To prove this theorem, we follow the general outline as stated in the proof overview. In particular, the proof contains three main steps, as elaborated below.

(1) First, we will show that when S is an intersection disperser with appropriate parameters, the two-level consistency graph satisfies red/blue transitivity with certain parameters.

(2) Next, we will argue that, for any red/blue transitive graphs that contains sufficiently many blue edges, we can find a non-red biclique of large size; recall that non-red biclique is two subsets V_1, V_2 of vertices such that there is no red-edge between them.

(3) Finally, we show that, if we can find a large non-red biclique in the two-level consistency graph such that the two subcollections corresponding to each side of the biclique are sufficiently uniform, then we can decode a good assignment to our 3-CNF formula Φ.

Each of the next three subsections is dedicated to each part of the proof. The main lemmas from these subsections (Lemmas 17, 20 and 23) together imply Theorem 16.

Unless stated otherwise, we note that, all results in this section hold for any parameters $\Delta, \ell, k \in \mathbb{N}$ such that $\ell \geqslant 2$, any $0 < \eta, \zeta, \gamma, \mu < 1$ and any 3-CNF formula Φ such that each variables appears in at most Δ clauses. To avoid notational cumbersomeness, we will leave these quantifiers out of the lemma statements. Moreover, throughout the section, we use m and n to denote $|\mathcal{C}|$ and $|\mathrm{X}|$ respectively. To avoid degeneracy cases, we will also assume without loss of generality that each variable appears in at least one clause.

4.1 Red/Blue-Transitivity of Two-Level Consistency Graph

The first step in our proof is to show that the two-level consistency graph $G^{\sigma,\zeta}$ is red/blue-transitive, assuming that S is an intersection disperser, as formalized below.

▶ **Lemma 17.** *If* S *is an* $(r, \ell, \zeta/(3\Delta))$*-intersection disperser, then, for any labeling* σ *of* $\Gamma_{\Phi,\mathsf{S}}$*,* $G^{\sigma,\zeta}$ *is* $((r\ell)^{2(\ell-1)}, \ell)$*-red/blue-transitive.*

In other words, we would like to show that, for every $S_1, S_2 \in \mathsf{S}$ that are joined by a red edge in $G^{\sigma,\zeta}$, there are at most $(r\ell)^{2(\ell-1)}$ red-filled ℓ-walks from S_1 to S_2. The intersection disperser does not immediately imply such a bound, due to the requirement in the definition that the subcollections are disjoint. Rather, it only directly implies a bound on number of *disjoint* ℓ-walks from S_1 to S_2, where two ℓ walks from S_1 to S_2, $(T_1 = S_1, \ldots, T_{\ell+1} = S_2), (T'_1 = S_1, \ldots, T'_{\ell+1} = S_2) \in \mathcal{W}_\ell^{G^{\sigma,\zeta}}(S_1, S_2)$, are said to be *disjoint* if they do not share any vertex except the starting and ending vertices, i.e., $\{T_2, \ldots, T_\ell\} \cap \{T'_2, \ldots, T'_\ell\} = \emptyset$. Multiple walks are said to be disjoint if they are mutually disjoint. The following claim is immediate from the definition of intersection dispersers; its proof is omitted here.

▶ **Claim 18.** *If* S *is an* $(r, \ell, \zeta/(3\Delta))$*-intersection disperser, then, for any labeling* σ*,* $2 \leqslant p \leqslant \ell$ *and* $\{S_1, S_2\} \in E_r^{\sigma,\zeta}$*, there are less than* r *disjoint* p*-walks from* S_1 *to* S_2 *in* $G^{\sigma,\zeta}$*.*

Since all 2-walks from S_1 to S_2 are disjoint, the above claim immediately gives a bound on the number of red-filled 2-walks from S_1 to S_2. To bound the number of red-filled walks of larger lengths, we will use induction on the length of the walks. Suppose that we have bounded the number of red-filled i-walks sharing starting and ending vertices for $i \leqslant z - 1$. The key idea in the proof is that we can use this inductive hypothesis to show that, for any $S_1, S_2, S \in \mathsf{S}$, few z-walks from S_1 to S_2 contain S. Here we say that a z-walk $(T_1 = S_1, \ldots, T_z = S_2)$ from S_1 to S_2 *contains* S if $S \in \{T_2, \ldots, T_z\}$. In other words, each

z-walk from S_1 to S_2 is not disjoint with only few other z-walks from S_1 to S_2. This allows us to show that, if there are too many z-walks, then there must also be many disjoint z-walks as well, which would violate Claim 18. A formal proof of Lemma 17 based on this intuition is given below.

Proof of Lemma 17. For every integer i such that $2 \leqslant i \leqslant \ell$, let $P(i)$ denote the following: for every $S_1, S_2 \in \mathsf{S}$, $|\widehat{\mathcal{W}}_i^{G^{\sigma,\zeta}}(S_1, S_2)| \leqslant (ri)^{2(i-1)}$. For convenient, let $B_i = (ri)^{2(i-1)}$.

$P(2)$ follows from Claim 18. Now, suppose that, for some integer z such that $3 \leqslant z \leqslant \ell$, $P(2), \ldots, P(z-1)$ are true. To prove $P(z)$, let us first show that, for any fixed starting and ending vertices, any vertex cannot appear in too many red-filled z-walks:

▶ **Claim 19.** *For every $S_1, S_2, S \in \mathsf{S}$, the number of red-filled z-walks from S_1 to S_2 containing S in $G^{\sigma,\zeta}$ is at most $B_z/(zr)$.*

Proof. First, observe that the number of red-filled z-walks from S_1 to S_2 containing S is at most the sum over all positions $2 \leqslant j \leqslant z$ of the number of z-walks from S_1 to S_2 such that the j-th vertex in the walk is S, i.e., $\sum_{j=2}^{z} |\{(T_1, \ldots, T_{z+1}) \in \widehat{\mathcal{W}}_z^{G^{\sigma,\zeta}}(S_1, S_2) \mid T_j = S\}|$.

Now, for each $2 \leqslant j \leqslant z$, to bound the number of red-filled z-walks from S_1 to S_2 whose j-th vertex is S, let us consider the following three cases based on the value of j:

1. $3 \leqslant j \leqslant z-1$. Observe that, for any such walk $(T_1 = S_1, T_2, \ldots, T_j = S, \ldots, T_z, T_{z+1} = S_2)$, the subwalk $(T_1 = S_1, \ldots, T_j = S)$ and $(T_j = S, \ldots, T_{z+1} = S_2)$ must be red-filled walks as well. Since the numbers of red-filled $(j-1)$-walks from S_1 to S and red-filled $(z+1-j)$-walks from S to S_2 are bounded by B_{j-1} and B_{z+1-j} respectively (from the inductive hypothesis), there are at most B_{j-1} choices of $(T_1 = S_1, \ldots, T_j = S)$ and B_{z+1-j} choices of $(T_j = S, \ldots, T_{z-1}, T_z = S_2)$. Hence, there are at most $B_{j-1}B_{z+1-j}$ red-filled z-walks from S_1 to S_2 whose j-th vertex is S.
2. $j = 2$. In this case, the subwalk $(T_2, \ldots, T_{z+1})$ must be a red-filled $(z-1)$-walk from S to S_2. Hence, the number of red-filled z-walks from S_1 to S_2 where $T_j = S$ is at most B_{z-1}.
3. $j = z$. Similar to the previous case, we also have the bound of B_{z-1}.

Summing the above bounds over all j's, the number of red-filled z-walks from S_1 to S_2 containing S is at most $\sum_{j=2}^{z} B_{j-1}B_{z+1-j} \leqslant \sum_{j=2}^{z}(rz)^{2(z-2)} \leqslant B_z/(zr)$ as desired. ◀

Having proved the above claim, it is now easy to show that $P(z)$ is true. Suppose for the sake of contradiction that there exists $S_1, S_2 \in \mathsf{S}$ such that $|\widehat{\mathcal{W}}_z^{G^{\sigma,\zeta}}(S_1, S_2)| > B_z$. Consider the following procedure of selecting disjoint walks from $\widehat{\mathcal{W}}_z^{G^{\sigma,\zeta}}(S_1, S_2)$. First, initialize $U = \widehat{\mathcal{W}}_z^{G^{\sigma,\zeta}}(S_1, S_2)$ and repeat the following process as long as $U \neq \emptyset$: select any $(T_1, \ldots, T_{z+1}) \in U$ and remove every $(T'_1, \ldots, T'_{z+1})$ that is not disjoint with $(T_1, \ldots, T_{z+1})$ from U. Observe that, each time a walk $(T_1, \ldots, T_{z+1})$ is selected, the number of walks removed from U is at most B_z/r; this is because each removed walk must contain at least one of $T_2, \ldots, T_z$, but, from the above claim, each of these vertices are contained in at most $B_z/(zr)$ walks. Since we start with more than B_z walks, at least r disjoint walks are picked, which, due to Claim 18, is a contradiction. Thus, $P(z)$ is true as desired. ◀

4.2 Finding Non-Red Biclique in Red/Blue-Transitive Graph

In the second step of our proof, we will show that any (q, ℓ)-red/blue transitive graph with sufficiently many edges must contain a sufficiently large non-red biclique, as stated below.

▶ **Lemma 20.** *For every $k, q, \ell, d \in \mathbb{N}$ such that $d \geqslant \max\{(2qk)^{1/\ell}, 2\ell^2\}$ and every k-vertex (q,ℓ)-red/blue-transitive graph $G = (V, E_r \cup E_b)$ such that $|E_b| \geqslant 2kd$, there exist $V_1, V_2 \subseteq V$ each of size at least $\log d / \log(4\ell^2) - 1$ such that, for every $u \in V_1$ and $v \in V_2$, $\{u, v\} \notin E_r$.*

As stated in the outline, we prove Lemma 20 by first finding subsets of vertices $U_1, U_2 \subseteq V$ such that for $1/\ell^2$ fraction of $(u_1, u_2) \in U_1 \times U_2$, u_1 and u_2 are not joined by a red edge and then use the Kővári-Sós-Turán Theorem to find the desired non-red biclique. Specifically, to prove Lemma 20, we show the following:

▶ **Lemma 21.** *For every $k, q, \ell, d \in \mathbb{N}$ such that $d \geqslant (2qk)^{1/\ell}$ and every k-vertex (q,ℓ)-red/blue-transitive graph $G = (V, E_r \cup E_b)$ such that $|E_b| \geqslant 2kd$, there exists subsets of vertices $U_1, U_2 \subseteq V$ each of size at least d such that $|\{(u,v) \in U_1 \times U_2 \mid \{u,v\} \notin E_r\}| \geqslant |U_1||U_2|/\ell^2$.*

The Kővári-Sós-Turán Theorem can be stated as follows.

▶ **Theorem 22** (Kővári-Sós-Turán (KST) Theorem [32])**.** *For every $t, M, N \in \mathbb{N}$ such that $t \leqslant \min\{M, N\}$, any $K_{t,t}$-free bipartite graph with N vertices one side and M vertices on the other contain at most $(t-1)^{1/t}(N - t + 1)M^{1-1/t} + (t-1)M$ edges.*

A simple calculation shows that Lemma 20 follows from Lemma 21 and the KST Theorem. We now move on to the proof of Lemma 21, which is exactly as sketched earlier in Subsection 2.

Proof of Lemma 21. We start by preprocessing the graph so that every vertex has blue-degree at least d. In particular, as long as there exists a vertex v whose blue-degree is at most d, we remove v from G. Let $G' = (V', E'_r \cup E'_b)$ be the graph at the end of this process. Note that we remove less than kd blue edges in total. Since at the beginning $|E_b| \geqslant 2kd$, we have $|E'_b| \geqslant kd$. Observe also that G' remains (q,ℓ)-red/blue-transitive.

Since V' is (q,ℓ)-red/blue-transitive, for every $u, v \in V'$, there can be at most q red-filled ℓ-walks from u to v. Summing this up over all pairs (u,v)'s implies that the number of red-filled ℓ-walk in G' is at most qk^2.

Moreover, notice that $|\mathcal{W}_\ell^{G'}| \geqslant (kd) \cdot d^{\ell-1} \geqslant 2qk^2$; this is because there are at least kd choices for (v_1, v_2) (i.e. all blue edges) and, for $(v_1, \ldots, v_{i-1})$, there are at least d choices for v_i.

Hence, we have $|\widehat{\mathcal{W}}_\ell^{G'}|/|\mathcal{W}_\ell^{G'}| \leqslant 1/2$. This implies that $1/2 \leqslant \Pr_{(v_1,\ldots,v_{\ell+1}) \in \mathcal{W}_\ell^{G'}}[(v_1, \ldots, v_{\ell+1}) \notin \widehat{\mathcal{W}}_\ell^{G'}]$. By union bound, this probability is at most $\sum_{\substack{i,j \in [\ell+1] \\ j > i+1}} \Pr_{(v_1,\ldots,v_{\ell+1}) \in \mathcal{W}_\ell^{G'}}[\{v_i, v_j\} \notin E'_r]$.

Now, note that the number of pairs of $i, j \in [\ell+1]$ such that $j > i+1$ is $\binom{\ell+1}{2} - \ell \leqslant \ell^2/2$. This implies that there exists one such i, j such that $\Pr_{(v_1,\ldots,v_{\ell+1}) \in \mathcal{W}_\ell^{G'}}[\{v_i, v_j\} \notin E'_r] \geqslant 1/\ell^2$. Observe that the probability $\Pr_{(v_1,\ldots,v_{\ell+1}) \in \mathcal{W}_\ell^{G'}}[\{v_i, v_j\} \notin E'_r]$ is bounded above by

$$\max_{u,v} \Pr_{(v_1,\ldots,v_{\ell+1}) \in \mathcal{W}_\ell^{G'}}[\{v_i, v_j\} \notin E'_r \mid v_{i+1} = u \wedge v_{j-1} = v]$$

where the maximization is taken over all $u, v \in V'$ such that $\Pr_{(v_1,\ldots,v_{\ell+1}) \in \mathcal{W}_\ell^{G'}}[v_{i+1} = u \wedge v_{j-1} = v] > 0$. Hence, we can conclude that there exists $u^*, v^* \in V'$ such that

$$\Pr_{(v_1,\ldots,v_{\ell+1}) \in \mathcal{W}_\ell^{G'}}[\{v_i, v_j\} \notin E'_r \mid v_{i+1} = u^* \wedge v_{j-1} = v^*] \geqslant 1/\ell^2.$$

The expression on the left is exactly $|\{(u,v) \in N_b(u^*) \times N_b(v^*) \mid \{u,v\} \notin E'_r\}|/(|N_b(u^*)| \cdot |N_b(v^*)|)$. From this and from every vertex in G' has blue-degree at least d, $U_1 = N_b(u^*), U_2 = N_b(v^*)$ are the desired sets. ◀

4.3 Decoding a Good Assignment From Non-Red Biclique

Finally, we will decode a good assignment for Φ from a sufficiently large non-red biclique in the consistency graph $G^{\sigma,\zeta}$. Recall that a non-red biclique in $G^{\sigma,\zeta}$ simply corresponds to two subcollections $\mathsf{S}_1, \mathsf{S}_2$ such that, for every $(S_1, S_2) \in \mathsf{S}_1 \times \mathsf{S}_2$, $\mathrm{disagr}(\sigma_{S_1}, \sigma_{S_2}) \leqslant \zeta n$. The main result of this subsection is that, given such $\mathsf{S}_1, \mathsf{S}_2$, if both S_1 and S_2 are sufficiently uniform, then we can find a good assignment for Φ. This is stated more precisely below.

▶ **Lemma 23.** *Let $\mathsf{S}_1, \mathsf{S}_2$ be any (γ, μ)-uniform collections of subsets of $\mathcal{C}$. If there is a labeling σ of $\mathsf{S}_1 \cup \mathsf{S}_2$ such that $\mathrm{disagr}(\sigma_{S_1}, \sigma_{S_2}) \leqslant \zeta n$ for every $S_1 \in \mathsf{S}_1$ and $S_2 \in \mathsf{S}_2$, then $\mathrm{val}(\Phi) \geqslant 1 - 2\mu - 6\Delta\zeta/\gamma^2$.*

As outlined earlier, the assignment we take is the majority assignment ψ_{maj} of $\{\sigma_{S_1}\}_{S_1 \in \mathsf{S}_1}$. The key to proving that ψ_{maj} violates few clauses is that, if a clause C is violated, then, for each $S_2 \in \mathsf{S}_2$ that contains C, σ_{S_2} and ψ_{maj} must disagree on at least one variable in $\mathrm{var}(C)$ because σ_{S_2} satisfies C but ψ_{maj} violates it. Hence, if C appears often in both S_1 and S_2, then it contributes to many disagreements between S_1 and S_2; the uniformity condition help us ensure that most C indeed appears often in S_1 and S_2. On the other hand, $\mathrm{disagr}(\sigma_{S_1}, \sigma_{S_2})$ is small for every $S_1 \in \mathsf{S}_1$ and $S_2 \in \mathsf{S}_2$, meaning that there cannot be too many disagreements in total. Comparing this upper and lower bound gives us the desired result. Due to space constraint, we omit the full analysis from this version of the paper.

5 Inapproximability Results of 2-CSPs and DSN

The inapproximability results for 2-CSPs can be shown simply by plugging in the appropriate parameters to Theorem 16. More specifically, for ETH-hardness, since there is a polylog$m$ loss in the PCP Theorem (Theorem 6), we need to select our $\alpha = 1/\mathrm{polylog}m$ so that the size (and running time) of the reduction is $2^{o(m)}$. Recall in Lemma 15 that we need $m \geqslant \Omega(\alpha^{\ell})$, meaning that ℓ can be at most $O(\log m / \log\log m)$. We will pick ℓ to be just $\sqrt{\log m}$. We will finally pick k to be $\exp(\ell \log \ell/\alpha) = \exp(\mathrm{polylog}m)$; this is so that the non-edge biclique size $\log k/(\ell \log(4\ell^2))$ (from Theorem 16) is large enough that we can use Lemma 15 to guarantee its uniformity. Other parameters are chosen accordingly. We omit the full proof, which consists almost solely of calculations, from this version of the paper.

For the inapproximability based on Gap-ETH, we do not incur a loss of polylog$m$ from the PCP Theorem anymore. Thus, we can choose α to be any function that converges to zero as k goes to infinity, e.g., $\alpha = 1/\log\log k$. Now note that the parameter r in Theorem 16 for the intersection disperser property grows with $(1/\alpha)^{\ell}$ (see Lemma 15). Since the soundness guarantee in Theorem 16 is of the form $k^{O(1/\ell)}(r\ell)^{O(1)}/k = k^{O(1/\ell)}(1/\alpha)^{O(\ell)}/k$, it is minimized when ℓ is roughly $\sqrt{\log k}$, which yields the bound $2^{(\log k)^{1/2+o(1)}}/k$.

The inapproximability results for DSN can be proved by simply plugging the hardness of approximation of 2-CSPs to the following known reduction from 2-CSPs to DSN.

▶ **Lemma 24** ([16, Lemma 27][6])**.** *There is a polynomial time reduction that, given a 2-CSP instance Γ where the constraint graph is a complete graph on k variables, produces an edge-weighted directed graph G and a set of demands $\mathcal{D} = \{(s_1, t_1), \ldots, (s_{k^2-k}, t_{k^2-k})\}$ s.t.*

- *If $\mathrm{val}(\Gamma) = 1$, then there exists a subgraph H of cost 1 that satisfies all demands.*
- *If $\mathrm{val}(\Gamma) < \gamma$, then every subgraph satisfying all demand pairs has cost more than $\sqrt{2/\gamma}$.*

[6] While the reduction is attributed to Dodis and Khanna [25], the lemma below is extracted from [16] since, in [25], the full description of the reduction and its properties are left out due to space constraint.

6 Conclusion and Discussions

We prove ETH-hardness of approximating 2-CSPs within $k^{1-o(1)}$ factor where k denotes the number of variables. This ratio is nearly tight since a trivial algorithm yields an $O(k)$-approximation. Under Gap-ETH, we strengthen our result to rule out not only polynomial time but also FPT time algorithms parameterized by k. Due to a known reduction, our result implies $k^{1/4-o(1)}$ factor inapproximability of DSN where k is the number of demand pairs.

Of course the polynomial SSC still remains open and resolving it will advance our understanding of approximability of many problems. Even without fully resolving the conjecture, it may still be good to further study the interaction between the number of variables k and the alphabet size n. For instance, while we show the inapproximability result with ratio $k^{1-o(1)}$, the dependency between n and k is quite bad; in our ETH-hardness reduction, n is $2^{2^{(\log k)^{\Theta(1)}}}$. Would it be possible to improve this dependency (say, to $n = k^{\text{polylog}k}$)?

Another interesting direction is to try to prove similar hardness results as ours for other problems. For example, Densest k-Subgraph (DkS) is one such candidate problem; similar to 2-CSPs with k variables, the problem can be approximated trivially to within $O(k)$-factor and no polynomial (or even FPT) time $k^{1-\varepsilon}$-approximation algorithm is known for the problem. Hence, it may also be possible to prove ETH-hardness of factor $k^{1-o(1)}$ for DkS.

Acknowledgments. We would like to thank Prahladh Harsha for useful discussions. Pasin would also like to thank Rajesh Chitnis and Andreas Emil Feldmann for insightful discussions regarding DSN.

References

1 Amir Abboud and Greg Bodwin. Reachability preservers: New extremal bounds and approximation algorithms. *CoRR*, abs/1710.11250, 2017. `arXiv:1710.11250`.

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, may 1998.

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

4 Sanjeev Arora and Madhu Sudan. Improved low degree testing and its applications. In *STOC*, pages 485–495, 1997.

5 Mihir Bellare, Shafi Goldwasser, Carsten Lund, and Alexander Russell. Efficient probabilistically checkable proofs and applications to approximations. In *STOC*, pages 294–304, 1993.

6 Eli Ben-Sasson, Oded Goldreich, Prahladh Harsha, Madhu Sudan, and Salil P. Vadhan. Robust PCPs of proximity, shorter PCPs, and applications to coding. *SIAM J. Comput.*, 36(4):889–974, 2006.

7 Eli Ben-Sasson, Yohay Kaplan, Swastik Kopparty, Or Meir, and Henning Stichtenoth. Constant rate PCPs for Circuit-SAT with sublinear query complexity. *J. ACM*, 63(4):32:1–32:57, 2016.

8 Eli Ben-Sasson and Madhu Sudan. Short PCPs with polylog query complexity. *SIAM J. Comput.*, 38(2):551–607, 2008.

9 Piotr Berman, Arnab Bhattacharyya, Konstantin Makarychev, Sofya Raskhodnikova, and Grigory Yaroslavtsev. Approximation algorithms for spanner problems and directed steiner forest. *Inf. Comput.*, 222:93–107, 2013.

10 Edouard Bonnet, Bruno Escoffier, Eun Jung Kim, and Vangelis Th. Paschos. On subexponential and FPT-time inapproximability. *Algorithmica*, 71(3):541–565, 2015.

11 Parinya Chalermsook, Marek Cygan, Guy Kortsarz, Bundit Laekhanukit, Pasin Manurangsi, Danupon Nanongkai, and Luca Trevisan. From Gap-ETH to FPT-inapproximability: Clique, Dominating Set, and more. In *FOCS*, pages 743–754, 2017.

12 Siu On Chan. Approximation resistance from pairwise-independent subgroups. *J. ACM*, 63(3):27:1–27:32, 2016.

13 Moses Charikar, Chandra Chekuri, To-Yat Cheung, Zuo Dai, Ashish Goel, Sudipto Guha, and Ming Li. Approximation algorithms for directed steiner problems. *J. Algorithms*, 33(1):73–91, 1999.

14 Moses Charikar, MohammadTaghi Hajiaghayi, and Howard J. Karloff. Improved approximation algorithms for label cover problems. *Algorithmica*, 61(1):190–206, 2011.

15 Chandra Chekuri, Guy Even, Anupam Gupta, and Danny Segev. Set connectivity problems in undirected graphs and the directed steiner network problem. *ACM Trans. Algorithms*, 7(2):18:1–18:17, 2011.

16 Rajesh Chitnis, Andreas Emil Feldmann, and Pasin Manurangsi. Parameterized approximation algorithms for directed steiner network problems. *CoRR*, abs/1707.06499, 2017.

17 Eden Chlamtác, Michael Dinitz, Guy Kortsarz, and Bundit Laekhanukit. Approximating spanners and directed steiner forest: Upper and lower bounds. In *SODA*, pages 534–553, 2017.

18 Irit Dinur. The PCP theorem by gap amplification. *J. ACM*, 54(3):12, 2007.

19 Irit Dinur. Mildly exponential reduction from gap 3SAT to polynomial-gap label-cover. *ECCC*, 23:128, 2016.

20 Irit Dinur, Eldar Fischer, Guy Kindler, Ran Raz, and Shmuel Safra. PCP characterizations of NP: Toward a polynomially-small error-probability. *Computational Complexity*, 20(3):413–504, 2011.

21 Irit Dinur, Prahladh Harsha, and Guy Kindler. Polynomially low error PCPs with polyloglog n queries via modular composition. In *STOC*, pages 267–276, 2015.

22 Irit Dinur and Tali Kaufman. High dimensional expanders imply agreement expanders. In Chris Umans, editor, *58th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2017, Berkeley, CA, USA, October 15-17, 2017*, pages 974–985. IEEE Computer Society, 2017. doi:10.1109/FOCS.2017.94.

23 Irit Dinur and Inbal Livni Navon. Exponentially small soundness for the direct product z-test. In *CCC*, pages 29:1–29:50, 2017.

24 Irit Dinur and David Steurer. Analytical approach to parallel repetition. In *STOC*, pages 624–633, 2014.

25 Yevgeniy Dodis and Sanjeev Khanna. Design networks with bounded pairwise distance. In *STOC*, pages 750–759, 1999.

26 Moran Feldman, Guy Kortsarz, and Zeev Nutov. Improved approximation algorithms for directed steiner forest. *J. Comput. Syst. Sci.*, 78(1):279–292, 2012.

27 Johan Håstad. Some optimal inapproximability results. *J. ACM*, 48(4):798–859, 2001.

28 Russell Impagliazzo, Valentine Kabanets, and Avi Wigderson. New direct-product testers and 2-query pcps. *SIAM J. Comput.*, 41(6):1722–1768, 2012. doi:10.1137/09077299X.

29 Russell Impagliazzo and Ramamohan Paturi. On the complexity of k-SAT. *J. Comput. Syst. Sci.*, 62(2):367–375, 2001.

30 Russell Impagliazzo, Ramamohan Paturi, and Francis Zane. Which problems have strongly exponential complexity? *J. Comput. Syst. Sci.*, 63(4):512–530, 2001.

31 Subhash Khot, Guy Kindler, Elchanan Mossel, and Ryan O'Donnell. Optimal inapproximability results for MAX-CUT and other 2-variable CSPs? *SIAM J. Comput.*, 37(1):319–357, 2007.

32 Tamás Kővári, Vera T. Sós, and Pál Turán. On a problem of K. Zarankiewicz. *Colloquium Mathematicae*, 3(1):50–57, 1954.

33 Pasin Manurangsi. Almost-polynomial ratio ETH-hardness of approximating densest k-subgraph. In *STOC*, pages 954–961, 2017.

34 Pasin Manurangsi and Prasad Raghavendra. A birthday repetition theorem and complexity of approximating dense CSPs. *CoRR*, abs/1607.02986, 2016.

35 Dana Moshkovitz. Low-degree test with polynomially small error. *Computational Complexity*, 26(3):531–582, 2017.

36 Dana Moshkovitz and Ran Raz. Sub-constant error probabilistically checkable proof of almost-linear size. *Computational Complexity*, 19(3):367–422, 2010.

37 Prasad Raghavendra. Optimal algorithms and inapproximability results for every CSP? In *STOC*, pages 245–254, 2008.

38 Ran Raz. A parallel repetition theorem. *SIAM J. Comput.*, 27(3):763–803, 1998.

39 Ran Raz and Shmuel Safra. A sub-constant error-probability low-degree test, and a sub-constant error-probability PCP characterization of NP. In *STOC*, pages 475–484, 1997.

40 Ronitt Rubinfeld and Madhu Sudan. Robust characterizations of polynomials with applications to program testing. *SIAM J. Comput.*, 25(2):252–271, 1996.

Towards a Unified Complexity Theory of Total Functions*†

Paul W. Goldberg[1] and Christos H. Papadimitriou[2]

1 **Department of Computer Science, University of Oxford, Oxford, UK**
Paul.Goldberg@cs.ox.ac.uk
2 **Department of Computer Science, Columbia University, New York, USA**
christos@cs.columbia.edu

Abstract

The class TFNP, of NP search problems where all instances have solutions, appears not to have complete problems. However, TFNP contains various syntactic subclasses and important problems. We introduce a syntactic class of problems that contains these known subclasses, for the purpose of understanding and classifying TFNP problems. This class is defined in terms of the search for an error in a concisely-represented formal proof. Finally, the known complexity subclasses are based on existence theorems that hold for finite structures; from Herbrand's Theorem, we note that such theorems must apply specifically to finite structures, and not infinite ones.

1998 ACM Subject Classification F.1.3 Complexity Measures and Classes , F.4.1 Mathematical Logic

Keywords and phrases Computational complexity, first-order logic, proof system, NP search functions, TFNP

Digital Object Identifier 10.4230/LIPIcs.ITCS.2018.37

1 Introduction

The complexity class TFNP is the set of *total function* problems that belong to NP; that is, every input to such a nondeterministic function has at least one output, and outputs are easy to check for validity – but it may be hard to find an output. It is known from Megiddo [21] that problems in TFNP cannot be NP-complete unless NP is equal to co-NP. On the other hand, various TFNP problems, such as Factoring and NASH are believed to be genuinely hard [26, 10, 8].

Presently, our understanding of the complexity of TFNP problems is a bit fragmented. Currently, our main means for deriving evidence of hardness for TFNP problems is by showing completeness in one of the five known subclasses of TFNP, corresponding to well-known elementary non-constructive existence proofs:

- PPP (embodying the pigeonhole principle);
- PPAD (embodying the principle "every directed graph with an unbalanced node must have another");
- PPADS (same as PPAD, except we are looking for an *oppositely unbalanced* node);
- PPA ("every graph with an odd-degree node must have another"), and
- PLS ("every dag has a sink").

* This work was supported by NSF grant CCF-1408635.
† A full version of the paper is available at ECCC, https://eccc.weizmann.ac.il/report/2017/056/

Much is known about these classes. PPP is known to contain PPAD and PPADS, while essentially all other possible inclusions are known to be falsifiable by oracles, see for example [1]. They all have complete problems (actually, the most commonly used definition of, for example, PPAD is "all NP search problems reducible to End of the Line"), and most (PLS, PPAD, PPA) have many other natural complete problems besides the basic one.

Even the union of these classes does not provide a home for all natural TFNP problems. For example, Factoring is only known to be reducible to PPP and PPA through randomized reductions [16]. The problem Ramsey (e.g., "Given a Boolean circuit encoding the edges of a graph with 4^n nodes, find n nodes that are either a clique or an independent set") is not known to be in any one of the five classes, and the same obtains for a problem that could be called Bertrand-Chebyshev ("Given n, produce a prime between n and $2n$").

The status quo in TFNP, as described above, is a bit unsatisfactory. Many natural questions arise: Are there other important complexity subclasses of TFNP, corresponding to novel nonconstructive arguments? Can the three rogue problems above (along with a few others) be classified in a more satisfactory way?

More importantly, *is there a more holistic, unified approach to the complexity of TFNP problems?* For example, are there TFNP-complete problems? The answer here is strongly believed to be "no", as TFNP (the set of all polynomial-depth nondeterministic computations that have a witness, for every input) is very similar in spirit and detail to the classes UP (computations with at most one witness, for every input) and BPP (computations whose fraction of witnesses is bounded away from half, for every input), both known to have no complete problems under oracles [27, 12]. Indeed, Pudlák ([25], Section 6) presents a similar result specifically for TFNP. Hence, this route for a unified complexity view of total functions is not available.

This paper aims to develop a more unified complexity theory of TFNP problems. We define a new subclass of TFNP that includes all five known classes. This new class, which we call PTFNP[1] (for "provable TFNP"), does have complete problems, and these problems are therefore natural generalisations of all known completeness results in TFNP.

In particular, we define a kind of *consistency search problem*, a notion that has recently been studied in the literature on Bounded Arithmetic [3]. Fix a consistent deductive system – in this paper we use a propositional proof system that we call Q-EFF (for "quantified boolean formulae with extended Frege functions"; it allows lines of a proof to define new n-ary functions). Now consider a Boolean circuit which, when input an integer j, produces the jth line of an exponentially long purported proof in this system (the line itself is of polynomial length). Suppose further that this proof arrives at a contradiction (one of the lines is "false"). There surely must be a mistaken line in this proof; the challenge is to find it! We call this problem Wrong Proof, and we define PTFNP as the set of all search problems reducible to it; it is obviously a subset of TFNP. We establish that PTFNP contains PPP (and by extension, PPAD and PPADS), and also PPA and PLS. The study of exponentially-long proofs that are presented concisely via a circuit was introduced by Krajíček [19].

Of course, any finite collection of problems – or classes with complete problems – can be generalised in a rather trivial way, by proposing a new problem or class that artificially incorporates the key features of the old ones. However, Wrong Proof makes no explicit

[1] The class should perhaps be called $\text{PTFNP}_{\text{Q-EFF}}$ since it is defined with respect to a deductive system Q-EFF that we introduce and use in our proofs here. Similar definitions with respect to other proof systems are possible. In this paper we just refer to it as PTFNP. A similar point applies to the problem Wrong Proof used to define PTFNP.

reference to the problems that are complete for the above complexity classes. Its proof system Q-EFF uses quantified boolean formulae with polynomially-many propositional variables, an exponential sequence of n-ary function symbols, and no predicates. The novel features that we exploit are the ability to use exponentially many steps, together with the exponential sequence of function symbols.

Does PTFNP contain Factoring, Ramsey, and Bertrand-Chebyshev? In the final section we discuss these questions further. Finally, notice that the heretofore "five subclasses" of TFNP correspond to five elementary non-constructive existence arguments in combinatorics, and all these five elementary arguments share one intriguing property: *They only hold for finite structures,* and are false in infinite ones. We show in Section 6 that this is no coincidence: Herbrand's Theorem from 1930 [13, 6] tells us that any existential sentence in predicate calculus that is true for all models (finite and infinite) is equivalent to the disjunction of a finite number of quantifier-free formulas; it follows that the corresponding TFNP problem is necessarily in P.

1.1 Related Recent Work

Various connections have been made between the complexity of TFNP problems and formal proofs, a research direction that seems timely and productive. In a recent paper [2], Arnold Beckmann and Sam Buss, working within the tradition of bounded arithmetic [3], prove certain results that appear to be closely related to the present ones. They define a problem closely related to our Wrong Proof, and in fact in two versions, one corresponding to Frege systems, and another to extended Frege. Then they show these to be complete for the classes of total function problems in NP whose totality is provable within the bounded arithmetic systems U_2^1 and V_2^1, respectively. Our system Q-EFF differs in using propositional variables only, but arithmetic theories can be translated into propositional ones (see [18], Chapter 9).

There are some well-known reducibilities amongst PPAD-like complexity classes, for example that PPAD reduces to PPADS, which reduces to PPP. Buss and Johnson [7] connect these results with derivability relationships (in a proof system) amongst the combinatorial principles that guarantee that they represent total search problems; so for example, the principle underlying PPAD can be derived from the one underlying PPADS, and generally, any such derivability result would tell us that the deriving corresponding complexity classes generalises the other. Our focus here, in contrast, is on formal proofs that correspond with individual *instances* of TFNP problems (finding an error in the proof allows us to find a solution for the corresponding problem-instance).

Pudlák [25] shows how every TFNP problem reduces to a *Herbrand consistency search problem*: any TFNP problem X is characterised by an associated formula Φ whose Herbrand extension is guaranteed to be satisfiable, but the challenge of finding a satisfying assignment is equivalent to X. This correspondence is somewhat reminiscent of Fagin's theorem. The focus of [25] is not on syntactic guarantees that we have a total search problem: it would be hard to check whether a given Φ corresponds to a TFNP problem. By contrast, our definition of Wrong Proof is intended as a highly-general TFNP problem for which there is a syntactic guarantee that any instance has a solution.

In contrast with most TFNP-related work within bounded arithmetic, we focus on the "white box" concise circuit model of the functions that define the problems characterising the complexity classes of interest. In some respects this makes a significant difference: for example, a recent paper of Komargodski et al. [17] shows that any such TFNP problem has a *query complexity* proportional to the description-size of a problem instance. However, a reduction using the oracle model should allow a logical description of a circuit to be plugged in.

Finally, Hubáček et al. [14] show that hard-on-average NP problems lead to hard-on-average TFNP problems. The TFNP problems thus constructed are specific to the associated NP problems; our concern here, in contrast, is to identify a single easily-understood TFNP problem that generalises previous ones.

1.2 Background on propositional proofs and the pigeonhole principle

In 1979, Cook and Reckhow [9] initiated the study of the proofs of propositional tautologies, with regard to the question of how long do such proofs need to be. Abstractly, a *proof system* for a language (here, the set of tautologies) is a scheme for producing efficiently-checkable certificates for words in that language. As noted in [9], a *polynomially bounded* proof system for tautologies is only possible if NP is equal to co-NP. They obtain results that various proof systems can efficiently simulate each other; these results allow us to conclude that one such system is polynomially bounded if and only if another such system is.

[9] introduce *Frege* and *extended Frege* systems: roughly, in a Frege system a proof consists of a sequence of lines containing propositional formulae that are either generated by some axiom scheme (and are known to hold for that reason) or are derivable by modus ponens from two formulae in previous lines of the proof. In an extended Frege system, we also allow lines that introduce a new propositional variable and set it to equal a propositional formula ϕ over pre-existing variables. The new variable can then be plugged in to a larger formula as a shorthand for ϕ, and if this process is iterated, it may result in an exponential saving in space. It remains a central open problem in proof complexity whether extended Frege proofs can in general be simulated by Frege proofs, with only a polynomial blowup in size of the proof.

In studying this question, various candidate classes of formulae have been considered, the most widely-studied being ones that express the *pigeonhole principle*, as introduced in [9]. The "$n+1$ into n" version of this, denoted PHP_n^{n+1}, states that a function from $n+1$ input values to n output values must map two different inputs to the same output. That is, $f : [n+1] \to [n]$ must have a *collision*: two inputs that f maps to the same output[2]. f can be described by a propositional formula ψ (whose variables indicate which numbers map to which according to f, specifically, variable P_{ij} is TRUE if and only if i is mapped to j) stating "each number in the domain maps to some number in the codomain, and any pair map to different values." By the pigeonhole principle, ψ is unsatisfiable, so its negation ϕ is a tautology (and ϕ has size polynomial in n). [9] gave polynomially-bounded extended Frege proofs of these expressions. Buss [5] subsequently gave polynomially-bounded Frege proofs of these, and in [4] quasi-polynomial size Frege proofs that are a reformulation of the extended Frege proofs of [9]. See [4] for a discussion of other candidate classes of formulae and progress that has been made on them.

Papadimitriou [23] introduced the PIGEONHOLE CIRCUIT problem, in which a pigeonhole function on an *exponential-sized* domain is concisely presented via a boolean circuit C. ψ as constructed above would be exponentially large in C, but a "dual" statement that two inputs to C map to the same output can still be expressed as a concise propositional formula ϕ. By construction, ϕ is satisfiable, and a short proof of this fact consists of a satisfying assignment, but in general such a satisfying assignment appears to be hard to find, and this search characterises the complexity class PPP. In seeking to better understand the challenge, we find a new point of contact between the pigeonhole principle and proof complexity. The

[2] We use the standard notation that for a positive integer x, $[x]$ denotes the set $\{1, 2, \ldots, x\}$.

difference here is we have a propositional formula that is known to be satisfiable; we want to exhibit a proof of this; but the naive approach of just exhibiting a satisfying assignment is believed to be hard, so instead we fall back on a long and "opaque" proof of satisfiability.

A general question we have only partly answered is, what sort of logic is needed to express such a proof? Our deductive system Q-EFF is first-order, but we require (exponentially many) lines that define the behaviour of additional function symbols (mapping n-bit strings to single-bit outputs). This seems to be a more powerful facility than the additional variable symbols allowed by extended Frege proofs (hence the name Q-EFF for "extended Frege functions"). As we discuss in the penultimate section, it would be of interest to see if these proofs could be done just defining exponentially many additional propositional variables.

1.3 Organisation of this paper

Section 2 gives details of our deductive system and the problem Wrong Proof. Section 3 shows how to prove unsatisfiability of certain existential expressions, in such a way that any error in the proof allows a satisfying assignment to be readily reconstructed. Sections 4 and 5 reduce PPP, PPA, and PLS problem-instances to proofs that corresponding existential expressions are satisfiable. (The expressions are the ones we can also "prove" unsatisfiable.) In Section 6 we apply Herbrand's theorem to show that only "finitely valid" combinatorial principles may give rise to hard total search problems. We conclude in Section 7.

2 Deductive systems and the Wrong Proof problem

A deductive system (or proof system) is a mechanism for generating expressions in some well-defined (formal) language. The expressions should come with a semantics, defining which ones are true and which false. A basic property of a system is *consistency*, that it should not be able to generate two expressions that contradict each other. Consistency is ensured if the rules of the system are valid, in the sense that we cannot deduce any false expressions from true ones. For the deductive system Q-EFF in this paper, the language (and corresponding semantics) of interest is simple and straightforward, and it is not hard to check that it is consistent. The Wrong Proof problem of Definition 2 formalises the computational challenge of receiving a proof of two expressions that contradict each other, and searching for an erroneous step in the proof (guaranteed to exist by the contradiction that we are shown).

The set of expressions that can be produced by a deductive system are called the *theorems* of the system. The system is usually given in terms of a set of *axioms* and *inference rules* that allow theorems to be derived from other theorems. A proof consists of a sequence of numbered *lines*. A line contains a well-formed formula that either holds due to some axiom, or is inferable from the contents of previous lines. A typical line contains one of the following kinds of expression:

$$\ell, \ell' \vdash A, \quad \text{or} \quad \ell \vdash A, \quad \text{or} \quad \vdash A,$$

where A is a well-formed formula inferred at the current line, and ℓ, ℓ' are the numbers of earlier lines (ℓ, ℓ' are thus strictly smaller than the current line number). The expression "$\ell, \ell' \vdash A$" means that the current line claims that A is inferable from the formulae located at lines ℓ and ℓ' (using one of the given inference rules). "$\ell \vdash A$" means that A is inferable from the formula located at line ℓ. "$\vdash A$" means that A holds ipso facto (due to an axiom, e.g. rule (1) lets us write $\vdash (A \vee \neg A)$, for any well-formed formula A).

Our system makes use of a kind of *extension axiom* line, written as $f(x) \leftrightarrow \phi(x)$, where f is a new function symbol whose value on input x is defined by ϕ. f should not occur within

ϕ, or in any previous line. So, these lines allow us to define new boolean functions that may appear in later lines.[3]

▶ **Definition 1.** With respect to some given consistent deductive system, a *circuit-generated proof* consists of a directed boolean circuit C having n input nodes. C has a corresponding formal proof having 2^n lines. The output of C on input $\ell \in [2^n]$ contains the theorem that has been deduced at line ℓ, together with the numbers of any earlier line(s) from which ℓ's theorem has been deduced.

In constructing circuit-generated proofs, it is often convenient to identify various exponentially-long sequences of line numbers without assigning numerical values to those line numbers. We can accommodate a collection of such sequences in a circuit-generated proof of size $O(n)$, possibly padded out with unused lines whose theorems consist of the constant TRUE.

▶ **Definition 2.** Let S be a consistent deductive system having the property that any line ℓ of a proof that uses S can be checked for correctness in time polynomial in the length of ℓ. An instance of Wrong Proof consists of a circuit-generated proof Π_C represented by boolean circuit C.

Π_C contains two given lines (say, lines 2^n and $2^n - 1$) that contradict each other: One of them contains as its theorem some expression A and other contains expression $\neg A$. The challenge is to identify some line number ℓ whose corresponding theorem is not derivable in the way stated by $C(\ell)$. Since S is consistent and we have observed a contradiction, such a line must exist.

Wrong Proof is in TFNP: any incorrect line of an instance of Wrong Proof can readily be verified to be incorrect. We have so far defined Wrong Proof rather abstractly, with respect to an unspecified deductive system. In this paper we focus on a specific deductive system that we describe in detail in the rest of this section.

2.1 The formulae and theorems of our system Q-EFF; some notation

We work with expressions of quantified propositional logic (variables take values TRUE/FALSE), augmented with a sequence of n-ary function symbols. We also use, for convenience, symbols such as x and y to denote vectors of n propositional variables, and expressions like $x < y$ to denote relationships between x and y, regarding these vectors as representing numbers in $[2^n]$. $x^{(0)}, x^{(1)}, x^{(2)}$ denote respectively the n-vectors (FALSE, ..., FALSE), (FALSE, ..., FALSE, TRUE), (FALSE, ..., FALSE, TRUE, FALSE), or the numbers $2^n, 1, 2$. Since the all-zeroes vector $x^{(0)}$ corresponds to 2^n, this means that $x^{(0)} \geq x$ for any other vector x (this convention tends to reduce clutter in our expressions).

In this paper, the two contradictory statements in an instance of Wrong Proof take the form $\exists x, x'(\phi(x, x'))$ and $\neg\exists x, x'(\phi(x, x'))$, asserting that some $2n$-variable formula ϕ is (respectively, isn't) satisfiable. We continue with more detail on the expressions used in our proofs.

[3] This facility to define the behaviour of new functions is a rather novel feature of our system, and gives rise to the question of whether we should be able to make do with standard extended Frege axioms. An extended Frege system is a propositional proof system that allows us to use extension axiom lines of the form $x^{(new)} \leftrightarrow \phi$, where $x^{(new)}$ is a variable symbol that has not occurred previously in the proof, and ϕ is a formula that gives the value of $x^{(new)}$ in terms of pre-existing variables. So, we are allowing ourselves to define new functions on vectors of boolean variables, as opposed to just individual variables. In Section 3.1 we explain why it is useful to have these extension-axiom lines that define new functions.

For complexity parameter n, the vocabulary we use contains a polynomial-size collection of variable symbols, together with an *exponential-size* collection of n-ary function symbols; these are denoted by f_i where $i \in [2^n]$. In our proofs, f_{2^n} is defined in terms of an instance of some TFNP problem, and (for each $i \in [2^n]$) f_{i-1} is defined in terms of f_i via an extension-axiom line. There are no predicates. The expressions we use are first-order, in that they may have quantification over the variable symbols, but not the functions.

While we work with expressions whose variables represent vectors of propositional variables, note that such expressions represent polynomially-larger expressions whose variables are simple propositional variables. Variable x represents $(x_1, \ldots, x_n)$ where the x_i are propositional variables, and expressions involving x can be converted to basic propositional formulae in the individual x_i without an excessive blowup in the size of the formula. This extra syntax makes our expressions more concise and readable. For example, given non-zero vectors x, x', the expression $x < x'$ represents the following propositional formula involving the variables x_i and x'_i (treating x_1 and x'_1 as the most significant bits):

$$\neg x_1 \wedge x'_1 \vee (x_1 = x'_1 \wedge (\neg x_2 \wedge x'_2 \vee (x_2 = x'_2 \wedge (\neg x_3 \wedge x'_3 \vee \cdots (\neg x_n \wedge x'_n))) \cdots))$$

Another notational convenience that we use is expressions such as $\forall x < y(\phi(x, y))$, meaning $\forall x, y(x < y \rightarrow \phi(x, y))$, or if y is a vector of propositional constants, it would mean $\forall x(x < y \rightarrow \phi(x, y))$. Similarly, $\exists x \neq y(\phi(x, x'))$ means $\exists x, x'(x \neq x' \wedge \phi(x, x'))$.

2.2 Axioms and inference rules

We use the following kinds of rules:

- Axioms (written as $\vdash A$) let us write down certain expressions that can be seen to evaluate to TRUE based on some easily-checkable property, for example A is of the form $B \vee \neg B$.
- Inference rules, written as $A, B \vdash C$ for example, say that given expressions A and B, we can write the expression C.
- Equivalences, written as $A \equiv B$, say that two expressions are logically equivalent. An equivalence represents a *rule of replacement* in that it may be applied to sub-expressions of any expression that appears in a line of a proof. For example, using the equivalence $A \wedge B \equiv B \wedge A$ we could take a line ℓ containing the expression $\text{TRUE} \vee (x_i \wedge y_i)$ and write a new line containing $\ell \vdash \text{TRUE} \vee (y_i \wedge x_i)$.
- "Extension axiom" lines define new n-ary functions, and are written as $f(x) \leftrightarrow \phi(x)$, where f is a new symbol that has not appeared previously in the proof, and ϕ specifies how f behaves on input (n-vector) x. So, this kind of line means $\forall x(f(x) \triangleq \phi(x))$, and the system can use $\forall x(f(x) = \phi(x))$ as a theorem.

Some of the rules we list below are redundant in the sense that they could be simulated using the others. We prefer to limit ourselves to rules that are not too novel and ad-hoc, that are clearly consistent, and which, crucially, allow that any individual line of a proof can be checked for correctness in time polynomial in n. Section 2.3 contains rules that we prove can be simulated by the ones in Section 2.2; usage of these additional rules allows some of the formal proofs to be presented more cleanly. We have not however tried to minimise the collection of rules in Section 2.2; some of the rules in the section can be simulated using the others.

As noted earlier, our extension axiom lines are somewhat novel. A standard extension-axiom line of an extended Frege proof may introduce a new propositional variable and set its value to equal some expression in terms of pre-existing values. Our extension-axiom rules (see rule (13)) allow us to define new *functions* via expressions that define their behaviour in

terms of pre-existing functions. So we call the proof system Q-EFF (for "extended Frege functions") on account of this novel feature.

In the following, A, B, C represent arbitrary well-formed formulae and x, y are length-n vectors of propositional variables, where x (say) may also be thought of as ranging over integers in the range $[2^n]$, as noted in Section 2.1. The equivalences we allow ourselves to use include standard rules of replacement, such as commutativity, associativity, and distributivity of propositional connectives, removal of double negation, and de Morgan's rules. We also use $A \equiv A \vee A \equiv A \wedge A \equiv A \wedge \textsc{true} \equiv A \vee \textsc{false}$, also $A \to B \equiv \neg B \vee A$, and the identity $A \to (B \to C) \equiv (A \wedge B) \to C$. We also allow a step of a proof to rename a bound variable throughout the subexpression where it occurs. These equivalences may be applied to any expression arising in a derivation, also they may be applied (in a simple step) to any well-formed subexpression of a larger expression arising in a derivation. So, a proof line of the form $\ell \vdash A$ may state that A is derived from expression A', where A' is the theorem derived at line ℓ, via applying one of these basic manipulations to A', or to some subexpression of A'. It is easy to see that any such step may be checked for correctness in polynomial time, and there is no need for a line to specify which rule is being used.

For any well-formed expression A, we may use any of the following lines in our proofs:

$$\vdash (A \to A), \quad \vdash (A \vee \neg A), \quad \vdash \textsc{true}. \tag{1}$$

Modus ponens (rule (2)) states that if lines ℓ and ℓ' contain theorems A and $A \to B$ respectively, a subsequent line containing the expression "$\ell, \ell' \vdash B$" is a valid line.

$$A,\ A \to B\ \vdash\ B. \tag{2}$$

"Conjunction introduction" (rule (3)) states that if lines ℓ and ℓ' contain theorems A and B respectively, a subsequent line containing the expression "$\ell, \ell' \vdash A \wedge B$" is valid.

$$A,\ B\ \vdash\ A \wedge B. \tag{3}$$

A "case analysis" rule (4) (a form of disjunction elimination) means that if lines ℓ and ℓ' contain theorems $B \to A$ and $\neg B \to A$, then a subsequent line containing "$\ell, \ell' \vdash A$" is valid.

$$B \to A,\ \neg B \to A\ \vdash\ A. \tag{4}$$

The disjunction introduction rule (5) means that if line ℓ contains theorem A, then a subsequent line containing $\ell \vdash A \vee B$ is valid.

$$A\ \vdash\ (A \vee B). \tag{5}$$

Antecedent strengthening:

$$(A \to C)\ \vdash\ (A \wedge B \to C). \tag{6}$$

Basic equivalences for quantified variables: let x_i be an individual propositional variable; let $A(\textsc{true})$ and $A(\textsc{false})$ be obtained by plugging in the constants $\textsc{true}$ and $\textsc{false}$ respectively in place of x_i, in $A(x_i)$. Then we have:

$$\begin{aligned} \exists x_i(A(x_i)) &\equiv A(\textsc{true}) \vee A(\textsc{false}) \\ \forall x_i(A(x_i)) &\equiv A(\textsc{true}) \wedge A(\textsc{false}) \end{aligned} \tag{7}$$

Distributive rules for quantifiers (recall x is a vector of variables):

$$\begin{array}{rcl} \exists x(A(x)) \vee \exists x(B(x)) & \equiv & \exists x(A(x) \vee B(x)) \\ \forall x(A(x)) \wedge \forall x(B(x)) & \equiv & \forall x(A(x) \wedge B(x)) \end{array} \tag{8}$$

(In the context of circuit-generated proofs, the distributive rules (8) can be derived from the previous rules. Recall that x denotes the n-vector $(x_1, \ldots, x_n)$. Starting from the expression $\forall x(A(x) \wedge B(x))$, we go via intermediate expressions of the form $\forall(x_1, \ldots, x_j)(\forall(x_{j+1}, \ldots, x_n)A(x) \wedge \forall(x_{j+1}, \ldots, n)B(x))$ to end up with $\forall x(A(x)) \wedge \forall x(B(x))$, while keeping all intermediate expressions to be of polynomial length.)

Bringing quantifier to front: suppose A contains no variables in x, then if $\circ$ is any boolean connective, we have

$$\begin{array}{l} A \circ \exists x(B) \equiv \exists x(A \circ B) \\ A \circ \forall x(B) \equiv \forall x(A \circ B) \end{array} \tag{9}$$

Universal instantiation: let $A(t)$ be the expression obtained by plugging in term t in place of variable symbol x (t is any term, i.e. a propositional variable or constant, or a function symbol applied to other terms.)

$$\forall x(A(x)) \ \vdash \ A(t). \tag{10}$$

Universal generalization: if x and y are n-vectors of propositional variables, and x is a vector of free variables, we have

$$A(x) \ \vdash \ \forall y A(y). \tag{11}$$

Existential generalization: if $A(x)$ is obtained by plugging in variable(s) x in place of term(s) t, we have

$$A(t) \ \vdash \ \exists x(A(x)). \tag{12}$$

Extended Frege-style definitions of functions:

We use extension axioms written as:

$$f(x) \leftrightarrow \phi(x) \tag{13}$$

where ϕ is an expression that defines the value of $f(x)$. ϕ may include functions defined earlier, but not f. f is a new function symbol, x is a vector of variable symbols, and $\phi(x)$ is a formula that specifies the value taken by f on any input x. This rule can be understood as saying $\forall x(f(x) \triangleq \phi(x))$.

2.3 Further rules derivable from the ones of Section 2.2

It is useful to note the following further rules for writing down lines of a proof, which can be simulated by the ones of Section 2.2. We can assume we have the "hypothetical syllogism" rule, $A \to B, \ B \to C \vdash A \to C$ (we can simulate this using the rules of Section 2.2: a combination of modus ponens and case analysis). We can also assume we have an "axiom" saying that expressions of the following form can be written down for free: $\forall x(A(x)) \to A(t)$, where t is a n-vector of terms that is plugged in for (n-vector) x in A. (We can write down $\forall x(A(x)) \to \forall x(A(x))$, equivalently $\forall x(A(x)) \to \forall y(A(y))$, where y is another n-vector of propositional variables, equivalently $\forall x, y(A(x) \to A(y))$, then by universal instantiation,

$\forall x(A(x) \to A(t))$, which is equivalent to $\forall x(A(x)) \to A(t)$.) In a similar way, we can write down expressions of the form $A(t) \to \exists x(A(x))$.

We also use the equivalences (derivable from (7) and de Morgan's rules):

$$\begin{aligned} \neg\exists x(A) &\equiv \forall x(\neg A) \\ \neg\forall x(A) &\equiv \exists x(\neg A) \end{aligned} \tag{14}$$

In constructing circuit-generated proofs, it is also convenient to allow the following kind of line. Suppose ϕ is a propositional formula over a vector x of n terms, consisting of variables, or functions applied to variables. Let $i \in [2^n]$ be a satisfying assignment of ϕ, so i is a vector of n constants TRUE/FALSE. We use the following rule, which can be simulated by previous ones:

$$\vdash \; x = i \to \phi(x). \tag{15}$$

We also make use of equivalence (16), which can be simulated in a straightforward way using the previous rules. Letting x be an n-vector of propositional variables and i an n-vector of propositional constants, and ϕ a quantifier-free boolean formula, we have

$$x = i \to \phi(x) \quad \equiv \quad \phi(i). \tag{16}$$

3 Preliminaries to the reductions to Wrong Proof

In this section we establish results that are useful subsequently, and we discuss certain features that our reductions all have in common with each other.

An instance of Wrong Proof is supposed to consist of proofs of two contradictory statements, and in our reductions, these statements take the form $\exists(x, x')\phi(x, x')$ and $\neg\exists(x, x')\phi(x, x')$, for n-vectors x, x' of propositional variables. ϕ depends on the specific instance of a TFNP problem that we reduce from.

Any problem in TFNP is reducible to the search for a satisfying assignment to a propositional formula ϕ, where ϕ obeys some syntactic constraint that guarantees that it does, in fact, have a satisfying assignment.[4] In reducing to Wrong Proof, we "prove" the contradictory statements $\exists(x, x')\phi(x, x')$ and $\neg\exists(x, x')\phi(x, x')$ where x, x' are vectors of n propositional variables. In fact, the ϕ that we use is not purely propositional; it includes a function symbol that is constructed (using our extension-axiom rule) to encode a TFNP problem-instance, in a way described in Section 3.2.

The proofs of these contradictory statements consist of sequences of applications of the rules of Sections 2.2, 2.3, and they are instances of Wrong Proof, i.e. long proofs presented via a circuit. The error occurs in the "proof" of $\neg\exists(x, x')\phi(x, x')$. Of course, it's trivial to exhibit a faulty proof of the unsatisfiability of ϕ, but we require something more, namely that any error should let us efficiently reconstruct a satisfying assignment of ϕ. Lemma 3 shows how to construct such a proof. The three expressions in the statement of Lemma 3 correspond to the existence principles underlying PPP, PPA, and PLS (recall that PPAD and PPADS are special cases of PPP).

[4] To see this, note that for any problem $X \in$ TFNP, any instance I of size n has a solution S_I of size $poly(n)$; solutions are checkable with a poly-time algorithm $\mathcal{A}$ that takes candidate solutions as input and outputs "yes" iff $\mathcal{A}$ received a valid solution. $\mathcal{A}$ can be converted to a circuit and thence to a propositional formula that is satisfied by inputs representing any valid solution S_I of instance I along with extra propositional variables for gates of the circuit.

The proofs of $\exists(x, x')\phi(x, x')$ are done separately for each TFNP problem of interest, in Sections 4 for PPP and 5 (but we leave to the full version, any detail on the proofs for PPA and PLS.). Section 3.1 introduces the general approach taken in Sections 4 and 5. Section 3.2 presents Lemma 3 that shows how to make a suitable proof of $\neg\exists(x, x')\phi(x, x')$.

3.1 Overview of the reductions of Sections 4 and 5

In Sections 4 and 5, we consider computational problems Pigeonhole Circuit, Lonely, and Iter, which are complete for PPAD, PPA, and PLS respectively. We reduce each of these problems to Wrong Proof.

Any instance of the problems Pigeonhole Circuit, Lonely, and Iter is defined in terms of a boolean circuit C. Section 3.2 begins with a general method to define a function f using the rules of Q-EFF, so that f is the function computed by C. We derive from C an existential formula $\Phi = \exists(x, x')\phi(x, x')$ in terms of f stating (correctly) that there is a solution associated with the instance of the problem. We have noted that Section 3.2 shows how to "prove" $\neg\Phi$. Sections 4 and 5 show how to construct contrasting (and correct!) circuit-generated proofs of Φ. The approach to proving that Φ is satisfiable, is based on a syntactic feature that assures us that it is, indeed, satisfiable. These syntactic features are different for the three problems under consideration (which is why we have three different complexity classes), so we need three distinct reductions.

At this point we are ready to explain our usage of extension axioms (rules of type (13)) to define long sequences of new n-ary boolean functions. In the context of Pigeonhole circuit, any instance I has an associated function $f_I : [2^n] \to [2^n - 1]$, and the search is for two inputs to f_I that map to the same output. Call such a pair of inputs a "collision" for f_I. We reduce the search for a collision for f_I to the search for a collision for a new function $f'_I : [2^n - 1] \to [2^n - 2]$. f'_I is defined in terms of f_I using an extension-axiom line. We reduce this in turn to the search for a collision for a new function $f''_I : [2^n - 2] \to [2^n - 3]$, and so on. With an exponential sequence of similar reductions (that can all be efficiently generated via a circuit), we eventually reduce to the search for a collision of a function from $\{1, 2\}$ to $\{1\}$, whose existence has a simple (formal) proof. Lonely and Iter have similar sequences of functions.

Functions defined using rules of type (13) have the codomain $\{\text{TRUE}, \text{FALSE}\}$. f_I can of course be defined in terms of n n-ary functions that map to individual bits of the output of f_I, as can each of the exponential sequence of functions that is derived from it.

We have aimed to make the presentation as consistent as possible for the three reductions to Wrong Proof. The following presentational aspects are shared by the reductions. We let C denote a typical instance of a TFNP problem, since the problem-instances we consider are represented as (boolean) circuits. Π_C denotes the corresponding instance of Wrong Proof. We describe Π_C in terms of the lines of Π_C, as opposed to the circuit that generates it: for the exponential sequences of lines that we define, we assume it is easy to check that they can be compactly represented using a circuit. f denotes the function computed by C; f is constructed using extension-axioms as described at the start of the next subsection. We set a new function f_{2^n} equal to f. The reductions use sequences of well-formed expressions that appear in the instances of Wrong Proof, that we denote A_i, C_i and F_i, for $i \in [2^n]$. F_i is an extension-axiom line that defines new function f_{i-1} in terms of f_i. A_i asserts implicitly (or non-constructively) that an instance of a problem corresponding to function f_i has a guaranteed solution (due to a syntactic property of f_i). C_i is an existential expression that asserts that same thing explicitly. We end up proving C_{2^n} that states the existence of a solution, and C_{2^n} is equivalent to Φ. This contradicts the expression $\neg\Phi$ that is "proved" using Lemma 3.

Here we give some detail on the first reduction (from PIGEONHOLE CIRCUIT), leaving out further detail that appears in appendices of the full version of this paper. We leave to the full version the details on the reductions from LONELY and ITER.

3.2 Construction of functions from circuits, and a method for locating the errors in instances of Wrong Proof

Given a boolean circuit C with n input nodes, Q-EFF can define a function f that computes C as follows. Each gate g of C has an associated n-ary function f_g mapping the inputs to C to the value taken at g. We can construct f using a sequence of extension-axiom rules (of type (13)), in which if, say, gate g is the AND of gates g' and g'', then we add the rule $f_g(x) \leftrightarrow f_{g'}(x) \wedge f_{g''}(x)$. If g is the j-th input gate, then f_g is defined by $f_g(x) \leftrightarrow x_j$, where x_j is the j-th component of n-vector x.

▶ **Lemma 3.** *Suppose f is defined according to the above construction. Consider the expressions*[5]

- $\exists(x,x')((x \neq x' \wedge f(x) = f(x')) \vee f(x) = x^{(0)})$,
- $\exists(x,x')(f(x^{(1)}) \neq x^{(1)} \vee (x \neq x^{(1)} \wedge f(x) = x) \vee (x' = f(x) \wedge x \neq f(x')))$,
- $\exists(x,x')(f(x^{(1)}) = x^{(1)} \vee f(x) < x \vee (x' = f(x) \wedge f(x') = f(x)))$.

We can efficiently construct circuit-generated proofs of the negations of these expressions in such a way that any error in the proof allows us to efficiently construct (x,x') satisfying the expression.

The expressions in the statement of Lemma 3 are the principles underlying PPP, PPA, and PLS, used in Theorems 4, 6, 7. They are all satisfiable, so their negations are all false.

Proof. The negation of any of the above expressions takes the form $\forall(x,x')(\phi(x,x'))$, where ϕ performs some test on values of x, x', $f(x)$, and $f(x')$. For example, the negation of the first of these expressions is

$$\forall(x,x')\neg\Big((x \neq x' \wedge f(x) = f(x')) \vee f(x) = x^{(0)}\Big). \tag{17}$$

We show how to construct a circuit-generated proof of (17) such that any error will identify a pair of n-vectors x, x' whose existence is claimed by the first of the three existential statements. The following approach applies also to the negations of the other two existential expressions in the statement of this lemma.

Let M be the matrix of (17), i.e. the subexpression $\neg((x \neq x' \wedge f(x) = f(x')) \vee f(x) = x^{(0)})$. We continue by giving a method for proving the following stronger expression, from which (17) is derivable:

$$\forall(x,x')(C_1 \wedge \ldots \wedge C_m \wedge M) \tag{18}$$

where C_i are clauses that construct the values of $f(x), f(x')$ by working through the values taken at the gates of the circuit; the C_i are of the form $f_g(x) = f_{g'}(x) \circ f_{g''}(x)$ (for $\circ \in \{\wedge, \vee\}$), or $f_g(x) = \neg f_{g'}(x)$, or $f_g(x) = x_j$ (in the case that g is the j-th input gate). M is a boolean combination of expressions of the form $f_g(x) = f_{g'}(x')$ or $f_g(x) \neq f_{g'}(x')$, for output gates g, g', or of the form $f_g(x) =$ TRUE/FALSE.

[5] Recall that $x^{(0)}$ and $x^{(1)}$ denote the all-zeroes bit-vector, and the bit vector corresponding to number 1.

Let $\phi'(x,x') = C_1 \wedge \ldots \wedge C_m \wedge M$, and for $i \in [2^{2n}]$ let Φ'_i be the formula $\forall(xx' \leq i)\phi'(x,x')$, where xx' represents the $2n$-digit number $2^n(x-1)+x'$. It can be formally proved that (18) is equivalent to $\Phi'_{2^{2n}}$; we omit the details. For each $i \in [2^{2n}]$, some line ℓ_i of the proof contains Φ'_i. We show below how to prove expressions of the form $(xx' = i) \to \phi'(x,x')$, which we then use to derive Φ'_i from Φ'_{i-1} in conjunction with $(xx' = i) \to \phi'(x,x')$. In particular we can derive $\Phi'_{i-1} \wedge ((xx' = i) \to \phi'(x,x'))$, equivalently $\forall xx'((xx' < i \to \phi'(x,x')) \wedge (xx' = i \to \phi'(x,x')))$, equivalently $\forall xx'((xx' < i \vee xx' = i) \to \phi'(x,x'))$, equivalently (via an equivalence shown in the full version of this paper), $\forall xx'(xx' \leq i \to \phi'(x,x'))$, which is the same as Φ'_i.

How to formally prove $(xx' = i) \to \phi'(x,x')$: For each gate g of C, in the order in which the functions f_g are defined, we can prove a line saying

$$(xx' = i) \to f_g(x) = j_g(x)$$

where $j_g(x) \in \{\text{TRUE}, \text{FALSE}\}$ is the appropriate propositional constant. This is done by using the extension-axiom line that defines f_g, with gate g's inputs. (If say g takes inputs from g' and g'', we use previous lines containing expressions $(xx' = i) \to f_{g'}(x) = j_{g'}(x)$, $(xx' = i) \to f_{g''}(x) = j_{g''}(x)$.)

Letting $g(1), \ldots, g(m)$ be the sequence of gates, listed in the order in which their functions $f_{g(1)}, \ldots, f_{g(m)}$ are defined, we have

$$(xx' = i) \to \bigwedge_{r \in [m]} (f_{g(r)}(x) = j_{g(r)}(x), f_{g(r)}(x') = j_{g(r)}(x'))$$

It then suffices to prove

$$\Big((xx' = i) \wedge \bigwedge_{r \in [m]} (f_{g(r)}(x) = j_{g(r)}(x), f_{g(r)}(x') = j_{g(r)}(x'))\Big) \to M$$

which is a line of type (15), and can be proved by the procedure of plugging in the constants $i, j_{g(r)}(x), j_{g(r)}(x')$ in place of the terms $x, x', f_{g(r)}(x), f_{g(r)}(x')$ in the way described below Equation (15). An error in the proof will correspond to this expression evaluating to FALSE, and getting treated as TRUE.

To conclude, note that we can construct a small circuit that on input $i \in [2^{2n}]$, outputs the above proof of $(xx' = i) \to \phi'(x,x')$. The circuit can be extended to a concise proof of (17). ◀

4 Reduction from PPP to Wrong Proof

In this section we establish the following result:

▶ **Theorem 4.** *Any problem that belongs to the complexity class* PPP *(which includes* PPAD *and* PPADS*) is reducible to* W*rong Proof (with respect to our deductive system* Q-EFF *of Sections 2.1, 2.2).*

The complexity class PPP is defined as the set of all problems reducible to the problem Pigeonhole Circuit, which is informally described as follows: suppose we are given a boolean circuit having n bits of input and output. Suppose that no input maps to the all-ones output. By the pigeonhole principle, there must be a *collision*, a pair of input vectors that map to the same output. The problem is to find a collision. Notice that this problem is in NP, since a collision is easy to verify, but finding one seems hard. We use the following definition of Pigeonhole Circuit.

▶ **Definition 5.** An instance of Pigeonhole Circuit consists of a circuit C having n input bits and n output bits. A solution consists of either a n-bit string that C maps to the all-zeroes string, or two n-bit strings that C maps to the same output string.

Proof. (of Theorem 4) We reduce from Pigeonhole Circuit to Wrong Proof. Given an instance C of Pigeonhole Circuit we need to construct (in time polynomial in the size of C) a circuit-generated proof Π_C (an exponentially-long, concisely-represented formal proof containing a known contradiction) whose error(s) allow us to find solution(s) to C.

Recall that n-bit strings correspond with numbers in $[2^n]$ (2^n being the all-zeroes string). We include in Π_C a function $f : [2^n] \to [2^n]$, which we construct using Q-EFF according to the first paragraph of Section 3.2. The (2^n into $2^n - 1$) pigeonhole principle assures us that

$$\exists(x, x')\Big((x \neq x' \wedge f(x) = f(x')) \vee f(x) = 2^n\Big) \tag{19}$$

Lemma 3 of Section 3.2 tells us how to generate a purported proof that (19) does not hold; the proof will be incorrect, but from error(s) in that proof we can efficiently recover satisfying assignments of $(x \neq x' \wedge f(x) = f(x')) \vee f(x) = 2^n$, which in turn identify solutions to the original Pigeonhole Circuit problem C.

So the challenge is to write down a (correct) circuit-generated proof of (19). Let T_C denote the formula of (19) (the "target" formula to be proved, for given C). The proof of (19) has a known line containing T_C, whose formal proof begins as follows.

Let $S_C = \exists x(f(x) = 2^n)$. Then using the case analysis rule (4), T_C is inferable from $S_C \to T_C$ and $\neg S_C \to T_C$. $S_C \to T_C$ is straightforward; note that it is of the form:

$$\exists x A(x) \to \exists x, y(A(x) \vee B(x, y))$$

In the full version of this paper we show how to prove this using Q-EFF.

So, the main challenge that remains is to generate a proof of

$$\neg S_C \to T_C. \tag{20}$$

We give the constructions of the formulae A_i, C_i, and F_i discussed in Section 3. Thus, A_i asserts a property of some instance i that implies, non-constructively, the existence of a solution. C_i is the explicit existential statement of a solution's existence. F_i is an extension axiom of the form (13), defining the construction of function f_{i-1} in terms of f_i.

For $i \in [2^n]$, $i \geq 2$, let A_i be the sentence

$$\forall x\Big(x \leq i \;\to\; f_i(x) \leq i - 1\Big). \tag{21}$$

A_i states that $f_i([i]) \subseteq [i-1]$ (which implies, non-constructively, that f_i has a collision in the range $[i-1]$).

For $i \in [2^n]$, $i \geq 2$, let C_i be the sentence

$$\exists x \neq x'\Big(x \leq i \;\wedge\; x' \leq i \;\wedge\; f_i(x) = f_i(x') \;\wedge\; f_i(x) \leq i - 1\Big). \tag{22}$$

C_i states *explicitly* that f_i has a collision in the range $[i-1]$, with the two colliding inputs in the range $[i]$. The pigeonhole principle tells us that C_i should follow from A_i, and we will achieve this (i.e derive C_i from A_i) using exponentially many steps of a circuit-generated proof.

We include a sequence of extension-axiom lines – of type (13) – as follows. For $i \in [2^n]$, $i \geq 2$, line $\ell(F_i)$ contains expression F_i defining function f_{i-1} in terms of f_i (see Figure 1). We also use a special line ℓ_F – also an extension-axiom line of type (13) – that sets f_{2^n} equal to f: formally, ℓ_F contains the expression $F := f_{2^n}(x) \leftrightarrow f(x)$. In the full version we show in detail how to prove A_{2^n} based on F together with $\neg S_C$. For $i \in [2^n]$, $i \geq 2$, F_i defines f_{i-1} as follows.

$$f_{i-1}(x) \leftrightarrow \begin{cases} i-2 & \text{if } \ x < i \wedge f_i(i) = i-1 \wedge f_i(x) = i-1 \\ f_i(i) & \text{if } \ x < i \wedge f_i(i) < i-1 \wedge f_i(x) = i-1 \\ f_i(x) & \text{otherwise. } \ (\text{i.e. } x \geq i \vee f_i(x) < i-1) \end{cases} \tag{23}$$

F_i states that f_{i-1} is derived from f_i as follows. f_{i-1} and f_i are intended to satisfy A_{i-1} and A_i respectively, and suppose we know that f_i satisfies A_i and want to construct f_{i-1} from f_i in such a way that f_{i-1} satisfies A_{i-1}. (23) ensures that for $x \in [i-1]$, $f_{i-1}(x) \in [i-2]$. If $x \in [i-1]$ is mapped by f_i to $i-1$, it is redirected to $i-2$ if $f_i(i) = i-1$, and if $f_i(i)$ is less than $i-1$, it is redirected to $f_i(i)$. *The construction is designed to allow us to reconstruct a collision for f_i based on an explicit statement of a collision for f_{i-1}.* For that, it does not work to just take inputs that f_i maps to $i-1$, and let f_{i-1} send them to $i-2$; the more complicated rule of (23) seems necessary. The construction is related to the one of [9], that also sets $f_{i-1}(x)$ to $f_i(i)$ whenever $f_i(x) = i-1$, but we have a different treatment of the case that $f_i(i) = i-1$, which allows us to recurse all the way down to $i = 2$.

We define a sequence of lines of Π_C as follows. For all $i \in [2^n]$, $i \geq 3$, we include lines $\ell(A_i)$ (each line number $\ell(A_i)$ and its contents are efficiently computable from i), such that $\ell(A_i)$ contains the expression:

$$\ell'(A_i), \ell''(A_i) \vdash (F_i \wedge A_i) \rightarrow A_{i-1}; \tag{24}$$

$\ell(A_i)$ states that if function f_{i-1} is constructed from f_i according to formula F_i, and f_i satisfies A_i, then f_{i-1} satisfies A_{i-1}. $\ell'(A_i)$ and $\ell''(A_i)$ contribute to a formal proof of the expression of $\ell(A_i)$; all these lines are distinct. In the full version we give details on how to do this, hence proving $f_i([i]) \subseteq [i-1]$ for all $i \geq 2$, by backwards induction starting at $f = f_{2^n}$. Given all these lines of type (24), together with a line we can derive containing $\neg S_C \rightarrow A_{2^n}$, and the lines containing F_i, we can infer a sequence of lines containing $\neg S_C \rightarrow A_i$, for all $i \geq 2$.

Π_C contains a special line $\ell(C_2)$, saying that if we have A_2, then C_2 can be proved. C_2 is the "obvious" statement that f_2, which maps both $x^{(1)}$ and $x^{(2)}$ to $x^{(1)}$, has a collision. Line $\ell(C_2)$ is of the form

$$\ell'(C_2) \vdash A_2 \rightarrow C_2; \tag{25}$$

for some other special line $\ell'(C_2)$ used in a single self-contained proof of (25). $\ell(C_2)$ states that C_2 can be deduced without any further assumptions about f_2. By construction, f_2 maps both $x^{(1)}$ and $x^{(2)}$ to $x^{(1)}$, so we know where to look for a collision! In the full version we give details on how to formally prove that f_2 has this "obvious" collision.

For $i \in [2^n]$, $i \geq 3$, we include lines $\ell(C_i)$, (again, these line numbers and the lines themselves are efficiently computable from i), where $\ell(C_i)$ contains the expression

$$\ell'(C_i), \ell''(C_i) \vdash (A_i \wedge F_i \wedge C_{i-1}) \rightarrow C_i; \tag{26}$$

$\ell(C_i)$ states that if C_{i-1} can be established, then given F_i and A_i we can deduce C_i (a collision for function f_i) where $\ell'(C_i)$ and $\ell''(C_i)$ are some further lines used in the proof

of (26). In the full version we give some more detail on how to construct a formal proof of (26) using Q-EFF.

Putting it all together, we noted earlier that we have a sequence of lines containing $\neg S_C \to A_i$, for $i \in [2^n], i \geq 2$. We also know that C_2 follows from A_2 (25). We may use these, along with the lines $\ell(F_i)$ that give us F_i, and the lines $\ell(C_i)$ (i.e. of the form (26)) to deduce (by repeated applications of modus ponens and conjunction introduction) $\neg S_C \to C_{2^n}$; using ℓ_F we get (20) as desired. This completes the construction of a formal proof according to the strategy outlined at the end of Section 3. ◀

5 PPA and PLS

The analogues of Theorem 4 for the classes PPA and PLS are stated below. We omit here the proofs, which are quite substantial but similar in spirit with that of Theorem 4, once appropriate adjustments have been made for the combinatorial idiosyncracies of these two classes. By the known inclusions, this covers all known syntactic subclasses of TFNP. The interested reader can see the details in the full version.

The analogue for PPA reduces from the problem Lonely as defined and shown PPA-complete in Beame et al. [1]; for PLS we make use of the problem Iter shown PLS-complete by Morioka [22] (Section 3.2).

▶ **Theorem 6.** *Any problem that belongs to the complexity class* PPA *is reducible to* W*rong Proof.*

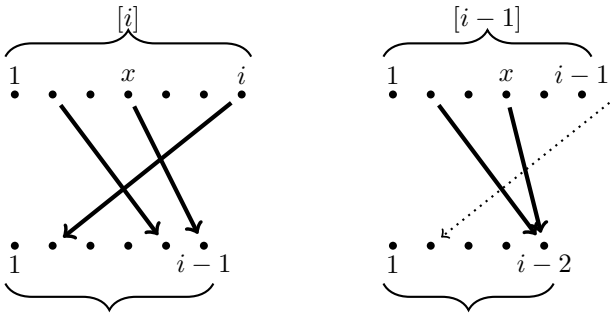
▶ **Theorem 7.** *Any problem that belongs to the complexity class* PLS *is reducible to* W*rong Proof.*

6 Finitary Existential Sentences and TFNP

To end on a different note, let us look back at the five classes: All five correspond to elementary combinatorial existence arguments (such as "every dag has a sink", recall the five bullets in the Introduction). Importantly, all five combinatorial existence arguments yielding complexity classes are *finitary:* They are true of finite structures and not true of all infinite structures. *Is this a coincidence?* Can there be an interesting complexity subclass of TFNP defined in terms of an existence argument that is not finitary, but is true of all structures, finite *and* infinite?

Seen as sentences in logic, these combinatorial arguments are statements of the form "for all finite structures (such as topologically sorted dags) there exists an element (a node) that satisfies a property (has no outgoing edges)." The corresponding logical expression is a sentence $\exists \bar{x} \Phi(\bar{x})$ in predicate logic, involving a set of existentially quantified variables $\bar{x}$ and an expression Φ with any number of other variables, as well as function symbols capturing structures such as undirected and directed graphs, pigeonhole functions, or total orders and potential functions. The "for all finite structures" quantification is implicit in the requirement that the sentence $\exists \bar{x} \Phi(\bar{x})$ be *valid on finite structures.*

And conversely, it is easy to see any such sentence yields a problem Find Witness$_\Phi$ in TFNP (and consequently a complexity class, through reductions). Find Witness$_\Phi$ is defined as follows: "Given a finite structure for Φ, where the finite universe can be assumed to be an initial segment of the nonnegative integers and the structures are presented implicitly through circuits computing the functions of Φ on elements of the universe encoded in binary, find a tuple $\hat{x}$ of integers that satisfy Φ."



$f_i([i]) = [i-1]$ $f_{i-1}([i-1]) = [i-2]$

"naive" choice of $f_{i-1}(x)$ for x such that $f_i(x) = i-1$, is to set $f_{i-1}(x)$ to be some fixed value in $[i-2]$ (here, $i-2$). We construct f_{i-1} as shown in examples below.

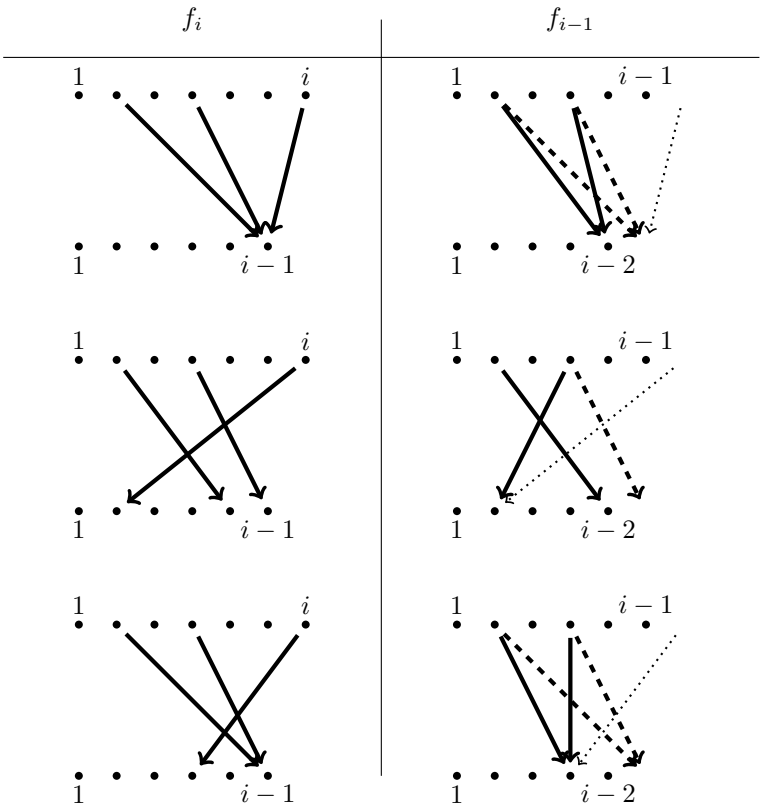


Figure 1 Construction of f_{i-1} from f_i (re proof of Theorem 4), such that from $f_i(x) < i$ for all $x \leq i$, we have $f_{i-1}(x) < i-1$ for all $x \leq i-1$. Dotted lines represent evaluations of f_{i-1} on i, and we are just interested in f_{i-1} on the domain $[i-1]$. Dashed lines are ones that have been "redirected" in construction of f_{i-1}.

The naive approach of setting f_i to some value less than i, may create collisions for f_{i-1} for which we can't reconstruct a collision for f_i based on a collision we found for f_{i-1}.

We can now formulate the question in the section's opening paragraph in logic terms: All five sentences Φ corresponding to the five known complexity subclasses of TFNP are of course true in any finite model, but all of them happen to be false for some infinite models (for example, "every dag has a sink" fails for the totally ordered integers). Is this necessary? *Can there be an interesting subclass of TFNP based on a* valid sentence $\exists\bar{x}\Phi(\bar{x})$*, that is, one that is true of* all *models, finite or infinite?*

Employing an ancient theorem in Logic due to Jacques Herbrand [13] (1930) one can show that the answer is negative:

▶ **Theorem 8.** *For any valid sentence in predicate logic of the form $\exists\bar{x}\Phi(\bar{x})$, the corresponding problem* Find Witness$_\Phi$ *can be solved in polynomial time.*

Sketch. Herbrand's theorem [13] states that any valid sentence

$$\exists x_1 \cdots \exists x_k \Phi(x_1, \ldots, x_k)$$

is equivalent to a finite disjunction of the form

$$\bigvee_{i=1}^{K} \Phi(t_{i1}, \ldots, t_{ik}),$$

where the t_{ij}'s are terms involving the function symbols and constants of Φ, for some fixed K depending on Φ. Solving Find Witness$_\Phi$ entails evaluating each of these K logical formulae of fixed size to identify the combination of terms, and thus ultimately elements of the universe computed in linear time (with respect to the length of the input) through the circuits of the input, that indeed satisfy Φ. ◀

7 Discussion

We have defined PTFNP, a subclass of the total function problems with NP verification of witnesses, which we see as a "syntactic" (in the sense of having complete problems) approximation of TFNP. We showed that PTFNP contains the five known classes PPP, PPA, PPAD, PPADS, and PLS.

Our understanding is that the present results are implicit in recent work in Bounded Arithmetic, but our system Q-EFF is of interest since it seems to allow more direct reductions. We also highlight the general topic of using more powerful logic to define more expressive versions of the Wrong Proof problem, and whether new TFNP problems can be defined as a result. With regard to the "rogue problems" such as Factoring and Ramsey, these have also been addressed in the Bounded Arithmetic literature, and similar results are obtainable for Ramsey [24, 15] and Factoring [20]. As noted in the Introduction, we are also working towards using our approach to placing these problems in PTFNP [11]. The topic of more powerful systems raises a general question of whether we reach a limit where no further TFNP problems can be expressed. This relates to the open problem in propositional proof complexity of whether there's a most powerful proof system, for which the answer is believed to be negative.

A related question is whether the reductions of the present paper should be modifiable to work with a *weaker* proof system than Q-EFF, and the work of Beckmann and Buss [2] indicates that this should be possible in principle, in particular that one should be able to use a system with extended Frege lines defining propositional variables rather than functions. With regard to the extended Frege function lines we use, it is tempting to try to define such

a function via its bit graph: instead of defining f, could we just introduce boolean variables $f(x)$ for each $x \in [2^n]$, and have separate (standard extended Frege) lines for each of them? This approach does not seem applicable, at least in a direct and straightforward way.

Acknowledgements. Many thanks to the "PPAD-like classes reading group" at the Simons Institute during the Fall 2015 program on Economics and Computation for many fascinating interactions, and to Sam Buss and Pavel Pudlák for an inspiring lunch conversation in November 2015. We also thank Arnold Beckmann and Sam Buss for helpful comments on earlier versions of this paper. Thanks also to the organisers of the 2015 "Algorithmic Perspective in Economics and Physics" research program at the Centre de Recerca Matemàtica (CRM), Barcelona, where this research was initiated.

References

1 Paul Beame, Stephen A. Cook, Jeff Edmonds, Russell Impagliazzo, and Toniann Pitassi. The relative complexity of NP search problems. *J. Comput. Syst. Sci.*, 57(1):3–19, 1998. doi:10.1006/jcss.1998.1575.

2 Arnold Beckmann and Sam Buss. The NP search problems of frege and extended frege proofs. *ACM Trans. Comput. Log.*, 18(2):11:1–11:19, 2017. doi:10.1145/3060145.

3 Sam Buss. *Bounded Arithmetic*. Bibliopolis, Naples, Italy, 1986. URL: www.math.ucsd.edu/~sbuss/ResearchWeb/BAthesis/.

4 Sam Buss. Quasipolynomial size proofs of the propositional pigeonhole principle. *Theor. Comput. Sci.*, 576:77–84, 2015. doi:10.1016/j.tcs.2015.02.005.

5 Samuel R. Buss. Polynomial size proofs of the propositional pigeonhole principle. *Journal of Symbolic Logic*, 52:916–927, 1987.

6 Samuel R. Buss. On Herbrand's Theorem. In *Logic and Computational Complexity*, volume 960 of *Lecture Notes in Computer Science*, pages 195–209. Springer, Berlin, Heidelberg, 1995. URL: www.math.ucsd.edu/~sbuss/ResearchWeb/herbrandtheorem/.

7 Samuel R. Buss and Alan S. Johnson. Propositional proofs and reductions between NP search problems. *Ann. Pure Appl. Logic*, 163(9):1163–1182, 2012. doi:10.1016/j.apal.2012.01.015.

8 Xi Chen, Xiaotie Deng, and Shang-Hua Teng. Settling the complexity of computing two-player nash equilibria. *J. ACM*, 56(3):14:1–14:57, 2009. doi:10.1145/1516512.1516516.

9 Stephen A. Cook and Robert A. Reckhow. The relative efficiency of propositional proof systems. *J. Symb. Log.*, 44(1):36–50, 1979.

10 Constantinos Daskalakis, Paul W. Goldberg, and Christos H. Papadimitriou. The complexity of computing a nash equilibrium. *SIAM J. Comput.*, 39(1):195–259, 2009. doi:10.1137/070699652.

11 Matthew Greaves. *Classifying the computational complexity of the Ramsey and Factoring Problems*. MSc dissertation, University of Oxford, 2017.

12 Juris Hartmanis and Lane A. Hemachandra. Complexity classes without machines: On complete languages for UP. *Theor. Comput. Sci.*, 58:129–142, 1988. doi:10.1016/0304-3975(88)90022-9.

13 Jacques Herbrand. *Recherches sur la théorie de la démonstration*. 1930. URL: http://eudml.org/doc/192791.

14 Pavel Hubácek, Moni Naor, and Eylon Yogev. The journey from NP to TFNP hardness. *Electronic Colloquium on Computational Complexity (ECCC)*, 23:199, 2016. URL: http://eccc.hpi-web.de/report/2016/199.

15 Emil Jeřábek. Approximate counting by hashing in bounded arithmetic. *The Journal of Symbolic Logic*, 74(3):829–860, 2009.

16 Emil Jerábek. Integer factoring and modular square roots. *J. Comput. Syst. Sci.*, 82(2):380–394, 2016. doi:10.1016/j.jcss.2015.08.001.

17 Ilan Komargodski, Moni Naor, and Eylon Yogev. White-box vs. black-box complexity of search problems: Ramsey and graph property testing. *Electronic Colloquium on Computational Complexity (ECCC)*, 24:15, 2017. URL: https://eccc.weizmann.ac.il/report/2017/015.

18 Jan Krajíček. *Bounded arithmetic, propositional logic, and complexity theory*, volume 60 of *Encyclopedia of Mathematics and Its Applications*. Cambridge University Press, 1995.

19 Jan Krajícek. Implicit proofs. *J. Symb. Log.*, 69(2):387–397, 2004. doi:10.2178/jsl/1082418532.

20 Jan Krajícek and Pavel Pudlák. Some consequences of cryptographical conjectures for S^1_2 and EF. *Inf. Comput.*, 140(1):82–94, 1998. doi:10.1006/inco.1997.2674.

21 Nimrod Megiddo. A note on the complexity of P-matrix LCP and computing an equilibrium. Technical Report RJ6439, IBM Almaden Research Center, San Jose, 1988.

22 Tsuyoshi Morioka. Classification of search problems and their definability in bounded arithmetic. *Electronic Colloquium on Computational Complexity (ECCC)*, 8(082), 2001. URL: http://eccc.hpi-web.de/eccc-reports/2001/TR01-082/index.html.

23 Christos H. Papadimitriou. On the complexity of the parity argument and other inefficient proofs of existence. *J. Comput. Syst. Sci.*, 48(3):498–532, 1994. doi:10.1016/S0022-0000(05)80063-7.

24 Pavel Pudlák. Ramsey's theorem in bounded arithmetic. In *Proceedings of the 4th Workshop on Computer Science Logic*, CSL '90, pages 308–317, London, UK, UK, 1991. Springer LNCS 553.

25 Pavel Pudlák. On the complexity of finding falsifying assignments for herbrand disjunctions. *Arch. Math. Log.*, 54(7-8):769–783, 2015. doi:10.1007/s00153-015-0439-6.

26 Ronald L. Rivest, Adi Shamir, and Leonard M. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Commun. ACM*, 21(2):120–126, 1978. doi:10.1145/359340.359342.

27 Michael Sipser. On relativization and the existence of complete sets. In *Proceedings of the 9th Colloquium on Automata, Languages and Programming*, pages 523–531, London, UK, 1982. Springer-Verlag.

Edge Estimation with Independent Set Oracles*

Paul Beame[†1], Sariel Har-Peled[‡2], Sivaramakrishnan Natarajan Ramamoorthy[§3], Cyrus Rashtchian[¶4], and Makrand Sinha[5]

1 **Paul G. Allen School of Computer Science & Engineering, University of Washington, Seattle, USA**
beame@cs.washington.edu
2 **Dept. of Computer Science, University of Illinois, Urbana-Champaign, USA**
sariel@illinois.edu
3 **Paul G. Allen School of Computer Science & Engineering, University of Washington, Seattle, USA**
sivanr@cs.washington.edu
4 **Paul G. Allen School of Computer Science & Engineering, University of Washington, Seattle, USA**
cyrash@cs.washington.edu
5 **Paul G. Allen School of Computer Science & Engineering, University of Washington, Seattle, USA**
makrand@cs.washington.edu

Abstract

We study the problem of estimating the number of edges in a graph with access to only an independent set oracle. Independent set queries draw motivation from group testing and have applications to the complexity of decision versus counting problems. We give two algorithms to estimate the number of edges in an n-vertex graph: one that uses only $\mathrm{polylog}(n)$ bipartite independent set queries, and another one that uses $n^{2/3} \cdot \mathrm{polylog}(n)$ independent set queries.

1998 ACM Subject Classification F.1.1 Models of Computation, F.2 Analysis of Algorithms and Problem Complexity

Keywords and phrases Approximate Counting, Independent Set Queries, Sparsification, Importance Sampling

Digital Object Identifier 10.4230/LIPIcs.ITCS.2018.38

1 Introduction

We study the problem of estimating the number of edges in a simple, unweighted, undirected graph $G = ([n], E)$, where $[n] := \{1, 2, \ldots, n\}$ and $m = |E|$, using only an oracle that answers independent set queries. For a parameter $\varepsilon > 0$, we wish to output an estimate $\widetilde{m}$ satisfying $(1-\varepsilon)m \leq \widetilde{m} \leq (1+\varepsilon)m$ with high probability. We consider randomized algorithms with access to one of the two following independent set oracles:

* A full version of the paper is available at [2], https://arxiv.org/abs/1711.07567
† Supported in part by the NSF under agreement CCF-1524246.
‡ Supported in part by NSF AF awards CCF-1421231 and CCF-1217462. Work done while visiting University of Washington on Sabbatical in 2017.
§ Supported by the NSF under agreements CCF-1149637, CCF-1420268 and CCF-1524251.
¶ Work partially completed while the author was at Microsoft Research.

Bipartite independent set (BIS) oracle: Given disjoint subsets $A, B \subseteq [n]$, a BIS query answers whether A, B satisfy $e(A, B) = 0$, where $e(A, B)$ denotes the number of edges with one endpoint in A and the other in B.

Independent set (IS) oracle: Given a subset $A \subseteq [n]$, an IS query answers whether A satisfies $e(A) = 0$, where $e(A)$ denotes the number of edges with both endpoints in A.

Previous work on graph parameter estimation has primarily focused on *local* queries, such as *degree* queries (which output the degree of a vertex v), *edge existence* queries (which answer whether a pair (u, v) forms an edge), or *neighbor* queries (which provide the i^{th} neighbor of a vertex v). However, such queries cannot achieve sub-polynomial query costs on certain lower bound graphs identified by Feige [13] and Goldreich and Ron [15], essentially due to the fact that these queries can only obtain *local* information about the graph. This motivates an investigation of other types of queries that may enable very efficient parameter estimation. The independent set queries described above naturally generalize an edge existence query, and their non-locality opens the door for sub-polynomial query algorithms for various graph parameter estimation tasks.

1.1 Motivation and Related Work

The most relevant previous work comes from the area of sub-linear time algorithms for graph parameter estimation. BIS and IS queries also have interesting connections to the classical area of group testing, to emptiness versus counting questions in computational geometry, and to the complexity of decision versus counting problems.

Graph Parameter Estimation. Many researchers have studied the problem of estimating various parameters of graphs using many types of queries. Feige [13] estimated the number of edges in a graph using degree queries, where a degree query returns the degree $\deg(v)$ of a specified vertex v in $G([n], E)$. In a follow-up work, Goldreich and Ron [15] estimated the number of edges in a graph using both degree and neighbor queries, where a neighbor query returns the j^{th} neighbor of a vertex v for $j, v \in [n]$. Related work has also appeared on estimating the number of stars [16], the minimum vertex cover [18], the number of triangles [10, 20], and the number of k-cliques [9]. A special case of BIS query termed a *group* query (where one of the bipartition sets is a singleton) was considered for testing k-colorability of graphs [3] and edge estimation [24].

The results of Feige [13] and Goldreich and Ron [15] on estimating the number of edges in a graph are quite relevant to our work. Feige [13] showed how to use $O\left(\sqrt{n}/\varepsilon\right)$ degree queries to output an estimate $\widetilde{m}$ that satisfies $(2-\varepsilon)m \leq \widetilde{m} \leq (2+\varepsilon)m$. Moreover, he showed that any algorithm achieving better than a 2-approximation must use a nearly linear number of queries in the worst case. Goldreich and Ron [15] showed that by using both degree and neighbor queries, the approximation could be improved to $(1-\varepsilon)m \leq \widetilde{m} \leq (1+\varepsilon)m$ by using $\sqrt{n} \cdot \text{poly}(\log n, 1/\varepsilon)$ queries. Finally, we mention that Feige [13] and Goldreich and Ron [15] have identified certain hard instances showing that these upper bounds cannot be improved, up to polylog(n) factors.

Group Testing. A classic estimation problem involves efficiently approximating the number of defective items or infected individuals in a certain collection or population [5, 7, 23]. To query a population, a small group is formed, and all the individuals in the group are tested in one shot. For example, in genome-wide association studies, combined pools of DNA may be tested as a group for certain variants [17]. In group testing, the result of a test often

indicates only whether there is at least one infected or defective unit, or if there is none. Such a dichotomous outcome resembles the independent set queries that we study. Group testing in the graph setting suggests the interpretation that we wish to test pairwise interactions between items or individuals, instead of singular events.

Computational Geometry. Certain geometric applications exhibit the phenomena that emptiness queries have more efficient algorithms than counting queries. For example, in three dimensions, for a set P of n points, half-space counting queries (i.e., what is the size of the set $|P \cap h|$, for a query half-space h), can be answered in $O(n^{2/3})$ time, after near-linear time preprocessing. On the other hand, emptiness queries (i.e., is the set $P \cap h$ empty?) can be answered in $O(\log n)$ time. Aronov and Har-Peled [1] used this to show how to answer approximate counting queries (i.e., estimating $|P \cap h|$), with polylogarithmic emptiness queries.

As another geometric example, consider the task of counting edges in disk intersection graphs using GPUs [14]. For these graphs, IS queries decide if a subset of the disks have any intersection (this can be done using sweeping in $O(n \log n)$ time [4]). Using a GPU, one could quickly draw the disks and check if the sets share a common pixel. In cases like this – when IS and BIS oracles have fast implementations – algorithms exploiting independent set queries may be useful.

Decision versus Counting Complexity. A generalization of IS and BIS queries has previously appeared in a line of work investigating the relationship between decision and counting problems [21, 22, 6]. Stockmeyer [21, 22] showed how to estimate the number of satisfying assignments for a given circuit with queries to an NP oracle. Ron and Tsur [19] observed that Stockmeyer implicitly provided an algorithm for estimating set cardinality using *subset* queries, where a subset query specifies a subset $X \subseteq \mathcal{U}$ and answers whether $|X \cap S| = 0$ or not. Subset queries generalize IS and BIS queries because S corresponds to the set of edges in the graph with $|S|$ and X is an arbitrary subset of pairs of vertices.

In what follows, we consider subset queries in the context of edge estimation and fix $|S| = m$ and $|\mathcal{U}| = \binom{n}{2}$. Stockmeyer provided an algorithm using $O(\log \log m \cdot \text{poly}(1/\varepsilon))$ subset queries to estimate m within a factor of $(1 + \varepsilon)$ with a constant success probability. Note that for a high probability bound, which is what we focus on in this paper, the algorithm would naively require $O(\log n \cdot \log \log m \cdot \text{poly}(1/\varepsilon))$ queries to achieve success probability at least $1 - 1/n$. Falahatgar, Jafarpour, Orlitsky, Pichapati, and Suresh [12] gave an improved algorithm that estimates m up to a factor of $(1+\varepsilon)$ with probability $1-\delta$ using $2 \log \log m + O\big((1/\varepsilon^2) \log(1/\delta)\big)$ subset queries. Nearly matching lower bounds are also known for subset queries [21, 22, 19, 12]. Ron and Tsur [19] also study a restriction of subset queries, called *interval queries*, where the universe $\mathcal{U}$ is ordered and the subsets are intervals of elements. We view the independent set queries as another natural restriction of subset queries.

Analogous to Stockmeyer's results, a recent work of Dell and Lapinskas [6] provides a framework that relates edge estimation using BIS and edge existence queries to a question in fine-grained complexity. They study the relationship between decision and counting versions of problems such as 3SUM and Orthogonal Vectors. We first describe their edge estimation result and then explain their connection to fine-grained complexity. They prove the following for bipartite graphs.

▶ **Theorem 1** ([6]). *For every $0 \leq \varepsilon \leq 1$, there exists an algorithm using $\frac{O(\log^6 n)}{\varepsilon^2}$ BIS queries and $\frac{n \cdot \text{polylog}(n)}{\varepsilon^4}$ edge existence queries, and outputs $\widetilde{m}$ satisfying $(1-\varepsilon)m \leq \widetilde{m} \leq (1+\varepsilon)m$ with probability at least $1 - 1/n^2$.*

Dell and Lapinskas [6] used their edge estimation algorithm to obtain approximate counting algorithms for problems in fine-grained complexity. For instance, given an algorithm for 3SUM with runtime T, they obtain an algorithm that estimates the number of YES instances of 3SUM with runtime $T \cdot \frac{O(\log^6 n)}{\varepsilon^2} + \frac{n \cdot \text{polylog}(n)}{\varepsilon^4}$. The relationship is quite simple and natural. The decision version of 3SUM corresponds to checking if there is at least one edge in a certain bipartite graph. The counting version then corresponds to counting the edges in this graph. We note that in their application, the large number $O(n \cdot \text{polylog}(n))$ of edge existence queries does not affect the dominating term in the overall time in their reduction; the larger term in the time is a product of the time to decide 3SUM and the number of BIS queries.

1.2 Our Results

We present two new algorithms. In what follows, let $G = ([n], E)$ be a simple, unweighted graph with $|E| = m$ edges. Our first algorithm, using BIS queries, gives the following.

▶ **Theorem 2.** *Given $n \geq 16$ and $\frac{36 \log n}{\sqrt{m}} \leq \varepsilon \leq \frac{1}{2}$, there exists an algorithm that makes $\frac{\text{polylog}(n)}{\varepsilon^4}$ BIS queries and outputs $\widetilde{m}$ satisfying $(1-\varepsilon)m \leq \widetilde{m} \leq (1+\varepsilon)m$ with probability at least $1 - \frac{\text{polylog}(n)}{n^2}$.*

We remark that since $\text{polylog}(n)$ BIS queries can simulate a degree query (see the full version [2] for a proof) one obtains a $(2 \pm \varepsilon)$-factor approximation of m by using Feige's algorithm [13], which uses degree queries. This algorithm uses $O(\sqrt{n} \cdot \text{polylog}(n)/\text{poly}(\varepsilon))$ BIS queries. Theorem 2, however, provides better guarantees in terms of the approximation and the number of BIS queries.

Compared to the result of Dell and Lapinskas [6] (Theorem 1), our algorithm uses a significantly fewer number of queries, since we do not have to make $n \cdot \text{polylog}(n)$ edge existence queries. In terms of their specific applications, it does not seem that our improvement implies anything significant for fine-grained complexity. It would be interesting to find problems where a more efficient BIS estimation algorithm would lead to better decision versus counting complexity results.

Our second algorithm, using only IS queries, gives the following.

▶ **Theorem 3.** *Given $n \geq 8$ and $\frac{324 \log^4 n}{\sqrt{m}} \leq \varepsilon \leq \frac{1}{2}$, there exists an algorithm that makes $\min\left\{\frac{n^2}{\varepsilon^2 m}, \frac{\sqrt{m}}{\varepsilon}\right\} \cdot \text{polylog}(n)$ IS queries and outputs $\widetilde{m}$ satisfying $(1-\varepsilon)m \leq \widetilde{m} \leq (1+\varepsilon)m$ with probability at least $1 - \frac{1}{n^2}$.*

The first term $\frac{n^2}{\varepsilon^2 m} \cdot \text{polylog}(n)$ comes from a folklore algorithm that estimates the number of edges using edge existence queries (see for a proof). The second term $\frac{\sqrt{m}}{\varepsilon} \cdot \text{polylog}(n)$ is the number of queries used by our new algorithm.

Since $\min\left\{\frac{n^2}{\varepsilon^2 m}, \frac{\sqrt{m}}{\varepsilon}\right\} \leq \frac{n^{2/3}}{\varepsilon^{4/3}}$, we also get the following corollary.

▶ **Corollary 4.** *Given $n \geq 8$ and $\frac{324 \log^4 n}{\sqrt{m}} \leq \varepsilon \leq \frac{1}{2}$, there is an algorithm that makes $\frac{n^{2/3}}{\varepsilon^{4/3}} \cdot \text{polylog}(n)$ IS queries and outputs $\widetilde{m}$ satisfying $(1-\varepsilon)m \leq \widetilde{m} \leq (1+\varepsilon)m$ such that with probability at least $1 - \frac{1}{n^2}$.*

Comparing the above theorems, we observe that, perhaps surprisingly, BIS queries are much more effective for estimating the number of edges than IS queries.

Table 1 Comparison of the best known algorithms using a variety of queries for estimating the number of edges m in a graph with n vertices. The bounds stated are for high probability results, with error probability at most $1/n$. Constant factors are suppressed for readability.

Query Types	Approximation	# Queries (up to const. factors)	Reference
Edge Existence	$1+\varepsilon$	$\frac{n^2}{m} \cdot \text{poly}(\log n, 1/\varepsilon)$	Folklore (see [2])
Degree	$2+\varepsilon$	$\frac{\sqrt{n}}{\varepsilon} \cdot \log n$	[13]
Degree + Neighbor	$1+\varepsilon$	$\sqrt{n} \cdot \text{poly}(\log n, 1/\varepsilon)$	[15]
Subset	$1+\varepsilon$	$\log n \cdot \text{poly}(1/\varepsilon)$	[22, 12]
BIS + Edge Existence	$1+\varepsilon$	$n \cdot \text{poly}(\log n, 1/\varepsilon)$	[6]
BIS	$1+\varepsilon$	$\text{poly}(\log n, 1/\varepsilon)$	This Work
IS	$1+\varepsilon$	$\min\left\{\sqrt{m}, \frac{n^2}{m}\right\} \cdot \text{poly}(\log n, 1/\varepsilon)$	This Work

1.2.1 Comparison with Other Queries.

Table 1 quantitatively summarizes the results for estimating the number of edges in a graph in the context of various query types. Given some of the results in Table 1 on edge estimation using other types of queries, a natural question is how well BIS and IS queries can simulate such queries. In the full version [2] of the paper, we show that polylog(n) BIS queries are sufficient to simulate degree queries. On the other hand, we do not know how to simulate a neighbor query (to find a specific neighbor) with few BIS queries, but a random neighbor of a vertex can be found with $O(\log n)$ BIS queries (see [3]). For IS queries, it turns out that estimating the degree of a vertex v up to a constant factor requires at least $\Omega\left(\frac{n}{\deg(v)}\right)$ IS queries (we expand on this in the full version of the paper).

Notation. Throughout this text, log and ln will denote the logarithm taken in base two and e, respectively. For a positive integer k, the set $\{1, \ldots, k\}$ will be denoted by $[k]$. The notation $x = \text{polylog}(n)$ means $x = O(\log^c n)$ for some constant $c > 0$. When we say $A_1, \ldots, A_k$ is a *partition* of the set A into k parts we allow A_i to be empty. In particular, a uniformly random partition of A into k parts is chosen by coloring each element of A with a random number in $[k]$ and identifying A_i with elements colored i.

2 Overview of the Algorithms

We describe our algorithms using BIS and IS queries separately.

2.1 BIS Algorithm

Our discussion of the BIS algorithm will parallel Figure 1, which depicts the main components of one level of our recursive algorithm.

Our algorithms rely on the ability to exactly count the edges between two subsets of vertices, in time nearly linear in the number of such edges. In particular, we provide a simple,

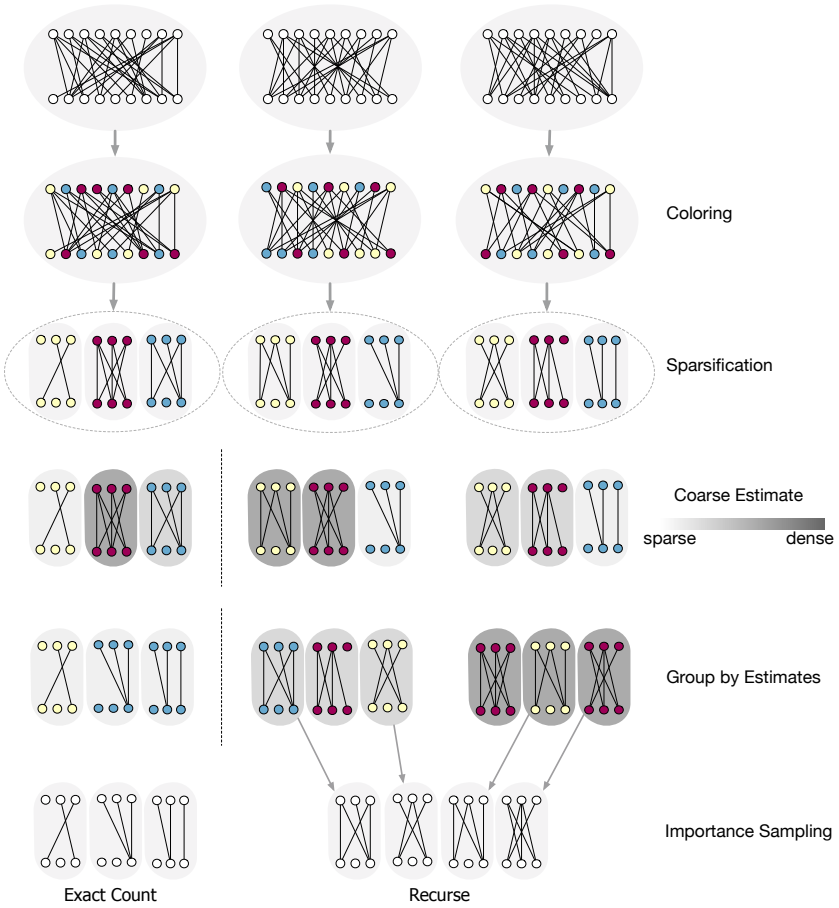


Figure 1 A depiction of one level of the BIS algorithm. In the first step, we color the vertices and sparsify the graph by only looking at the edges between vertices of the same color. In the second step, we coarsely estimate the number of edges in each colored subgraph. Next, we group these subgraphs based on their coarse estimates, and we subsample from the groups with a relatively large number of edges. In the final step, we exactly count the edges in the sparse subgraphs, and we recurse on the dense subgraphs.

deterministic divide and conquer algorithm to determine $e(A, B)$ using $O(e(A, B) \log n)$ BIS queries. More concretely, we will prove the following in the next section.

▶ **Lemma 5.** *For disjoint $A, B \subseteq [n]$, there is a deterministic algorithm that exactly computes $e(A, B)$ using at least $\frac{e(A,B)+1}{2}$ and at most $5 \cdot e(A, B)\lceil \log n \rceil + 1$ BIS queries.*

Given this algorithm, it is natural to wonder if the graph could be sparsified in such a way that the number of remaining edges is a good estimate for the original number of edges (after scaling). Consider sparsifying the graph by coloring the vertices of graph and only looking at the edges going between certain pairs of color classes (in our algorithm, these pairs will be a matching of the color classes). We prove that it suffices to only count the edges between the color classes, and we can ignore the edges with both endpoints inside a single color class.

▶ **Lemma 6** (Basic Sparsification). *Let G be an n-vertex graph with m edges. For k such that $1 \leq k \leq \lfloor n/2 \rfloor$, let $A_1, \ldots, A_k, B_1, \ldots, B_k$ be a uniformly random partition of $[n]$. Then,*

$$\mathbb{P}\left[\left| 2k \sum_{i=1}^{k} e(A_i, B_i) - m \right| \geq 18k \cdot \sqrt{m} \log n \right] \leq \frac{1}{n^4}.$$

An important consequence of this lemma is that we can assume without loss of generality that the graph is bipartite. Indeed, invoking the lemma with $k = 2$, estimating the edges between the two color classes is equivalent to estimating the total number of edges, up to a factor of two. In what follows, we consider colorings that respect the bipartition.

After coloring the graph, we have reduced the problem to estimating the total number of edges in a collection of bipartite subgraphs. However, certain subgraphs may still have a large number of edges, and it would be too expensive to directly use the exact counting algorithm. To remedy this, we develop an algorithm that coarsely estimates the number of edges in a subgraph, up a $O(\log^2 n)$ factor, using only $O(\log^3 n)$ BIS queries.

Using the coarse estimates we can form $O(\log n)$ groups of bipartite subgraphs, where each group contains subgraphs with a comparable number of edges. For the groups with only a polylogarithmic number of edges, we can exactly count edges using polylog(n) BIS queries. For the remaining groups, we subsample a polylogarithmic number of subgraphs from each group. Since the groups contained subgraphs with a similar number of edges, the number of edges in the subsampled subgraphs will be proportional to the total number of edges in the group, up to a scaling factor depending on the group, with high probability. This corresponds to the technique of *importance sampling* that is used for variance reduction when estimating a sum of random variables that have comparable magnitudes.

After exactly counting the edges in the sparse subgraphs, we are left with the task of estimating the number of edges in the subsampled subgraphs. By the sparsification guarantee, the number of edges in these subgraphs has gone down by a factor of k (which will be a constant) with high probability. We now recurse on the collection of subsampled subgraphs. Since the number of edges has gone down by a constant fraction, we only need to repeat this process a logarithmic number of times. Overall, at every level of the recursion, we work with a polylogarithmic number of subgraphs, and hence, we only make a polylogarithmic number of BIS queries in total.

Note that we have to scale the estimates after the sparsification step and the subsampling step. We handle this in our algorithm by associating each subgraph with a weight, and outputting the weighted sum of the estimates of the subgraphs. The scaling factors after sparsification and subsampling are reflected by updating the weights appropriately. These weights are also used when grouping subgraphs based on the coarse estimates. Overall, the final estimate corresponds to the weighted sum of the estimates of the subgraphs. Figure 1 depicts the main components of one level of our algorithm.

We now describe the algorithms for exact counting and coarse estimation in more detail.

2.1.1 Exact Counting

Let A, B be disjoint subsets of vertices. We explain how to use BIS queries to compute $e(A, B)$, the number of edges between A and B. We use a divide and conquer approach. Let A_1, A_2 and B_1, B_2 be equipartitions of A and B, respectively. Observe that $e(A, B) = e(A_1, B_1) + e(A_1, B_2) + e(A_2, B_1) + e(A_2, B_2)$. For any pair (A_i, B_j) with no edges, we determine $e(A_i, B_j) = 0$ with one BIS query. Otherwise, we recursively determine $e(A_i, B_j)$.

We build a quadtree starting with (A, B) as the root. If $|A| = |B| = 1$, then we query this pair directly and label it with the value of $e(A, B)$, which is 0 or 1 in this case. Otherwise,

we still query the pair (A, B), and if $e(A, B) = 0$, label the node as 0 and terminate. In the remaining case, we know that $e(A, B) \neq 0$, and the algorithm will recurse on the four children of (A, B), which will correspond to the pairs $(A_1, B_1), (A_1, B_2), (A_2, B_1)$, and (A_2, B_2).

To determine $e(A, B)$, the algorithm simply needs to sum the labels of all the leaves in this tree. The number of queries is equal to total number of nodes in the tree. One can prove that the number of nodes is at most $O(e(A, B) \cdot \log n) + 1$. The intuition is that the number of leaves labeled with a 1 is exactly $e(A, B)$, and the number labeled with a zero is at most $O(e(A, B) \cdot \log n) + 1$.

2.1.2 Coarse Estimator

We explain how to estimate $e(A, B)$ up to a factor of $O(\log^2 n)$ using $O(\log^3 n)$ BIS queries. As a warm-up, assume all vertices in A have degree in the range $(2^i, 2^{i+1}]$ for some $i > 0$. Sample a subset $A' \subseteq A$ by including every vertex in A independently with probability $2^i/\widetilde{e}$ for an estimate $\widetilde{e}$ we will determine later. Then, sample $B' \subseteq B$ by including every vertex in B independently with probability $1/2^i$. When $\widetilde{e} \approx e(A, B)$, there will be an edge between A' and B' with good probability, and when $\widetilde{e} \gg e(A, B) \cdot \log(n)$, then there will be no such edge with good probability. Therefore, we can perform a geometric search for $\widetilde{e} \approx e(A, B)$ to determine $e(A, B)$.

Now, we are left with the task of finding a value i^* such that a good fraction of all the edges in the graph are incident on the vertices in A that have degree roughly 2^{i^*}. By grouping vertices of degree $(2^i, 2^{i+1}]$ together, the pigeonhole principle guarantees that there is an $i^* \in \{0, \ldots, \log n\}$ such that the edges touching the vertices of degree roughly 2^{i^*} is $1/\log n$ fraction of all the edges. To find the value of i^*, we perform a geometric search over the possible values $i = 1, 2, 4, 8, \ldots, 2 \log n$.

The two sources of error thus come from the estimate of i^* and the acceptance probability based on the estimate for $\widetilde{e}$. Since each contributes a factor of $O(\log n)$ to the error, the coarse estimate for $e(A, B)$ will have the following guarantee.

▶ **Lemma 7.** *Let $n \geq 16$. For disjoint $A, B \subseteq [n]$, the algorithm* $\mathsf{CoarseEstimator}(A, B)$ *that makes $c_{\mathsf{ce}} \log^3 n$ BIS queries (for a constant c_{ce}) and outputs $\widetilde{e} \leq n^2$ such that with probability at least $1 - \frac{4 \log n}{n^4}$, it holds that $\frac{e(A,B)}{8 \log n} \leq \widetilde{e} \leq e(A, B) \cdot 8 \log n$.*

2.2 IS Algorithm

As with the BIS algorithm, the main building block for the IS algorithm is an efficient way to exactly count edges using IS queries. The strategy for the BIS exact counting algorithm fails because the total number of nodes in the tree can be much large than $e(A \cup B)$, up to $O(n^2)$ in the worse case. However, if we pick the sets A_1, A_2 at random, then the overall number of queries will be small with high probability. Thus, this randomized modification of the BIS exact counting algorithm computes $e(A, B)$ using $O(e(A \cup B) \log^2 n)$ IS queries with high probability. In particular, we prove the following result.

▶ **Lemma 8.** *For every disjoint $A, B \subseteq [n]$, there is a randomized algorithm that exactly computes $e(A, B)$ using at least $\frac{e(A,B)+1}{2}$ and at most $c \cdot e(A \cup B) \lceil \log^2 n \rceil + 1$ IS queries for a constant c, with probability at least $1 - \frac{1}{n^6}$.*

With this lemma in hand, we will again sparsify the graph to reduce the overall number of IS queries. In contrast to the BIS queries, we do not know how to design a coarse estimator using only IS queries. This prohibits us from designing an analogous recursive algorithm. Instead, we estimate the number of edges in one shot, by coloring the graph with a large

number of colors and estimating the number of edges going between a matching of the color classes. We now discuss the reasons behind using a matching of the color classes.

An initial sparsification attempt might be to count only the edges going between a single pair of colors. If the total number of colors is $2k$, then we expect to see $m/\binom{2k}{2}$ edges between this pair. Therefore, we could set k to be large and invoke Lemma 8. Scaling by a factor of $\binom{2k}{2}$, we would hope to get an *unbiased* estimator for m.

Unfortunately, a star graph demonstrates that this approach fails, due to large variance. If we randomly color the vertices of the star graph with $2k$ colors, then out of the $\binom{2k}{2}$ pairs of color classes, only $2k-1$ pairs have any edge going across. So, if we only chose one pair of color classes, then with high probability one of the following two cases occurs: either (i) there is no edge crossing the color pair, or (ii) the number of edges crossing the pair is $\approx m/2k$. In both cases, the estimate after scaling by a factor of $\binom{2k}{2}$ is far from the truth.

At the other extreme, the vast majority of edges will be present if we look at the edges crossing *all pairs* of color classes. Indeed, the only edges we miss have both endpoints in a color class, and this accounts for only a $1/k$ fraction of the total number of edges. Thus, this does not achieve any substantial sparsification.

By using a matching of the color classes, we simultaneously get a reliable estimate of the number of edges and a sufficiently sparsified graph, as already testified by Lemma 6. Let $A_1, \cdots, A_k, B_1, \cdots, B_k$ be a random partition of the vertices into $2k$ color classes. Lemma 6 implies that the estimator $2k \sum_{i=1}^{k} e(A_i, B_i)$ is in the range $m \pm O(k\sqrt{m}\log n)$ with high probability. Hence, when k to is less than $\varepsilon\sqrt{m}/\mathrm{polylog}(n)$, we approximate m up to a factor of $(1 \pm O(\varepsilon))$. We use a geometric search to find such a k efficiently.

To bound on the number of IS queries, we claim that we can compute $\sum_{i=1}^{k} e(A_i, B_i)$ using Lemma 8, with a total of $\left(k + \frac{m}{k}\right) \cdot \mathrm{polylog}(n)$ IS queries. The first term arises since we use one query for each of the k color pairs (even if there are no edges between them). For the second term, we pay for both (i) the edges between the color classes and (ii) the total number of edges with both endpoints within a color class (since the number of IS queries in Lemma 8 scales with $e(A \cup B)$). By the sparsification lemma, we know that (i) is bounded by $O(\frac{m}{k})$ with high probability and we can prove an analogous statement for (ii). Hence, plugging in a $k \approx \frac{\varepsilon\sqrt{m}}{\mathrm{polylog}(n)}$, the total number of IS queries is bounded by $\sqrt{m} \cdot \mathrm{polylog}(n)/\varepsilon$.

2.3 Outline

The rest of the paper is organized as follows. In Section 3, we formally present the algorithm to exactly count edges between two subsets of vertices using BIS queries (Lemma 5). In Section 4, we prove our sparsification result (Lemma 10). In Section 5.1, we present the algorithm that uses BIS queries to coarsely estimate the number of edges between two subsets of vertices (Lemma 7). We combine these building blocks to construct our edge estimation algorithm using BIS queries in Section 5.2. In Section 6, we present our algorithm using IS queries. We conclude in Section 7 and mention open questions.

3 Exact Edge Counting using BIS and IS Queries

In this section, we prove Lemma 5. We give a deterministic algorithm that builds a tree by repeatedly partitioning A and B using BIS queries. The leaves of this tree identify the edges we want to count and the number of BIS queries made to construct this tree is $O(e(A, B)\log n)$. We prove Lemma 8 in the full version [2] of the paper where we give a similar randomized algorithm that uses random partitioning and makes $O(e(A \cup B)\log^2 n)$ IS queries. To present the proof, we will need the following definition.

▶ **Definition 9.** An *equipartition* of a set A with $|A| \geq 2$ is a partition A_1, A_2 of A satisfying $|A_1| = \lceil |A|/2 \rceil$ and $|A_2| = \lfloor |A|/2 \rfloor$. A *random equipartition* is an equipartition chosen uniformly at random from all equipartitions.

3.1 Proof of Lemma 5

We construct a rooted tree where every vertex is identified with a pair (A', B'), where $A' \subseteq A$ and $B' \subseteq B$, and the root of the tree is the input pair (A, B). The leaves of the tree will correspond to (A', B') with either $|A'| = |B'| = 1$ or $e(A', B') = 0$. These leaves will be labeled with 1 if $e(A', B') > 1$ and 0 otherwise. The tree has the property that every edge contributing to $e(A, B)$ appears as a leaf, and there will be exactly $e(A, B)$ leaves labeled 1. Thus, the leaves and their labels suffice to compute $e(A, B)$. Finally, this tree can be built and labeled using BIS queries.

We now proceed to describe the tree, which is constructed recursively by the following deterministic process. If $e(A, B) = 0$, then assign a 0 to the root and terminate. Otherwise, let (A', B') be the current internal node. Then

- if $e(A', B') \geq 1$, $|A'| > 1$ and $|B'| > 1$, let A'_1, A'_2 (resp. B'_1, B'_2) be an equipartition of A' (resp. B'). Add the nodes $(A'_1, B'_1), (A'_1, B'_2), (A'_2, B'_1)$ and (A'_2, B'_2) as the children.
- if $e(A', B') \geq 1$, $|A'| = 1$ and $|B'| > 1$, let B'_1, B'_2 be any equipartition of B'. Add the nodes (A', B'_1) and (A', B'_2) as the children.
- if $e(A', B') \geq 1$, $|A'| > 1$ and $|B'| = 1$, let A'_1, A'_2 be any equipartition of A'. Add the nodes (A'_1, B') and (A'_2, B') as the children.

We claim that this process uses at most $5e(A, B)\lceil \log n \rceil + 1$ BIS queries. First, note that since we make one query for each node, we simply have to bound the number of nodes. We argue that each internal node of this tree lies on a path from the root to a leaf with value 1. Indeed, if (A', B') is an internal node, then $e(A', B') \geq 1$, and hence some leaf in the sub-tree rooted at (A', B') is labeled 1.

Notice that the depth of the tree is $\lceil \log n \rceil$ and the number of leaves with value 1 is $e(A, B)$. This implies that the total number of internal nodes is at most $e(A, B)(\lceil \log n \rceil - 1)$. The number of leaves that are assigned a 0 is at most $4e(A, B)\lceil \log n \rceil + 1$ (since each node has at most 4 children). Therefore, the total number of nodes in the tree is at most $e(A, B)\lceil \log n \rceil + 4e(A, B)\lceil \log n \rceil + 1$, in turn implying that the total number of BIS queries made is at most $5e(A, B)\lceil \log n \rceil + 1$.

The number of BIS queries made is always at least $\max\{e(A, B), 1\} \geq \frac{e(A,B)+1}{2}$ since every edge with one endpoint in A and the other in B is identified.

4 Sparsification by Coloring

We present and prove our sparsification lemma. For technical reasons, we need a slightly more general sparsification statement than the one (Lemma 6) described in Section 2.

▶ **Lemma 10.** *Let $G = ([n], E)$ be a graph with m edges. For any $1 \leq k \leq \lfloor n/2 \rfloor$, let $A_1, \ldots, A_{2k}$ be a uniformly random partition of $[n]$. Then,*

(a) $\mathbb{P}\left[\left| \frac{m}{2k} - \sum_{i=1}^{k} e(A_i, A_{k+i}) \right| \geq 9\sqrt{m} \log n \right] \leq \frac{1}{n^4}$

(b) $\mathbb{P}\left[\left| \frac{m}{2k} - \sum_{i=1}^{2k} e(A_i) \right| \geq 9\sqrt{m} \log n \right] \leq \frac{1}{n^4}.$

Furthermore, for disjoint sets $A, B \subseteq [n]$ and $2 \le k \le \max\{|A|, |B|\}$, let $A_1, \ldots, A_k$ and $B_1, \ldots, B_k$ be uniformly random partitions of A and B, respectively. Then,

(c) $$\mathbb{P}\left[\left| \frac{e(A,B)}{k} - \sum_{i=1}^{k} e(A_i, B_i) \right| \ge 9\sqrt{e(A,B)} \log n \right] \le \frac{1}{n^4}$$

Proof.

(a) Consider the random process that colors vertex t at step $t \in [n]$ with a uniformly random color $X_t \in [2k]$. The colors correspond to the partition of $[n]$ into classes $A_1, \ldots, A_{2k}$. Define $f(X_1, \ldots, X_n) = \sum_{i=1}^{k} e(A_i, A_{k+i})$, and notice that $\mathbb{E}[f] = m/(2k)$ and that $0 \le f(X_1, \ldots, X_n) \le m$.

When the vertex t is colored, let $N_{i,t}$ be the number of neighbors of t colored with color i among the first $t-1$ vertices. Now, define $d_t = \sum_{i \in [2k]} N_{i,t}$ to be the total number of colored neighbors of vertex t. Observe that d_t is deterministic and that $\sum_{t=1}^{n} d_t = m$ holds, since d_t is the number of edges between the vertex t and the vertices in $[t-1]$. Notice that $\mathbb{E}[N_{i,t}] = d_t/(2k)$. For $i \in [2k]$, let $\mathcal{E}_{i,t}$ be the event that

$$\left| N_{i,t} - \frac{d_t}{2k} \right| \le 2\sqrt{d_t \ln n}. \tag{1}$$

Note that when $d_t = 0$, the event $\mathcal{E}_{i,t}$ holds with probability 1 and when $\deg(t) \ge 1$, applying Lemma 17(a) (with $r = d_t$, $\mu = \frac{d_t}{2k}$ and $s = 2\sqrt{d_t \ln n}$) gives us that $\mathcal{E}_{i,t}$ holds with probability at least $1 - 2n^{-8}$. Since $2k \le n$, a union bound implies that with probability at least $1 - 2n^{-7}$ the events $\mathcal{E}_{1,t}, \ldots, \mathcal{E}_{2k,t}$ hold simultaneously, that is, $\mathcal{E}_t = \cap_{i \in [2k]} \mathcal{E}_{i,t}$ holds. Defining $\mathcal{E} = \cap_{t \in [n]} \mathcal{E}_t$, by a union bound we have that $\mathcal{E}$ holds with probability at least $1 - 2n^{-6}$.

To prove our claim, we will use Lemma 19, a version of Azuma's inequality that takes into account a rare bad event. We will set the bad event to be $\overline{\mathcal{E}}$. We have just argued that $\mathbb{P}[\overline{\mathcal{E}}] \le 2n^{-6}$. Letting $c_t = 4\sqrt{d_t \ln n} + \frac{4}{n}$, we will show

$$|\mathbb{E}[f \mid X_1, \ldots, X_{t-1}, X_t = i, \mathcal{E}] - \mathbb{E}[f \mid X_1, \ldots, X_{t-1}, X_t = j, \mathcal{E}]| \le c_t \tag{2}$$

for any two colors $i, j \in [2k]$ chosen for the vertex t. For $t \in [n]$, let M_t be the number of edges incident to the set of uncolored vertices $\{t+1, \ldots, n\}$ that go between the colored pairs $(A_1, A_{1+k}), (A_2, A_{2+k}), \ldots, (A_k, A_{2k})$. For every $i, j \in [2k]$,

$$\begin{aligned}
&|\mathbb{E}[f \mid X_1, \ldots, X_{t-1}, X_t = i, \mathcal{E}] - \mathbb{E}[f \mid X_1, \ldots, X_{t-1}, X_t = j, \mathcal{E}]| \\
&= |N_{i,t} + \mathbb{E}[M_t \mid X_1, \ldots, X_{t-1}, X_t = i, \mathcal{E}] - N_{j,t} - \mathbb{E}[M_t \mid X_1, \ldots, X_{t-1}, X_t = j, \mathcal{E}]| \\
&\le |N_{i,t} - N_{j,t}| \\
&\qquad + |\mathbb{E}[M_t \mid X_1, \ldots, X_{t-1}, X_t = i, \mathcal{E}] - \mathbb{E}[M_t \mid X_1, \ldots, X_{t-1}, X_t = j, \mathcal{E}]|,
\end{aligned}$$

where the inequality follows from the triangle inequality. Note that for any $i \in [n]$, we have $0 \le \mathbb{E}[M_t \mid X_1, \ldots, X_{t-1}, X_t = i, \overline{\mathcal{E}}] \le m$. Also, since M_t only involves vertices $\{t+1, \ldots, n\}$, we have, for any $j \in [n]$,

$$\mathbb{E}[M_t \mid X_1, \ldots, X_{t-1}, X_t = i] = \mathbb{E}[M_t \mid X_1, \ldots, X_{t-1}, X_t = j]. \tag{3}$$

and combining (1) and (3), and conditioning on $\mathcal{E}$,

$$\begin{aligned}
\Delta_t \quad &\le \quad \max_{i,j \in [2k]} |N_{i,t} - N_{j,t}| \\
&\qquad + \frac{\mathbb{P}[\overline{\mathcal{E}}]}{\mathbb{P}[\mathcal{E}]} \left| \mathbb{E}[M_t \mid X_1, \ldots, X_t = i, \overline{\mathcal{E}}] - \mathbb{E}[M_t \mid X_1, \ldots, X_t = j, \overline{\mathcal{E}}] \right| \\
&\le \quad \max_{i,j \in [2k]} |N_{i,t} - N_{j,t}| + \frac{2m \cdot \mathbb{P}[\overline{\mathcal{E}}]}{\mathbb{P}[\mathcal{E}]} < 4\sqrt{d_t \ln n} + \frac{4}{n} = c_t.
\end{aligned}$$

To apply Lemma 19, using $\sum_{t=1}^{n} d_t = m$, we compute $\sum_t c_t^2$ as follows:

$$\begin{aligned}\sum_{t\in[n]} c_t^2 &= 16\ln n \sum_{t\in[n]} d_t + \sum_{t\in[n]} \frac{16}{n^2} + \frac{32}{n}\sum_{t\in[n]} \sqrt{d_t \ln n} \\ &= 16m\ln n + \frac{16}{n} + \frac{32}{n}\sum_{t\in[n]} \sqrt{d_t \ln n} \\ &\le 16m\ln n + \frac{16}{n} + 32\sqrt{\frac{m\ln n}{n}} \le 17m\ln n,\end{aligned}$$

where the penultimate inequality follows from the concavity of the square-root function. Setting $s = 9\sqrt{m}\log n - 1 > 9\sqrt{m}\ln n - 1$ and invoking Lemma 19 finishes the proof.

(b) The proof is analogous to the one in part (a) with $f(X_1,\dots,X_n) = \sum_{i=1}^{2k} e(A_i)$.

(c) Let $A = \{a_1,\dots,a_{|A|}\}$ and $B = \{b_1,\dots,b_{|B|}\}$. The vertex sequence is $a_1,\dots,a_{|A|}$ followed by $b_1,\dots,b_{|B|}$ where $X_i \in [k]$ is the color of the ith vertex, for $i \in [|A|+|B|]$. Then $f(X_1,\dots,X_n) = \sum_{i=1}^{k} e(A_i,B_i)$, and the proof is analogous to part (a). ◀

5 Edge Estimation using BIS Queries

First, we prove Lemma 7 about the coarse estimator. Then, we use this coarse estimator and importance sampling (Lemma 18), sparsification (Lemma 10), and exact edge counting (Lemma 5) to design our overall algorithm.

5.1 Coarse Estimator

We prove Lemma 7 by giving an efficient algorithm that coarsely estimates the number of edges $e(A,B)$ between A and B using $O(\log^3 n)$ BIS queries. To describe the algorithm, we will need some more notation. For a subset $S \subseteq [n]$, define $N(S)$ to be the union of the neighbors of all the vertices in S and for a vertex v, let $\deg_S(v)$ denote the number of neighbors of v that lie in S. For $i \in [\log n]$, define the set of vertices in A with degree between 2^i and 2^{i+1} as $A_i = \{v \mid v \in A,\ 2^i < \deg_B(v) \le 2^{i+1}\}$, and let A_0 denote the vertices in A with $\deg_B(v) \le 2$. We start with the following claim.

▶ **Claim 11.** *There exists an $i^* \in \{0,1,\dots,\log n\}$ such that*

$$e(A_{i^*},B) \ge \frac{e(A,B)}{\log n + 1} \qquad \text{and} \qquad |A_{i^*}| \ge \frac{e(A,B)}{2^{i^*}} \cdot \frac{1}{2(\log n+1)}.$$

Proof. Since $\sum_{i=0}^{\log n} e(A_i,B) = e(A,B)$, the proof of the first inequality follows from averaging. To see the second inequality, observe that for every i, we have $e(A_i,B) \le |A_i|2^{i+1}$. Hence, using the first inequality $|A_{i^*}| \ge \frac{e(A_{i^*},B)}{2^{i^*+1}} \ge \frac{e(A,B)}{2^{i^*}} \cdot \frac{1}{2(\log n+1)}$. ◀

Suppose we have an estimate $\widetilde{e}$ for $e(A,B)$. Consider CheckEstimate from Algorithm 1 for checking if $\widetilde{e}$ is correct up to logarithmic factors using logarithmically many BIS queries.

We have the following claim about the test described in Algorithm 1.

▶ **Claim 12.** *Let $n \ge 16$. If $e(A,B) > 0$, then*

(a) *if $\widetilde{e} \ge 4e(A,B)(\log n + 1)$,* CheckEstimate$((A,B),\widetilde{e})$ *accepts with probability at most $\frac{1}{4}$.*

(b) *if $\widetilde{e} \le \frac{e(A,B)}{4\log n}$,* CheckEstimate$((A,B),\widetilde{e})$ *accepts with probability at least $\frac{1}{2}$.*

Algorithm 1: CheckEstimate$((A, B), \widetilde{e})$

Input: $((A, B), \widetilde{e})$ where $A, B \subseteq [n]$ are disjoint and $\widetilde{e}$ is a guess for $e(A, B)$

```
for i = 0, 1, ..., log n do
    Sample A' ⊆ A by choosing each vertex in A with probability min{2^i/ẽ, 1}.
    Sample B' ⊆ B by choosing each vertex of B with probability 1/2^i.
    if e(A', B') ≠ 0 then
        Output accept;
    end
end
Output reject.
```

Algorithm 2: CoarseEstimator(A, B)

Input: (A, B) where $A, B \subseteq [n]$ are disjoint

Output: An estimate $\widetilde{e}$ for the number of edges $e(A, B)$

```
if e(A, B) = 0 then
    Output 0;
end
for j = 2 log n, ..., 0 do
    Run t := 128 log n independent trials of CheckEstimate((A, B), 2^j).
    if at least 3t/8 of them output accept then
        Output 2^j;
    end
end
```

Proof.

(a) For any value of the loop variable i, the probability that a fixed edge is present in the induced subgraph on A' and B' is $\min\left\{\frac{2^i}{\widetilde{e}}, 1\right\} \cdot \frac{1}{2^i} \leq \frac{1}{\widetilde{e}}$. Thus, $\mathbb{E}[e(A', B')] = \frac{e(A,B)}{\widetilde{e}} \leq \frac{1}{4(\log n+1)}$ and hence, the probability that the event $e(A', B') \neq 0$ happens for a particular value of i is $\mathbb{P}[e(A', B') \neq 0] \leq \mathbb{E}[e(A', B')] \leq \frac{1}{4(\log n+1)}$. By the union bound over the loop variable, the probability that the test accepts is at most $\frac{1}{4}$.

(b) It is enough to show that the probability is at least $\frac{1}{2}$ when the loop variable attains the value i^* where i^* is given by Claim 11. Then, we have that $|A_{i^*}| \geq \frac{e(A,B)}{2^{i^*}} \frac{1}{2(\log n+1)}$ and

$$\mathbb{P}[A' \cap A_{i^*} = \varnothing] = \left(1 - \frac{2^{i^*}}{\widetilde{e}}\right)^{|A_{i^*}|} \leq \exp\left(-\frac{m}{\widetilde{e}} \frac{1}{2(\log n + 1)}\right)$$
$$\leq \exp\left(-\frac{4 \log n}{2(\log n + 1)}\right) \leq \frac{1}{e^{1.6}},$$

where the penultimate inequality follows since $\widetilde{e} \leq \frac{e(A,B)}{4(\log n+1)}$ and the final uses $n \geq 16$. Furthermore, since $\deg_B(v) \geq 2^{i^*}$ for any $v \in A_{i^*}$, it follows that when $A' \cap A_{i^*} \neq \varnothing$, then $|N(A' \cap A_{i^*})| \geq 2^{i^*}$. So, we can bound

$$\mathbb{P}[B' \cap N(A' \cap A_{i^*}) = \varnothing \mid A' \cap A_{i^*} \neq \varnothing] \leq \left(1 - \frac{1}{2^{i^*}}\right)^{2^{i^*}} \leq \frac{1}{e}.$$

From the above, we get

$$\mathbb{P}[e(A',B') \neq 0] = \mathbb{P}[A' \cap A_{i^*} \neq \varnothing]\ \mathbb{P}[B' \cap N(A' \cap A_{i^*}) \neq \varnothing \mid A' \cap A_{i^*} \neq \varnothing]$$
$$\geq \left(1 - \frac{1}{e^{1.6}}\right)\left(1 - \frac{1}{e}\right) \geq \frac{1}{2}. \qquad \blacktriangleleft$$

Armed with the above test, we can easily estimate the number of edges up to a $O(\log n)$ factor by just doing a search, where we start with $\widetilde{e} = n^2$ and halve the number of edges each iteration. The algorithm is given in Algorithm 2 and the following claim gives an analysis.

▶ **Claim 13.** *Let $n \geq 16$.* $\mathsf{CoarseEstimator}(A,B)$ *outputs $\widetilde{e} \leq n^2$ satisfying $\frac{e(A,B)}{8\log n} \leq \widetilde{e} \leq 8e(A,B)\log n$ with probability at least $1 - \frac{4\log n}{n^4}$. The number of BIS queries made is $c_{\mathsf{ce}} \log^3 n$ for a constant c_{ce}.*

Proof. For any fixed value of j such that $2^j \geq 4(e(A,B)\log n + 1)$, the expected number of accepts is at most $\frac{t}{4}$ using Claim 12(a). The probability that we see at least $\frac{3t}{8} = \frac{t}{4} + \frac{t}{8}$ accepts can be bounded by $e^{-2t\left(\frac{1}{8}\right)^2} \leq n^{-2}$ by taking $\mu_u = \frac{t}{4}$ and $s = \frac{t}{8}$ in Lemma 17(a). By a union bound over j, the probability that none of the iterations satisfying $2^j \geq 8e(A,B)\log n \geq 4e(A,B)(\log n + 1)$ accept is at least $1 - \frac{2\log n + 1}{n^4}$ by the choice of $t = 128\log n$.

On the other hand, when $2^j \leq \frac{e(A,B)}{4\log n}$, the expected number of accepts is at least $\frac{t}{2}$ and so the probability that we see at least $\frac{3t}{8} = \frac{t}{2} - \frac{t}{8}$ accepts is at least $1 - e^{-2t\left(\frac{1}{8}\right)^2} \geq 1 - \frac{1}{n^4}$ by applying Lemma 17(a) with $\mu_l = \frac{t}{2}$ and $s = \frac{t}{8}$. Hence, conditioned on the event that the estimator has not accepted for any j satisfying $2^j > \frac{e(A,B)}{4\log n}$, the probability that we accept for the unique j that satisfies $\frac{e(A,B)}{8\log n} \leq 2^j < \frac{e(A,B)}{4\log n}$, is at least $1 - n^{-4}$.

Overall, by the union bound, the probability of outputting an estimate $\widetilde{e}$ that does not satisfy $\frac{e(A,B)}{8\log n} \leq \widetilde{e} \leq 8e(A,B)\log n$ is at most $\frac{2\log n + 2}{n^4} \leq \frac{4\log n}{n^4}$. The number of queries is at most $128\log n \cdot (2\log n + 1)(\log n + 1) = 256\log^3 n + O(\log^2 n)$ since for each value of j there are $t = 128\log n$ trials of $\mathsf{CheckEstimate}$, each of which makes $\log n + 1$ BIS queries. $\blacktriangleleft$

5.2 Overall BIS Algorithm (Proof of Theorem 2)

We now describe $\mathsf{EdgeEstimator}$ (Algorithm 3) that makes $\frac{\mathrm{polylog}(n)}{\varepsilon^4}$ BIS queries to estimate m within a factor of $(1 \pm \varepsilon)$. The subroutine $\mathsf{BipartiteEstimator}$ is given by Algorithm 4.

▶ **Theorem 14.** *Let $n \geq 16$. If $\frac{36\log n}{\sqrt{m}} \leq \varepsilon \leq \frac{1}{2}$, then with probability at least $1 - \frac{\mathrm{polylog}(n)}{n^2}$*
(a) $|m - \mathsf{EdgeEstimator}(\varepsilon)| \leq \varepsilon \cdot m$,
(b) $\mathsf{EdgeEstimator}(\varepsilon)$ *uses $O\left(\frac{\log^{16} n}{\varepsilon^4}\right)$ BIS queries.*

To prove the above, we first analyze $\mathsf{BipartiteEstimator}$. To this end, we need some more definitions. Let L be a list of (A,B,w), for $A,B \subseteq [n]$ that are disjoint and $w \geq 1$. For every $0 < \delta < \frac{1}{32\log n}$, L is *good* if $|L| \leq 2t\log(n)$ and for every $(A,B,w) \in L$, $e(A,B) \geq s$, where $t = 2^{13} \cdot \frac{\log^5 n}{\delta^2}$ and $s = 81 \cdot \frac{k\log^2 n}{\delta^2}$ as defined in Step 5 of $\mathsf{BipartiteEstimator}$. Define $e(L) = \sum_{(A,B,w)\in L} e(A,B)$.

We have the following guarantee on $\mathsf{BipartiteEstimator}$.

▶ **Lemma 15.** *Let $n \geq 16$. If L is good and $0 < \delta < \frac{1}{32\log n}$, then with probability at least $1 - \frac{16kt\log^2 n \cdot \log_{\lceil \frac{k}{2} \rceil} e(L)}{n^4}$, the following holds*
(a) $|\mathsf{wt}(L) - \mathsf{BipartiteEstimator}(L,k,\delta)| \leq 4\delta \log_{\lceil \frac{k}{2} \rceil} e(L) \cdot \mathsf{wt}(L)$.

Algorithm 3: EdgeEstimator(ε)

Input: (n, ε): n is the number of vertices and ε is an error parameter.
Output: Estimate $\widetilde{m}$ for the number of edges $m = |E|$.

Partition the vertices randomly into A and B.
Compute $\widetilde{e} = \mathsf{CoarseEstimator}(A, B)$.
if $\widetilde{e} \leq \frac{2^{20} \cdot \log^5 n}{\varepsilon^2}$ **then**
  Compute $\widetilde{m} = 2 \cdot e(A, B)$ exactly using Lemma 5.
end
else
  Compute $\widetilde{m} = 2 \cdot \mathsf{BipartiteEstimator}\left([(A, B, 1)], 4, \frac{\varepsilon}{32 \log n}\right)$ (Algorithm 4).
end
return $\widetilde{m}$.

Algorithm 4: BipartiteEstimator(L, k, δ)

Input: (L, k, δ): L is a list of triples (A, B, w) where $A, B \subseteq [n]$ are disjoint subsets and $w \geq 1$ is a positive weight for the pair (A, B), k is an integer, and δ is an error parameter.
Output: Estimate for the sum $\mathsf{wt}(L) := \sum_{(A,B,w) \in L} w \cdot e(A, B)$.

$L_{\mathsf{ref}} = \mathsf{Refine}(L, k)$.
for *each* $(A, B, w) \in L_{\mathsf{ref}}$ **do**
  $\widetilde{e}(A, B) = \mathsf{CoarseEstimator}(A, B)$.
end
Set $t := 2^{13} \cdot \frac{\log^5 n}{\delta^2}$ and $s := 81 \cdot \frac{k \log^2 n}{\delta^2}$.
Define $L_{\mathsf{light}} = \{(A, B, w) \mid (A, B, w) \in L_{\mathsf{ref}},\ \widetilde{e}(A, B) \leq 8s \log n\}$.
Compute $\mathsf{wt}(L_{\mathsf{light}}) = \sum_{(A,B,w) \in L_{\mathsf{light}}} w \cdot e(A, B)$ exactly using Lemma 5.
Remove L_{light} from L_{ref}.
For $j \in [2 \log n]$, let $S_j = \{(A, B, w) \mid (A, B, w) \in L_{\mathsf{ref}},\ w \cdot \widetilde{e}(A, B) \in (2^j, 2^{j+1}]\}$.
  Define $\widetilde{e}(S_j) = \sum_{(A,B,w) \in S_j} \widetilde{e}(A, B)$.
for $j = 1, \cdots, 2 \log n$ **do**
  if $|S_j| > t$ **then**
    Sample t elements uniformly and independently from S_j. Denote this multiset by S'_j.
    Update L_{ref} by replacing each $(A, B, w) \in S_j$ that was sampled in S'_j with $\left(A, B, \frac{pw|S_j|}{t}\right)$, where p is the number of copies of (A, B, w) in S'_j.
    Remove each $(A, B, w) \in S_j$ from L_{ref} that is not in S'_j.
  end
end
Let L_{sub} be the current list.
return $\mathsf{wt}(L_{\mathsf{light}}) + \mathsf{BipartiteEstimator}(L_{\mathsf{sub}}, k, \delta)$.

Algorithm 5: Refine(L, k)

Input: (L, k): L is a list of triples (A, B, w) where $A, B \subseteq [n]$ are disjoint subsets and $w \geq 1$ is a positive weight for the pair (A, B), k is an integer

```
for each (A, B, w) ∈ L do
    Partition A and B uniformly each into k classes A_1, ..., A_k and B_1, ..., B_k.
    Update L by replacing (A, B, w) with (A_1, B_1, wk), ··· , (A_k, B_k, wk).
end
return L.
```

(b) *the number of BIS queries is at most* $c_{\mathsf{bp}} kst \log^4 n \cdot \log_{\lceil \frac{k}{2} \rceil} e(L)$ *where* $c_{\mathsf{bp}} = 2^{10} + 2c_{\mathsf{ce}}$ *and* c_{ce} *is the constant from Lemma 7.*

We prove Theorem 14 using Lemma 15. Lemma 15 is applied with $L = [(A, B, 1)]$, $k = 4$ and $\delta = \frac{\varepsilon}{32 \log n}$. Since $e(L) \leq n^2$, the number of BIS queries is at most $\frac{c \log^{16} n}{\varepsilon^4}$ for a constant c (after plugging in the values of k, s, t,δ).

We first informally describe BipartiteEstimator, expanding on the overview presented in Section 2. The list L, which is the input to BipartiteEstimator, corresponds to the collection of bipartite subgraphs along with their weights. The quantity $e(L)$ denotes the total number of edges in this collection without the weights. The algorithm BipartiteEstimator returns an estimate of $\mathsf{wt}(L)$, which is the weighted sum of the number of edges in the subgraphs.

At every level of the recursion, we maintain two invariants about the list we recurse on. The first is that the list size is $O(t \log n)$. This comes from the fact that we only keep a small collection of bipartite subgraphs that we recurse on. The second invariant is that for every element (A, B, w) in the list, we have $e(A, B) \geq s$. This says that we only recurse on those subgraphs that are dense. These invariants are captured in the definition of L being good. Both parameters t and s will be set to polylog(n) while k will be a constant.

Let L be the input to BipartiteEstimator. The algorithm starts with sparsifying each subgraph $(A, B, w) \in L$ by further partitioning it into k parts $(A_1, B_1), \ldots, (A_k, B_k)$. Denoting the new list by L_{ref}, Lemma 10 then guarantees that $e(L_{\mathsf{ref}}) \approx e(L)/k$ (graph is sparsified) and $\mathsf{wt}(L_{\mathsf{ref}}) \approx \mathsf{wt}(L)$ (weighted sum of the number of edges in the sparsified graph is a good estimate of the original number of edges) since we partition each $(A, B, w) \in L$ individually and increase the weight of each one by a factor of k.

Next, the algorithm BipartiteEstimator computes the coarse estimates $\widetilde{e}(A, B)$ for every (A, B, w) in L. Whenever the coarse estimate is $O(s \log n) = \text{polylog}(n)$, it computes $w \cdot e(A, B)$ exactly using the algorithm from Lemma 5. For the rest of the elements in the list which correspond to the dense subgraphs, the algorithm groups them into $2 \log n$ lists $S_1, \ldots, S_{2 \log n}$ according to the weighted coarse estimates $w \cdot \widetilde{e}(A, B)$ such that $w \cdot \widetilde{e}(A, B) \approx 2^j$ for every element in the list S_j. This allows us to use importance sampling (Lemma 18) – we can subsample t elements from each list and increase the weights by a factor of $|S_j|/t$. Since the coarse estimates are an $O(\log^2 n)$ factor approximation of the true estimates, and as we set $t = \text{polylog}(n)$, we are guaranteed that for the subsampled lists S'_j, we have that $\mathsf{wt}(S'_j) \approx \mathsf{wt}(S_j)$. The algorithm BipartiteEstimator then recurses on this subsampled collection of bipartite subgraphs and the subsampling step ensures that the size of the list we recurse on is $O(t \log n)$. Also, assuming our coarse estimates were correct, each subgraph in the subsampled list is dense, so our invariant about the input list being good is maintained.

Overall, the quantity $e(L)$ goes down by a factor of k in every level of the recursion because of sparsification, so there are $O(\log_k e(L)) = O(\log n)$ levels. Each level of the

recursion incurs an additive error of $\delta/\log n$, so that the overall error is $O(\delta)$. Also, in each level of the recursion, the queries are only made by CoarseEstimator and for the exact counting of the edges. The dominating term comes from the exact counting and since we only count the edges exactly if $\widetilde{e}(A,B) \leq O(s \log n)$, the total number of queries made is $O(kst \log^4 n \cdot \log_k e(L)) = \text{polylog}(n)$.

We move on to prove Theorem 14 and defer the proof of Lemma 15 to the full version [2].

Proof of Theorem 14. Let A, B be the random partition chosen in Step 1 of the algorithm. Let $\mathcal{E}_1$ be the event that both

$$|2 \cdot e(A,B) - m| \leq 18\sqrt{m} \cdot \log n, \tag{4}$$

and

$$\frac{e(A,B)}{8 \log n} \leq \widetilde{e} \leq e(A,B) \cdot 8 \log n. \tag{5}$$

Let $\mathcal{E}_2$ be the event that when Step 7 of the algorithm is executed,

$$|2e(A,B) - \widetilde{m}| \leq \frac{\varepsilon}{2} \cdot e(A,B), \tag{6}$$

and the number of BIS queries (made in Step 7) is at most $\frac{c \cdot \log^{16} n}{\varepsilon^4}$ for a constant c to be set later. Conditioned on $\mathcal{E}_1 \cap \mathcal{E}_2$, we will bound the number of queries the algorithm makes, and we will prove that the estimate is within the desired range. Then, we will show that $\mathcal{E}_1 \cap \mathcal{E}_2$ holds with high probability.

Correctness. Observe that $\widetilde{m}$ computed in either Step 4 or Step 7 satisfies (6). Indeed, for Step 4, this follows by the setting of $\widetilde{m} = 2e(A,B)$, and for Step 7, this follows by conditioning on $\mathcal{E}_2$. Thus, the triangle inequality combined with (4) and (6) implies

$$\begin{aligned} |m - \mathsf{EdgeEstimator}(\varepsilon)| = |m - \widetilde{m}| &\leq |m - 2e(A,B)| + |2e(A,B) - \widetilde{m}| \\ &\leq 18\sqrt{m} \log n + \frac{\varepsilon}{2} \cdot e(A,B) \leq \frac{\varepsilon}{2} \cdot m + \frac{\varepsilon}{2} \cdot e(A,B) \leq \varepsilon m, \end{aligned}$$

where we used the assumption $\varepsilon \geq \frac{36 \log n}{\sqrt{m}}$ and the fact $e(A,B) \leq m$.

Number of Queries. In Step 2, CoarseEstimator makes $O(\log^3 n)$ queries, by Lemma 7. In Step 4, the algorithm from Lemma 5 makes at most $5e(A,B) \cdot \log n + 2$ queries. This is bounded by $5e(A,B) \cdot \log n + 2 \leq 40\widetilde{e} \cdot \log^2 n + 2 = O\left(\frac{\log^7 n}{\varepsilon^2}\right)$, where we used (5) to upper bound $e(A,B)$ and the assumption on $\widetilde{e}$ in Step 3 to achieve the final bound. Finally, in Step 7, conditioned on $\mathcal{E}_2$, the number of queries is $O\left(\frac{\log^{16} n}{\varepsilon^4}\right)$.

Probability. We now analyze the probability of $\mathcal{E}_1 \cap \mathcal{E}_2$. In particular, we will show $\mathbb{P}\left[\overline{\mathcal{E}_1}\right] \leq \frac{4 \log n}{n^4}$ and $\mathbb{P}\left[\overline{\mathcal{E}_2} | \mathcal{E}_1\right] \leq \frac{c' \log^8 n}{n^2}$, for a constant $c' > 0$. This suffices since then

$$\mathbb{P}\left[\overline{\mathcal{E}_1} \cup \overline{\mathcal{E}_2}\right] = \mathbb{P}\left[\overline{\mathcal{E}_1}\right] + \mathbb{P}\left[\overline{\mathcal{E}_2} \cap \mathcal{E}_1\right] \leq \mathbb{P}\left[\overline{\mathcal{E}_1}\right] + \mathbb{P}\left[\overline{\mathcal{E}_2} | \mathcal{E}_1\right] = \frac{\text{polylog}(n)}{n^2}.$$

The claim that $\mathbb{P}\left[\overline{\mathcal{E}_1}\right] \leq \frac{4 \log n}{n^4}$ follows directly from Lemma 10, Lemma 7 and a union bound.

To bound $\mathbb{P}\left[\overline{\mathcal{E}_2} | \mathcal{E}_1\right]$, we invoke Lemma 15 to show that both (6) holds and the number of BIS queries is $\frac{c \log^{16} n}{\varepsilon^4}$ with high probability where $c = 4c_{\mathsf{bp}} \cdot 2^{13} \cdot 81$ and c_{bp} is the constant

Algorithm 6: EdgeEstimatorIS(n, ε)

Input: (n, ε): n is the number of vertices, and ε is an error parameter.
Output: Estimate $\widetilde{m}$ for the number of edges $m = |E|$.

1 **In parallel for** $\widetilde{k} = 2^j$ *with* $1 \le 2^j \le \lfloor n/2 \rfloor$ **do**
2 $\quad \widetilde{m}_{\widetilde{k}} = \mathsf{ColorCountIS}(n, \widetilde{k})$
3 $\quad$ Set $k^* = \left\lfloor \frac{\varepsilon \widetilde{k}}{18 \log^4 n} \right\rfloor$
4 $\quad$ Terminate the whole parallel for loop when the earliest iteration finishes.
5 **end**
6 **return** $\widetilde{m} = \mathsf{ColorCountIS}(n, k^*)$.

Algorithm 7: ColorCountIS(n, k)

Input: (n, k): n is the number of vertices, and k is the number of colors.
Output: Estimate $\widetilde{m}$ for the number of edges $m = |E|$.

1 Partition the vertices into $2k$ color classes $A_1, \ldots, A_{2k}$ uniformly at random.
2 Exactly compute $\widetilde{m} = \sum_{i=1}^{k} e(A_i, A_{k+i})$ using the algorithm from Lemma 8.
3 **return** $2k \cdot \widetilde{m}$.

from Lemma 15. The lemma requires that L is good, which holds because $L = [(A, B, 1)]$, $k = 4$, $\delta = \frac{\varepsilon}{32 \log n}$, and using (5),

$$e(A,B) \ge \frac{\widetilde{e}}{8 \log n} > \frac{2^{10} \cdot 81 \cdot \log^4 n}{\varepsilon^2} \ge \frac{81 k \log^2 n}{\delta^2} = s,$$

where the second inequality holds since BipartiteEstimator executes in Step 7 only when $\widetilde{e} > \frac{2^{13} \cdot 81 \cdot \log^5 n}{\varepsilon^2}$. To verify (6), we see $\mathsf{wt}([(A,B,1)]) = e([(A,B,1)]) = e(A,B) \le n^2$. Lemma 15 implies that with probability at least $1 - \frac{c_p \log^{10} n}{\varepsilon^2 n^4}$ (where c_p is a constant). Thus,

$$\begin{aligned} |2e(A,B) - \widetilde{m}| &= 2 \left| \mathsf{wt}([(A,B,1)]) - \mathsf{BipartiteEstimator}\left([(A,B,1)], 4, \frac{\varepsilon}{32 \log n}\right) \right| \\ &\le 8 \cdot \frac{\varepsilon}{32 \log n} \cdot \log(n^2) e(A,B) \le \frac{\varepsilon}{2} \cdot e(A,B). \end{aligned}$$

and the number of BIS queries is $O\left(\frac{\log^{16} n}{\varepsilon^4}\right)$. Since $\varepsilon \ge \frac{36 \log n}{\sqrt{m}} \ge \frac{36 \log n}{n}$, it follows that $\mathbb{P}\left[\overline{\mathcal{E}_2} | \mathcal{E}_1\right] \le \frac{c_p \log^{10} n}{\varepsilon^2 n^4} \le \frac{c' \log^8 n}{n^2}$ for a constant $c' > 0$. ◀

6 Edge Estimation using IS Queries

We briefly describe the sparsification based algorithm that uses $\sqrt{m} \cdot \mathrm{polylog}(n)/\varepsilon$ IS queries. The algorithm is given in Algorithm 6, and its guarantees are captured in Lemma 16. We defer the proof of Lemma 16 to the full version [2].

▶ **Lemma 16.** *For any $\varepsilon, m > 0$ such that $\varepsilon \ge \frac{324 \log^4 n}{\sqrt{m}}$, Algorithm 6 outputs $\widetilde{m}$ satisfying $(1-\varepsilon)m \le \widetilde{m} \le (1+\varepsilon)m$ and uses $O\left(\frac{\sqrt{m} \cdot \log^7 n}{\varepsilon}\right)$ IS queries with probability at least $1 - \frac{1}{n^3}$.*

To get Theorem 3 from the above lemma, we combine this with the folklore algorithm that computes a $(1\pm\varepsilon)$ approximation with $n^2 \cdot \text{polylog}(n)/(\varepsilon^2 m)$ edge existence queries, *i.e.*, we run both algorithms in parallel and output the estimate of the algorithm that terminates earlier. A complete analysis of the folklore algorithm can be found in the full version [2].

7 Conclusion

We studied the task of using either BIS or IS queries to estimate the number of edges in a graph. We presented randomized algorithms giving a $(1+\varepsilon)$-approximation using $\text{polylog}(n)/\varepsilon^4$ BIS queries and $\min\left\{n^2/(\varepsilon^2 m), \sqrt{m}/\varepsilon\right\} \cdot \text{polylog}(n)$ IS queries. Our algorithms estimated the number of edges by first sparsifying the graph and then exactly counting edges spanning certain bipartite subgraphs. We now describe open questions.

7.1 Open Directions

An obvious unresolved question is whether there is an algorithm to estimate the number of edges with $o(\sqrt{m})$ IS queries when $m = o(n^{4/3})$. In this context proving a lower bound of $\Omega(\sqrt{m})$ IS queries would also be very interesting. In the full version [2] of the paper we present arguments which suggest that a non-trivial lower bound might hold for IS queries.

Other open questions include using $\text{polylog}(n)$ BIS queries to estimate the number of cliques in a graph (see [9] for an algorithm using degree, neighbor and edge existence queries) or to sample a uniformly random edge (see [11] for an algorithm using degree, neighbor and edge existence queries). In general, any graph estimation problems may benefit from BIS or IS queries, possibly in combination with standard queries (such as neighbor queries). Finally, it would be interesting to know what other oracles, besides subset queries, enable estimating the number of edges (or other graph parameters) with $\text{polylog}(n)$ queries.

Acknowledgments. We thank anonymous referees for helpful comments about improving the presentation of our paper and for pointing out relevant references.

References

1 B. Aronov and S. Har-Peled. On Approximating the Depth and Related Problems. *SIAM J. Comput.*, 38(3):899–921, 2008.

2 Paul Beame, Sariel Har-Peled, Sivaramakrishnan Natarajan Ramamoorthy, Cyrus Rashtchian, and Makrand Sinha. Edge Estimation with Independent Set Oracles. *CoRR*, abs/1711.07567, 2017. `arXiv:1711.07567`.

3 Ido Ben-Eliezer, Tali Kaufman, Michael Krivelevich, and Dana Ron. Comparing the strength of query types in property testing: The case of k-colorability. *Computational Complexity*, 22(1):89–135, 2013.

4 Sergio Cabello and Miha Jejčič. Shortest paths in intersection graphs of unit disks. *Computational Geometry*, 48(4):360–367, 2015.

5 Chao L Chen and William H Swallow. Using Group Testing to Estimate a Proportion, and to Test the Binomial Model. *Biometrics*, pages 1035–1046, 1990.

6 Holger Dell and John Lapinskas. Fine-grained reductions from approximate counting to decision. *CoRR*, abs/1707.04609, 2017.

7 Robert Dorfman. The Detection of Defective Members of Large Populations. *The Annals of Mathematical Statistics*, 14(4):436–440, 1943.

8 Devdatt P Dubhashi and Alessandro Panconesi. *Concentration of Measure for the Analysis of Randomized Algorithms*. Cambridge University Press, 2009.

9 T. Eden, D. Ron, and C. Seshadhri. On Approximating the Number of k-cliques in Sublinear Time. *ArXiv e-prints*, 2017. arXiv:1707.04858.

10 Talya Eden, Amit Levi, Dana Ron, and C. Seshadhri. Approximately counting triangles in sublinear time. In Venkatesan Guruswami, editor, *IEEE 56th Annual Symposium on Foundations of Computer Science, FOCS 2015, Berkeley, CA, USA, 17-20 October, 2015*, pages 614–633. IEEE Computer Society, 2015.

11 Talya Eden and Will Rosenbaum. On Sampling Edges Almost Uniformly. *arXiv preprint arXiv:1706.09748*, 2017.

12 Moein Falahatgar, Ashkan Jafarpour, Alon Orlitsky, Venkatadheeraj Pichapati, and Ananda Theertha Suresh. Estimating the number of defectives with group testing. In *ISIT*, pages 1376–1380. IEEE, 2016.

13 Uriel Feige. On sums of independent random variables with unbounded variance and estimating the average degree in a graph. *SIAM Journal on Computing*, 35(4):964–984, 2006.

14 Aleksei V Fishkin. Disk graphs: A short survey. In *International Workshop on Approximation and Online Algorithms*, pages 260–264. Springer, 2003.

15 Oded Goldreich and Dana Ron. Approximating average parameters of graphs. *Random Structures & Algorithms*, 32(4):473–493, 2008.

16 Mira Gonen, Dana Ron, and Yuval Shavitt. Counting stars and other small subgraphs in sublinear-time. *SIAM J. Discrete Math*, 25(3):1365–1411, 2011.

17 Robert J Klein, Caroline Zeiss, Emily Y Chew, Jen-Yue Tsai, Richard S Sackler, Chad Haynes, Alice K Henning, John Paul SanGiovanni, Shrikant M Mane, Susan T Mayne, et al. Complement factor h polymorphism in age-related macular degeneration. *Science*, 308(5720):385–389, 2005.

18 Krzysztof Onak, Dana Ron, Michal Rosen, and Ronitt Rubinfeld. A near-optimal sublinear-time algorithm for approximating the minimum vertex cover size. *CoRR*, abs/1110.1079, 2011.

19 Dana Ron and Gilad Tsur. The power of an example: Hidden set size approximation using group queries and conditional sampling. *CoRR*, abs/1404.5568, 2014.

20 C. Seshadhri. A simpler sublinear algorithm for approximating the triangle count. *ArXiv e-prints*, 2015. arXiv:1505.01927.

21 Larry Stockmeyer. The complexity of approximate counting (preliminary version). In *Proceedings of the Fifteenth Annual ACM Symposium on Theory of Computing*, pages 118–126, Boston, Massachusetts, 25–27 1983.

22 Larry Stockmeyer. On approximation algorithms for #P. *SICOMP: SIAM Journal on Computing*, 14, 1985.

23 William H Swallow. Group Testing for Estimating Infection Rates and Probabilities of Disease Transmission. *Phytopathology (USA)*, 1985.

24 Jianguo Wang, Eric Lo, and Man Lung Yiu. Identifying the most connected vertices in hidden bipartite graphs using group testing. *IEEE Transactions on Knowledge and Data Engineering*, 25(10):2245–2256, 2013.

A Concentration Bounds

For proofs of the following bounds, see the book by Dubhashi and Panconesi [8].

▶ **Lemma 17** (Chernoff Bounds). *Let $X_1, \ldots, X_r$ be r i.i.d. random variables with $0 \leq X_i \leq 1$ and define $X = \sum_{i=1}^{r} X_i$. For $\mu = \mathbb{E}[X]$, let μ_l and μ_u be real numbers such that $\mu_l \leq \mu \leq \mu_u$.*

(a) *For any $s > 0$, we have $\mathbb{P}[X \leq \mu_l - s] \leq e^{-2s^2/r}$ and $\mathbb{P}[X \geq \mu_u + s] \leq e^{-2s^2/r}$.*

(b) *For any $0 \leq \delta < 1$, we have $\mathbb{P}[X \leq (1-\delta)\mu_l] \leq e^{-\frac{\mu_l \cdot \delta^2}{2}}$ and $\mathbb{P}[X \geq (1+\delta)\mu_u] \leq e^{-\frac{\mu_u \cdot \delta^2}{3}}$.*

(c) *For any $\delta \geq 1$, we have $\mathbb{P}[X \geq (1+\delta)\mu_u] \leq e^{-\frac{\mu_u \cdot \delta}{3}}$.*

▶ **Lemma 18** (Importance Sampling). *Let $U = \{x_1, \ldots, x_r\}$ be a set of numbers, all contained in the interval $[\frac{\alpha}{b}, \alpha b]$, for $\alpha > 0$ and $b \geq 1$. Let $\gamma > 0$ be a parameter. Consider the sum $\Gamma = \sum_{i=1}^{r} x_i$. Let X_i be a random sample chosen uniformly (and independently) from the set U, for $i = 1, \ldots, t$, and consider the estimate $Y = (r/t)\sum_{i=1}^{t} X_i$ for Γ. Then, for $t \geq \frac{b^4}{2\varepsilon^2}\left(1 + \ln \frac{1}{\gamma}\right)$, we have that $\mathbb{P}[|Y - \Gamma| \geq \varepsilon\Gamma] \leq \gamma$.*

We will need the following version of Azuma's that takes into account rare bad events.

▶ **Lemma 19.** *Let f be a function of r random variables $X_1, \ldots, X_r$ with $f(X_1, \ldots, X_r) \leq b$. Let $\mathcal{B}$ be any event and let for $i \in [r]$ let c_i satisfy*

$$\left|\mathbb{E}[f \mid X_1, \ldots, X_{i-1}, X_i = a_i, \bar{\mathcal{B}}] - \mathbb{E}[f \mid X_1, \ldots, X_{i-1}, X_i = a_i', \bar{\mathcal{B}}]\right| \leq c_i.$$

Then for any $s > 0$, we have that

$$\mathbb{P}\left[|f - \mathbb{E}[f]| > s + b \cdot \mathbb{P}(\mathcal{B})\right] \leq \exp\left(-\frac{2s^2}{\sum_{i=1}^{r} c_i^2}\right) + \mathbb{P}[\mathcal{B}]$$

Computing Exact Minimum Cuts Without Knowing the Graph

Aviad Rubinstein[*1], **Tselil Schramm**[†2], **and S. Matthew Weinberg**[‡3]

1 **Department of Computer Science, Harvard University, Cambridge, MA, USA**
aviad@seas.harvard.edu
2 **Department of Computer Science, UC Berkeley, Berkeley, CA, USA**
tschramm@cs.berkeley.edu
3 **Department of Computer Science, Princeton University, Princeton, NJ, USA**
smweinberg@princeton.edu

Abstract

We give query-efficient algorithms for the global min-cut and the s-t cut problem in unweighted, undirected graphs. Our oracle model is inspired by the submodular function minimization problem: on query $S \subset V$, the oracle returns the size of the cut between S and $V \setminus S$.

We provide algorithms computing an exact minimum s-t cut in G with $\widetilde{O}(n^{5/3})$ queries, and computing an exact global minimum cut of G with only $\widetilde{O}(n)$ queries (while learning the graph requires $\widetilde{\Theta}(n^2)$ queries).

1998 ACM Subject Classification F.2.2 Nonnumerical Algorithms and Problems

Keywords and phrases Query complexity, minimum cut

Digital Object Identifier 10.4230/LIPIcs.ITCS.2018.39

1 Introduction

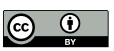
We give new algorithms for the minimum cut and s-t minimum cut problems in an unweighted, undirected graph $G = (V, E)$. Our algorithms do not assume access to the entire graph G; rather, they only interact with an oracle that, on query S, returns the value $c(S)$ of the cut between S and $V \setminus S$. Our goal is to minimize the number of queries to the oracle while computing (exact) optimum cuts.

Three easy algorithms

How many queries should we expect to be necessary? It is not hard to see that $\binom{n}{2}+n = O(n^2)$ queries suffice:[1] To find out whether there is an edge between u and v, we can query the oracle for $\{u\}$, $\{v\}$, and $\{u, v\}$. The edge is present iff $c(\{u\}) + c(\{v\}) - c(\{u, v\}) > 0$. After querying all n singletons and $\binom{n}{2}$ pairs, we have learned the entire graph. In fact, we can

* A. Rubinstein did most of the work while at UC Berkeley and a visitor at the Simons Institute for the Theory of Computing, supported by a Microsoft PhD Fellowship, as well as a Rabin Postdoctoral Fellowship, NSF grant CCF1408635, and Templeton Foundation grant 3966.
† T. Schramm is grateful for the support of an NSF Graduate Research Fellowship (1106400).
‡ S. M. Weinberg is grateful for support from NSF CCF-1717899. This work was completed in part while S. M. Weinberg was a research fellow at the Simons Institute for the Theory of Computing.

[1] We follow the standard convention and use $n = |V|$ to denote the number of vertices and $m = |E|$ to denote the number of edges. We also use $\widetilde{O}(x)$ to denote $O(x \cdot \text{polylog}(x))$, and similarly for $\tilde{\Omega}(\cdot)$ and $\tilde{\Theta}(\cdot)$.



improve slightly: the results of the cut queries are linear in $\binom{n}{2}$ unknown variables; thus $\binom{n}{2}$ linearly independent queries suffice.

For sparse graphs we can do even better: one can find a neighbor of a non-isolated vertex v in $O(\log n)$ queries using a "lion in the desert" algorithm (see Lemma 3). More generally, we can learn the entire graph using $\widetilde{O}(n+m)$ queries. But for dense graphs, $\widetilde{\Omega}(n^2)$ queries are necessary to learn the entire graph by a simple information theoretic argument (each query reveals at most $O(\log n)$ bits).

The search for a lower bound

Because $\widetilde{\Omega}(n^2)$ queries are needed to learn the graph, it is natural to conjecture that $\widetilde{\Theta}(n^2)$ is the optimal query complexity for computing the min cut. Such a lower bound would be of considerable interest: after breakthrough progress in recent years, $\widetilde{O}(n^2)$ is also the state of the art query complexity for the more general problem of submodular function minimization over subsets of n items [22].[2] Recent work of [6] indeed rules out certain kinds of algorithms with subquadratic query complexity, but determining whether submodular minimization requires $\widetilde{\Omega}(n^2)$ queries remains an exciting open problem (see Section 1.1 for more discussion), and graph cuts seemed like a promising candidate.

Our results

In defiance of our intuition, we provide algorithms for global minimum cut and minimum s-t cut that use a truly subquadratic number of queries. Our main results are:

▶ **Theorem 1** (Global Min Cut). *There exists a randomized algorithm that with high probability computes an exact global minimum cut in simple graphs using $\widetilde{O}(n)$ queries.*

▶ **Theorem 2** (Min s-t Cut). *There exists a randomized algorithm that with high probability computes an exact s-t minimum cut in simple graphs using $\widetilde{O}(n^{5/3})$ queries.*

It is worth mentioning that while our focus is query efficiency, all our algorithms run in polynomial time ($\widetilde{O}(n^2)$ or faster).

Techniques

All our algorithms are quite simple. Both results can be obtained using the following "meta-algorithm": (1) subsample a subquadratic number of edges; (2) compress the (original) graph by contracting all "safe" edges (i.e. those that do not cross the optimum cut, with high confidence, based on the subsample); and (3) learn all remaining edges.

Uniform sampling does not work for Step (1). We build on the *edge strength*-based sampling due to Benczúr and Karger [4], which in $\widetilde{O}(m)$ time yields graphs with few edges that approximate every cut well. Calculating the edge-strengths with $o(m)$ queries is non-trivial. Instead of computing the strength of every edge, we sub-sample the graph at different resolutions to classify the vertices of the graph into strongly connected components. See Section 3 for details.

For the global minimum cut problem we also provide an even simpler algorithm, which avoids edge-strength sampling: in a preprocessing step, we contract edges uniformly at

[2] Note that the more recent work of [6] requires $\tilde{O}(nM^3)$ queries, where the function is integral and M is the maximum value the function takes; for cuts in graphs, $M = n^2/4$!

random, as in Karger's algorithm [16]. After the right number of edge contractions, it suffices to sample edges uniformly at random in Step (1).

1.1 Related work

Graph cut minimization is a classical algorithms topic, with work dating back to Ford and Fulkerson [11], and too many consequent results to list. Of particular relevance to the present paper are the works of Karger and co-authors, including [16, 17, 19, 4, 18], which give randomized algorithms for computing minimum cuts and related quantities efficiently—in particular, this line of work establishes methodology for randomly compressing graphs while preserving cut information, which has been used in numerous follow-up works. Though our goal is *query* efficiency rather than runtime efficiency, we very much rely on their insights.

Another work of note is the recent result of Kawarabayashi and Thorup [20], who show that the global minimum cut can be computed deterministically in $\widetilde{O}(m)$ time. Though their setting differs from ours, our works are similar in that we too require structural theorems about the number of edges participating in minimum cuts in the graph (e.g. Lemma 8).

As mentioned above, our initial motivation for studying the min cut problem in this oracle model came from submodular function minimization (SFM). SFM was first studied by Grötschel, Lovász, and Schrijver in the 1980's [13], and has since been a popular topic of study (see e.g. [12] for a thorough treatment). For a submodular function over n items, the current best general algorithm requires $\widetilde{\Theta}(n^2)$ oracle queries [22]. Furthermore, [6] suggest that $\widetilde{\Theta}(n^2)$ is indeed the right bound: they prove an $\Omega(n^2)$ lower bound on the number of oracle calls made by a restricted class of algorithms (those that access the submodular function by naively evaluating the subgradient of its Lovász extension). For general algorithms, the current best lower bound is due to Harvey [14], who shows that $(\log_3 2 - o(1))n$ queries are needed. Graph cuts and s-t cuts are canonical examples of symmetric and asymmetric submodular functions,[3] and while it would be natural to conjecture that $\widetilde{\Omega}(n^2)$ queries are needed for graph cut problems, our work demonstrates that these problems do not provide a lower bound matching [22]'s algorithm (at least in unweighted graphs, and for randomized algorithms).

Note that there are works that bypass the $\widetilde{\Theta}(n^2)$ oracle queries barrier for special cases of SFM. For example, [6] provide an algorithm with $\widetilde{\Theta}(nM^3)$ oracle queries when the function value is integral and bounded within $[1, M]$ (for min cut M may be as large as $\Theta(n^2)$). Another special case of interest are decomposable submodular functions (e.g. [27, 25]).

We also mention a sequence of papers [8, 23, 5] which study the query efficiency of *learning* a graph under a similar query model (in which each cut query can be implemented in $O(1)$ queries). This series of papers establishes that $\Theta(m \log(n^2/m)/\log m)$ queries suffice to learn a graph on n vertices and m edges. This improves upon our naive algorithm for learning a graph (see Lemma 3) by polylogarithmic factors.

To our knowledge, no other works have previously considered the query complexity of graph cuts. However, the task of compressing graph cut information into efficient structures has been studied before from a variety of angles: sketching [2, 21], spectral sparsifiers [3], streaming spectral sparsifiers [15], skeletons [17, 4], backbones [7], and cactus representations [9, 24], to list a few. Note that an overwhelming majority of these works necessarily lose

[3] A submodular function is symmetric if $f(S) = f(\bar{S})$ for all S. $\emptyset$ and $[n]$ are always minimizers of a symmetric submodular function, so the "symmetric submodular function minimization problem" is to find a non-trivial minimizer (i.e. the minimizer $\notin \{\emptyset, [n]\}$), of which global min cut is a special case.

some (small) approximation factor through compression, and exact solutions are rare, but exist (e.g. [1]).

There is also an indirect connection between our work and lower bounds for distributed graph algorithms (e.g. [26, 10]), since our algorithms can be used to obtain upper bounds on the two-party communication complexity of min cuts in some models.[4]

1.2 Organization

In Section 2, we present our simple algorithm for global min cut, as well as important algorithmic primitives (such as subsampling edges). Then in Section 3, we introduce our query-efficient implementation of Benczúr and Karger's edge-strength based sampling, after which we demonstrate its application to global min cut in Section 4. Finally, Section 5 contains our result for min s-t cuts.

1.3 Discussion and Future Work

The main take-home message of our work is simple, randomized algorithms for *exact* global and s-t min cut with $\tilde{O}(n)$ and $\tilde{O}(n^{5/3})$ queries, respectively. In particular, our algorithm for global min cut learns (up to the polylog factors) just enough information to even specify one of the 2^n distinct cuts, and both are well below $\tilde{\Theta}(n^2)$. So in this natural oracle model, it is possible to find the exact global and s-t cut *without learning the underlying graph.*

Our work also motivates numerous directions for future work: Are weighted or directed graph cuts computable in $o(n^2)$ queries? Do deterministic min cut algorithms exist with truly subquadratic queries? Or can graph cuts still provide a $\Omega(n^2)$ submodular-function-minimization lower bound (perhaps for deterministic algorithms)? While graph cuts are indeed a very special case of submodular functions, can any of the ideas from our work be used in randomized algorithms for a broader class of submodular function minimization?

2 Global min-cut in $\widetilde{O}(n)$ queries

We begin by observing that if G has m edges, we can learn G entirely with $\widetilde{O}(m)$ queries. This is because locating a single edge takes only $O(\log n)$ time.

▶ **Lemma 3** (Learning an edge with $O(\log n)$ queries). *We can learn one neighbor of a vertex $v \in V$ in $O(\log n)$ queries.*

Proof. To find one neighbor of v, we perform the following recursive procedure: we partition $V \setminus v$ into two sets S_1 and S_2 of sizes $\lfloor \frac{n-1}{2} \rfloor, \lceil \frac{n-1}{2} \rceil$, respectively. We then query the cut values of $\{v\}$, S_i, and $S_i \cup \{v\}$, from which we can infer how many neighbors v has in S_i, for $i \in \{1, 2\}$ $\left(\left(c(\{v\}) + c(S_i) - c(S_i \cup \{v\})\right)/2\right)$. If v has no neighbors, return "no neighbors". Otherwise, if v has a neighbor in S_1, then proceed recursively in S_1; otherwise proceed recursively in S_2. ◀

The above observation suffices to learn the entire graph with $\widetilde{O}(m)$ queries, as it is easy to modify the algorithm to ignore known neighbors of v (if S_1 or S_2 contain only known neighbors of v, ignore them). If $m = \widetilde{O}(n)$, Theorem 1 follows easily.

[4] In a model where Alice and Bob can jointly compute a cut query in $O(\log n)$ communication, and have shared randomness, Theorem 1 (Theorem 2) provides a randomized protocol with $\tilde{O}(n)$ (resp. $\tilde{O}(n^{5/3})$) communication for computing the global (resp. s-t) min cut.

Otherwise, if $m \gg n$, a natural idea is to randomly subsample the edges of G until we are left with a sparse graph, and use this sparse graph to learn useful data about G. Indeed, we show that after a preprocessing step of n queries, sampling each random edge only requires $O(\log n)$ queries:

▶ **Corollary 4** (Sampling a random edge with $O(\log n)$ queries). *Given oracle access to the cut values of a graph on n vertices, after performing n initial queries we can sample a random edge in $O(\log n)$ additional queries (i.e. k uniformly random edges can be drawn in $n + O(k \log n)$ queries).*

Proof. First, as a preprocessing step, we perform n queries to determine the degree of every vertex. Now, we choose a random edge by choosing a random vertex v with probability proportional to its degree, then performing the procedure detailed in the proof of Lemma 3, but choosing to recurse on either S_1 or S_2 randomly with probability proportional to the degree of v into each set. ◀

Because we can sample random edges, we might hope to subsample G and obtain a sparse graph G' which has the same approximate cut values as G. In particular, if the minimum cut of G has value c, and we sample each edge independently with probability $\log n/c$, then the subsampled graph G' preserves all cuts with high probability within a $(\log n/c)(1 \pm \epsilon)$ factor. However, sampling with probability much smaller than $\log n/c$ will yield poor cut concentration in G'.

So if $c \approx \frac{m}{n}$, the next step in our algorithm is to do this simple uniform subsampling and work with G', which will have $\approx n \log n$ edges in expectation. But if $c \ll \frac{m}{n}$, the resulting G' will still have too many edges to learn, so we need some additional work.

Fortunately, when the minimum cut size c is small compared to the average degree, we can preprocess G to an intermediate G^* whose average degree is $\approx c$ without destroying the minimum cut via random contractions. Our preprocessing step essentially runs Karger's Algorithm [16] (reproduced here for completeness) for a well-chosen number of steps (not all the way to termination).

▶ **Algorithm 5** (Karger's Algorithm [16]).
Input: *A graph G.*
1. *For $j = 1, \ldots, n-2$:*
   (a) *Sample a random edge of G, and contract its two endpoints into a single "super-vertex".*
   (b) *Retain multi-edges, but remove self-loops.*

Output: *The cut between the two remaining super-vertices, which form a partition of G's vertices into two sets.*

In Karger's seminal paper, he proves that this algorithm finds the minimum cut in a graph with probability at least $\frac{1}{n^2}$, yielding a randomized algorithm for minimum cut.

We will not run Karger's algorithm to its completion, but rather only until there are cn total edges remaining in the graph (we can guess c within a factor of 2 at the cost of $\log n$ additional iterations). The following simple lemma shows that with constant probability, the minimum cut will survive:

▶ **Lemma 6** (Karger's Algorithm on small cuts). *Let G be a graph with minimum cut value $c > 0$. If we run Karger's algorithm on G until there are at most cn edges in the graph, then the minimum cut survives with constant probability.*

We will prove Lemma 6 in Section 2.1. Of course, we must verify that we can run T steps of Karger's algorithm with $\widetilde{O}(T)$ oracle queries:

▶ **Proposition 7.** *Given oracle access to the cut values of G, we can run T steps of Karger's algorithm using $\widetilde{O}(T)$ queries.*

Proof. The key observation is that keeping track of super-vertices requires no additional queries—it is simply a matter of treating all vertices belonging to a super-vertex as a single entity.

Each step of Karger's algorithm requires sampling a random edge, which we have already seen requires $O(\log n)$ oracle queries assuming the degree of every vertex is known. In order to keep track of the degree of super-vertices, we require only a single oracle query after every edge contraction: we ask for the cut value between the super-vertex and the remainder of the graph. ◀

At this point, our algorithm is as follows: we first run Karger's algorithm until the min cut size is comparable to the average degree, then subsample the graph to obtain a sparse graph that approximates the cuts of the original graph well. Applying concentration arguments, it's easy to see that this algorithm immediately yields an approximate min cut.

However, the following observation allows us to improve upon this, and learn the minimum cut exactly! Since the cuts in G' approximate the cuts in G well, any two nodes that are together in *every* approximate minimum cut in G' are safe to contract into a super-node (because they certainly aren't separated by the min cut). After these contractions, if there are sufficiently few edges remaining between the super-vertices, we can learn the entire remaining graph between the super-vertices and find the true minimum cut.

The following structural result shows that this is indeed the case: the total number of edges that participate in non-singleton approximately-minimum cuts is at most $O(n)$.[5]

▶ **Lemma 8** (Covering approximate min cuts with $O(n)$ edges)**.** *Let $G = (V, E)$ be an unweighted graph with minimum degree d and minimum cut value c. Let $\mathcal{C}$ be the set of all non-singleton approximate-minimum cuts in the graph, with cut value at most $c + \epsilon d$, for $\epsilon < 1$. Then $|\cup_{C\in\mathcal{C}} C|$ (the total number of edges that participate in cuts in $\mathcal{C}$) is $O(n)$.*

We remark that a similar claim is proven in [20]. While the theorem of [20] would be sufficient for our purposes,[6] our proof is extremely simple, and so we include it in Section 2.3.

This concludes our global min-cut algorithm. Below, we summarize the algorithm, and formally prove that it is correct.

▶ **Algorithm 9** (Global Min Cut with $\widetilde{O}(n)$ oracle queries)**.**
Input: *Oracle access to the cut values of an unweighted simple graph G.*
1. *Compute all of the single-vertex cuts.*
2. *For $c = 2^j$ for $j = 0, 1, \ldots, \log n$,*
   (a) *Repeat $\log n$ times:*
      (i) *Run Karger's Algorithm until there are a total of cn edges between the components in the graph, call the resulting graph G_1.*
      (ii) *Subsample each edge with probability $p = \frac{80 \ln}{\epsilon^2 c}$ to obtain a graph G_2 (any $\epsilon \in (0, 1/3)$ suffices)*

[5] A cut is non-singleton if each side has at least two nodes.
[6] Their result is stronger in the sense that they also show how to locate the cover in deterministic time $O(m)$, while our result is slightly simpler, and we only require $O(n)$ edges rather than $\widetilde{O}(n)$.

(iii) *Find all non-singleton cuts of size at most* $(1+3\epsilon)pc$ *in the graph, and contract any two nodes which are together in all such cuts, call the resulting graph* G_3

(iv) *Learn all of* G*'s edges between the super-vertices of* G_3 *to obtain* G_4 *(unless there are more than* $n \log n$ *edges, in which case abort and return to step 2a).*

(v) *Compute the minimum cut in* G_4*, and if it is the best seen so far, keep track of it.*

Output: *Return the best cut seen over the course of the algorithm.*

▶ **Theorem 10** (Mincut). *Algorithm 9 uses* $\widetilde{O}(n)$ *queries and finds the exact minimum cut in* G *with high probability.*

Proof. First, we will prove the correctness of the algorithm. Clearly, if one of the single-vertex cuts is the minimum cut, the algorithm finds this cut in step 1, so suppose that the best cut has value $\hat{c} < d_{\min}$, where $d_{\min}$ is the minimum degree in G.

In one of the iterations of step 2, c is within a factor of 2 of $\hat{c}$, and we focus on this iteration. In step 1, by Lemma 6, the minimum cut survives with at least constant probability. By the concentration arguments given in Lemma 11 and Corollary 13, in step 2 every cut in G_2 is close to the value of the cut in G_1 with high probability,[7] and so no edge in the minimum cut is contracted in step 3. Therefore, with constant probability, we find the minimum cut in step 5. Since we repeat this process $\log n$ times in step 2a, the total probability that we miss the global min cut in every iteration is polynomially small. This proves the correctness of the algorithm.

Now, we argue that at most $\widetilde{O}(n)$ queries are required. At every iteration of the inner loop, we run Karger's Algorithm for $O(n)$ steps ($\widetilde{O}(n)$ queries by Proposition 7). Then, we subsample each of cn edges each with probability $\widetilde{O}(1/c)$, or equivalently, we sample $\widetilde{O}(n)$ random edges ($\widetilde{O}(n)$ queries by Corollary 13). step 3 does not require any queries. By Lemma 8, step 4 requires learning only $O(n)$ edges ($\widetilde{O}(n)$ queries) if c is the true value of the minimum cut, and otherwise the step is aborted. Finally, step 5 requires no additional queries. Since the inner loop is repeated $\log^2 n$ times, this concludes the proof. ◀

In the following subsections, we provide proofs of key intermediate lemmas.

2.1 Compressing the graph with Karger's algorithm

▶ **Lemma 6** (restated). *[Karger's Algorithm on small cuts] Let* G *be a graph with minimum cut value* $c > 0$*. If we run Karger's algorithm on* G *until there are at most* cn *edges in the graph, then the minimum cut survives with constant probability.*

Proof. Fix a specific min cut C. We apply Karger's algorithm until the total number of edges drops to cn. At each step, the probability that we contract an edge from C is at most $1/n$, and we have at most n steps, so the probability that C survives is at least $(1-1/n)^n > 1/4$ for $n > 2$. ◀

2.2 Subsampling the graph

First, we show that if we sample with probability proportional to $\widetilde{O}(1/c)$, every cut in the subsampled graph has value close to its expectation. Because there are 2^n cuts, a simple

[7] where "close" means that the value of the cut in G_2 is within $(1 \pm \varepsilon)pk$, where k is the value of the cut in G_1.

Chernoff bound followed by a union bound is insufficient. Instead, we perform a more careful union bound by appealing to a polynomial bound on the number of approximately minimum cuts (as is standard in this setting, see e.g. [17]).

▶ **Lemma 11.** *Let $G = (V, E)$ be a multigraph with minimum cut value c, and let $G' = (V, E')$ be the result of sampling each edge of E with probability $p \geq \min\left(\frac{40 \ln n}{\epsilon^2 c}, 1\right)$. Then with high probability, every cut of value k in G has value $(1 \pm \epsilon)pk$ in G'.*

Proof. For each edge $e \in E$, consider the random binary variable $X_e \triangleq \begin{cases} 1 & e \in E' \\ 0 & \text{otherwise} \end{cases}$. Notice that $\mathbb{E}(X_e) = p$. Let C be a cut of size k. By a Chernoff bound, the probability that C has cut value deviating from its expectation by more than an ϵ-factor in G' is bounded by:

$$\mathbb{P}\left[\left|\sum_{e \in C^*} X_e - pk\right| > p\epsilon k\right] \leq 2\exp\left(-\frac{\epsilon^2 pk}{2}\right) \leq 2n^{-10k/c}, \tag{1}$$

where the last inequality follows by our choice of p.

Now, it follows from the analysis of Karger's algorithm (Lemma 12 below) that for every integer $\ell > 0$ there are at most $(2n)^{2\ell}$ cuts of value at most ℓc. Consider a cut C with value in $[\ell c, (\ell + 1)c]$ in G. Using (1), we have that the probability that its value in G' deviates from expectation by more than $\pm p(\ell + 1)\epsilon c$ is at most $n^{-10\ell}$. Taking a union bound over all such C and all values of ℓ the soundness holds with probability at least $1 - n^{-6}$. ◀

The following lemma, which we employed in order to bound the number of cuts of each size, is an oft-used consequence of Karger's algorithm (see e.g. [19]).

▶ **Lemma 12** (Bound on the number of small cuts)**.** *If a graph on n vertices has a minimum cut of size c, then there are at most $(2n)^{2\ell}$ cuts of size ℓc.*

Proof. Fix a specific cut C, such that $|C| = \ell c$. Consider Karger's algorithm, in which we contract a uniformly random edge in each step. After t steps, there are at least $(n - t)c/2$ edges in the graph (since no vertex can ever have degree less than c in Karger's algorithm). Then in the t-th step of Karger's algorithm there is probability at most $\frac{2\ell c}{(n-t)c}$ that an edge from C is contracted. Using a telescoping product argument, the probability that C survives for $n - 2\ell$ steps of the algorithm is at least $\prod_{t=0}^{n-2\ell}\left(1 - \frac{2\ell}{n-t}\right) = \frac{(2\ell)!}{n(n-1)\cdots(n-2\ell+1)} \geq n^{-2\ell}$. After $n - 2\ell$ steps, there are 2ℓ vertices remaining, so less than $2^{2\ell}$ cuts survived. Therefore in total there can only be $(2n)^{2\ell}$ such cuts in the original graph. ◀

As a corollary of Lemma 11, we have that the approximately minimum cuts of the subsampled graph correspond to approximately minimum cuts in the original graph:

▶ **Corollary 13.** *Let $G = (V, E)$ be a graph with minimum minimum cut value c. Let $G' = (V, E')$ be the result of sampling each edge in E with probability $p = \min\left(\frac{40 \ln n}{\epsilon^2 c}, 1\right)$, with $\epsilon \leq 1/3$. Then the following events occur with high probability:*

Completeness *the minimum cut of G has value at most $p(1 + \epsilon)c$ in G'.*

Soundness *every cut of value at most $p(1 + \epsilon)c$ in G' has value at most $(1 + 3\epsilon)c$ in G. Furthermore, no cut has value less than $p(1 - \epsilon)c$ in G'.*

Proof. This follows immediately from Lemma 11, because with high probability, every cut concentrates to within a $(1 \pm \epsilon)$ factor of its expectation. ◀

2.3 Covering approximate min cuts with $O(n)$ edges

▶ **Lemma 8** (restated). *[Covering approximate min cuts with $O(n)$ edges] Let $G = (V, E)$ be an unweighted graph with minimum degree d and minimum cut value c. Let $\mathcal{C}$ be the set of all non-singleton approximate-minimum cuts in the graph, with cut value at most $c + \epsilon d$, for $\epsilon < 1$. Then $|\cup_{C\in\mathcal{C}} C|$ (the total number of edges that participate in cuts in $\mathcal{C}$) is $O(n)$.*

Proof. Notice that any subset of $\mathcal{C}$ induces a partition over V, where two vertices are in the same component if they are on the same side of every cut. Let $K = C_1, \ldots, C_k$ be a minimal subset of $\mathcal{C}$, such that no additional cut $C \in \mathcal{C}$ splits any existing components into two components both of size $\geq \beta d$ when added to K. By definition, $k \leq n/\beta d$ and $|\cup_{i=1}^{k} C_i| \leq (c + \epsilon d) \cdot k$. The number of vertices with at least αd incident edges in K is therefore at most $2ck/\alpha d$. Call this set of vertices S.

Now, by definition of K, adding any other cut $C \in \mathcal{C}$ (not already covered by the edges in K) can only split an existing component if the size of at least one of them, B is small, $|B| < \beta d$. We argue that $B \subseteq S$. Suppose by contradiction that there exists a vertex $v \in B \setminus S$. Then v cannot have more than βd edges to other nodes inside B, because B is small. Since $v \notin S$, v can have at most αd edges that are already in the cuts $C_1, \ldots, C_k$. Thus, v has at least $\deg(v) - d(\alpha + \beta)$ edges crossing the new cut.

Since the new cut is not a singleton cut, we can move v to the other side of the cut, decreasing the size of the cut by $\deg(v) - 2d(\alpha + \beta))$.

We have that if $2(\alpha + \beta) < 1 - \epsilon$,

$$\deg(v) - 2d(\alpha + \beta) \geq d(1 - 2(\alpha + \beta)) > \epsilon d,$$

which is a contradiction, since this would give a cut of value less than c. So choosing $\alpha = \beta < (1 - \epsilon)/4$, we can conclude that S is the only set of vertices which will be separated into additional components if we refine our partition by adding minimum cuts from $\mathcal{C}$.

Therefore, the set $C_1, \ldots, C_k$ and all edges incident on S cover all edges participating in any non-singleton minimum cut. The cover consists of at most

$$(c + \epsilon d)k + |S| \cdot d \leq \frac{n(c + \epsilon d)}{\beta d} + \frac{2cn}{\alpha\beta d}$$

edges, so the conclusion follows. ◀

3 Connectivity-preserving sampling in the oracle model

Now we show how to subsample a graph with arbitrary connectivity to obtain a sparse graph in which all cut values are well-approximated (also known as a *sparsifier*). The algorithm and analysis are inspired by [4], but we must make modifications to both in order to optimize query efficiency. We begin with some definitions.

▶ **Definition 14.** A graph G is *k-strongly-connected* if there is no cut of size less than k in G. The *strong connectivity* of G, denoted $K(G)$, is the size of G's minimum cut.

▶ **Definition 15.** Given a graph $G = (V, E)$ and an edge $e = (u, v) \in E$, define e's *strength* k_e to be the maximum of the strong connectivities over all vertex-induced subgraphs of G containing e:

$$k_e = \max_{S\subseteq V \,:\, u,v\in S} K(G[S]),$$

where $G[S]$ denotes the vertex-induced subgraph of G on S.

The following theorem, due to Benczúr and Karger, shows that if we sample each edge with probability inversely proportional to its strength, every cut will be well-preserved.

▶ **Theorem 16** (Benczúr and Karger [4]). *Let $G = (V, E)$ be an unweighted graph. For each edge $e \in E$, let k_e denote the edge strength of e. Suppose we are given $\{k'_e\}_{e \in E}$ such that $\frac{1}{4}k_e \leq k'_e \leq k_e$. Let H be the graph formed by sampling each edge e with probability*

$$p_e = \min\left(\frac{100 \ln n}{k'_e \epsilon^2}, 1\right),$$

and then including it with weight $1/p_e$. Then with high probability, H has $O(n \ln n/\epsilon^2)$ edges, and every cut in H has value $(1 \pm \epsilon)$ of the original value in G.

While Benczúr and Karger give efficient algorithms for computing approximate edge strengths when the graph is known, in our setting we cannot afford to look at every edge. The following algorithm shows how to compute approximate edge strengths, and how to compute the sparsifier H, with $\widetilde{O}(n/\epsilon^2)$ oracle queries.

▶ **Algorithm 17** (Approximating Edge Strengths (and sampling a sparsifier H)).
Input: *An accuracy parameter ϵ, and a cut-query oracle for graph G.*
1. *(Initialize an empty graph H on n vertices).*
2. *For $j = 0, \ldots, \log n$, set $\kappa_j = n2^{-j}$ and:*
   - (a) *Subsample G' from G by taking each edge of G with probability $q_j = \min(100 \cdot 40 \cdot \frac{\ln n}{\kappa_j}, 1)$*
   - (b) *In each connected component of G':*
     - (i) *While there exists a cut of size $\leq q_j \cdot \frac{4}{5}\kappa_j$, remove the edges from that cut, and then recurse on the two sides. Let the connected components induced by removing the cut edges be $C_1, \ldots, C_r$.*
     - (ii) *For every $i \in [r]$ and every edge (known or unknown) with both endpoints in C_i, set the approximate edge strength $k'_e := \frac{1}{2}\kappa_j$ (alternatively, subsample every edge in $C_i \times C_i$ with probability $2q_j/\epsilon^2$ and add it to H with weight $\epsilon^2/2q_j$).*
     - (iii) *Update G by contracting C_i for each $i \in [r]$.*

Output: *The edge strength approximators $\{k'_e\}_{e \in E}$ (or the sparsifier H).*

▶ **Theorem 18.** *For each edge $e \in G$, the approximate edge strength given in Algorithm 17 is close to the true edge strength, $\frac{1}{4}k_e \leq k'_e \leq k_e$. Furthermore, the algorithm requires $\widetilde{O}(n/\epsilon^2)$ oracle queries to produce the sparsifier H, which satisfies:*
- *H has $O(n \ln n/\epsilon^2)$ edges*
- *The maximum weight of any edge e in H will be $O(\epsilon^2 k_e / \ln n)$*
- *Every cut in H is within a $(1 \pm \epsilon)$-factor of its value in G.*

Proof. The proof follows from two claims, which we state here and prove later:

▶ **Claim 19.** *At iteration $j = \lceil \log(n/k_e) \rceil$, the edge e is either assigned $k'_e = \frac{1}{2}\kappa_j = n/2^{j+1} \geq k_e/4$ or has already been assigned a larger value of k'_e.*

▶ **Claim 20.** *At iteration j, no edges e with $k_e < \frac{1}{2}\kappa_j$ are assigned a strength approximation.*

Given these two claims, we have that the approximate edge strength of every edge is within a factor of two of the true strength. Furthermore, to construct H, each iteration only requires $\widetilde{O}(n/\epsilon^2)$ cut queries. In step 2a, all components with strong connectivity larger than the current connectivity (κ_j) have been contracted, so there are no $2\kappa_j$-connected components. By Corollary 22 (stated shortly), the current G therefore has at most $O(n\kappa_j)$ edges. Therefore, in step 2a we have $q_j = \widetilde{O}(n/|E|)$, and the expected number of sampled

edges is therefore just $\widetilde{O}(n)$, and this step requires only $\widetilde{O}(n)$ cut queries. The operations in step 1 require no additional queries. Finally, again by Corollary 22, step 2 requires at most $\widetilde{O}(n/\epsilon^2)$ queries, and the consequent step requires no samples. The whole process is iterated $O(\log n)$ times, for a total of $\widetilde{O}(n/\epsilon^2)$ queries. The listed properties of H follow from Theorem 16.

Now, we prove our initial claims.

To prove Claim 19, consider the strongly connected component of strength k_e that e belongs to, C_e. Since we subsample edges with probability $q_j = 100 \cdot 40 \cdot \ln n/\kappa_j \geq 100 \cdot 40 \ln n/k_e$, with high probability every cut of C_e has size at least $\frac{9}{10} q_j k_e \geq \frac{9}{10} q_j \kappa_j$in G' (by concentration bounds identical to those in Lemma 11). Therefore, no minimum cut removed in step 1 will disconnect C_e. The claim follows.

To prove Claim 20, we note that by definition if $k_e < n/2^{j+1}$, then e cannot participate in any vertex-induced component with strong connectivity $\kappa_j/2$. We will prove that every component $C_1, \ldots, C_r$ created in step 1 is at least $(\kappa_j/2)$-connected. For this, it is necessary to prove that any cut of size less than $\kappa_j/2$ is removed. Let $C = \cup C_i$ be the components of G' after step 2a. First, we notice that at most n cuts in C are necessary to remove all non-strongly-connected edges. Let $S_1, \ldots, S_\ell$ be a sequence of at most $\ell \leq n$ cuts with sizes $a_1, \ldots, a_\ell$ respectively, so that $a_i \leq \kappa_j/2$ in G when restricted to the vertex-induced subgraph given by the vertices of C. Let $a'_1, \ldots, a'_\ell$ be the sizes of the cuts $S_1, \ldots, S_\ell$ in C (in the subsampled graph G').

By a Chernoff bound,

$$\mathbb{P}[a'_i - q_j a_i \geq s \cdot q_j a_i] \leq \begin{cases} \exp\left(-s q_j a_i/3\right) & s \geq 1 \\ \exp\left(-s^2 q_j a_i/3\right) & s \leq 1 \end{cases}$$

We choose $s = \frac{4}{5}\frac{\kappa_j}{a_i} - 1$ so that $(1+s) q_j a_i = q_j \cdot \frac{4}{5}\kappa_j$. Then because $a_i \leq \kappa_j/2$,

$$s q_j a_i = q_j \cdot \frac{4}{5} \cdot \kappa_j - q_j a_i \geq q_j \cdot \frac{3}{10} \cdot \kappa_j \geq 30 \ln n,$$

and and because $a_i \leq \kappa_j/2$, $s \geq \frac{3}{5}$, so

$$s^2 q_j a_i \geq 18 \ln n.$$

Thus, the probability that any of the cuts S_i has size $a'_i \geq \frac{4}{5} q_j \kappa_j$ in the subsampled graph G' is at most n^{-6}. Taking a union bound over all of the S_i, we have that with high probability, all of the S_i will be small enough in the subsampled graph to be removed. ◀

To argue that we did not sample too many edges (or require too many oracle queries) in step 2a, we must bound the number of edges with strength at least k and at most $2k$. The following lemma is the crux of the argument (this lemma is not novel and has appeared elsewhere, e.g. [4]).

▶ **Lemma 21.** *Let $G = (V, E)$ be a weighted graph without self-loops, and let $|V| = n$. Denote by $w(E)$ the total weight of the edges in E. If $w(E) \geq d(n-1)$, then G contains a strongly d-connected component.*

Proof. The proof is by induction—if $n = 2$, the conclusion is obvious. Now, by contradiction, let n be the smallest integer for which this is not the case. Since G is not d-connected, by removing a set of edges of total weight $< d$, we can split G into two components C_1, C_2 of

size n_1 and n_2 with edge sets E_1 and E_2, so that the total weight of edges among the two parts is at least $w(E_1) + w(E_2) \geq d(n-2)+1$. Since G and all of its induced subgraphs have no d-strongly-connected subgraphs, by the induction hypothesis both C_1 and C_2 must have $w(E_1) \leq d(n_1-1)$ and $w(E_2) \leq d(n_2-1)$. But then $w(E_1)+w(E_2) \leq d(n_1+n_1-2) = d(n-2)$, which is a contradiction. This completes the proof. ◀

▶ **Corollary 22.** *In an graph on n vertices which has strong connectivity k and no components with strong connectivity $\geq 2k$, there are $\Theta(nk)$ edges.*

Proof. In a strongly k-connected component, every vertex must have degree at least k, which gives the lower bound. To see the upper bound, we invoke Lemma 21 (which gives the desired conclusion by taking $d = 2k$). ◀

4 Global min-cut revisited

Now that we are in possession of a more sensitive sampling algorithm, we give a simplified global min cut algorithm ("simplified" by pushing all the complexity to the sampling procedure).

▶ **Algorithm 23** (Simpler global Min Cut with $\widetilde{O}(n)$ oracle queries)**.**
Input: *Oracle access to the cut values of an unweighted simple graph G.*
1. *Compute all of the single-vertex cuts.*
2. *Compute a sparsifier H of G using Algorithm 17 with G and with small constant ϵ.*
3. *Find all non-singleton cuts of size at most $(1+3\epsilon)$ times the size of the minimum cut in H, and contract any edge which is not in such a cut, call the resulting graph G'.*
4. *If the number of edges between the super-vertices of G' is $O(n)$, learn all of the edges between the super-vertices of G', and compute the minimum cut.*

Output: *Return the best cut seen over the course of the algorithm.*

▶ **Theorem 24.** *Algorithm 23 uses $\widetilde{O}(n)$ queries and finds the exact minimum cut in G with high probability.*

Proof. Let C^* be a minimum cut in G, and suppose the size of C^* is c. By Theorem 18, the sampling performed in step 2 will ensure that with high probability the minimum cut of H has value at least $(1-\epsilon)c$, and that the size of C^* in H is at most $(1+\epsilon)c$. For $\epsilon < 1/3$,

$$\frac{(1+\epsilon)c}{(1-\epsilon)c} = 1 + \frac{2\epsilon}{1-\epsilon} < 1 + 3\epsilon\,.$$

Therefore, in step 3 no edge in C^* will be contracted. Finally, by Lemma 8 at most $O(n)$ edges are left between the super-vertices of G' in step 4 (whp, assuming that all cuts are indeed preserved within $(1 \pm \epsilon)$). Therefore, if C^* is a non-singleton cut, it (or a cut of the same size) will be found. No step requires more than $\widetilde{O}(n)$ queries. ◀

5 s-t min-cut in $\widetilde{O}(n^{5/3})$ queries

Now, we use the low-query sampling algorithm developed in Section 3 to obtain sub-quadratic query complexity for computing min s-t cuts in undirected and unweighted graphs. Our algorithm follows the same general strategy as the minimum cut algorithm from the previous section: sample a connectivity-preserving weighted graph from G, then compress the graph by contracting edges that do not participate in the minimum cut.

▶ **Algorithm 25** (s-t min cut with $\widetilde{O}(n^{5/3})$ queries).
Input: *Oracle access to the cut values of an unweighted simple graph G.*
1. *Compute a sparsifier H of G using Algorithm 17 with G and with $\epsilon = n^{-1/3}$.*
2. *Compute a maximum s-t flow in H, and remove the participating edges from H; denote the result H'*
3. *Obtain G' from G by contracting all components that are $3\epsilon \cdot c$-connected in H'.*
4. *Learn all edges of G' and compute the minimum s-t cut in the resulting graph.*

Output: *The minimum s-t cut computed in step 4.*

▶ **Theorem 26.** *Algorithm 25 finds an exact s-t minimum cut in $\tilde{O}(n^{5/3})$ oracle calls.*

Proof. Our proof is based on the following claim:

▶ **Claim 27.** *The number of edges between the super-vertices in G' is at most $O(n^{5/3})$.*

The sparsifier H output in step 1 by Algorithm 17 has $O(n \ln n/\epsilon^2)$ edges and preserves all cuts to within a multiplicative $(1 \pm \epsilon)$. In particular the value of the s-t maximum flow is at most n, and so it is preserved to within an additive $\pm\epsilon n$.

Then, in step 2 we compute an (exact) s-t maximum flow F in H, and subtract F from H to obtain the graph H'. Note that without loss of generality, F is integral and non-circular. Let $f_H, f_G \leq n$ denote the size of the minimum s-t-cut in G, H (respectively). Since each edge has strength at most n in G, each edge has weight at most $\epsilon^2 n$ in H Therefore, by Lemma 28 (stated shortly), the total weight in flow F is at most $O(n\sqrt{f_H \cdot n\epsilon^2})$; since $f_H \approx f_G \leq n$ (up to a $(1 \pm \epsilon)$ factor), this simplifies to $O(\epsilon n^2)$ total weight.

If we could subtract exactly the maximum flow in G, we could safely contract all remaining connected components (since the max flow certainly saturates a min s-t cut). Since the (exact) s-t maximum flow in H approximates the flow in G to within an additive $\pm\epsilon n$ error, we claim that we can safely contract any $3\epsilon n$-connected component in H':

Let C be a $3\epsilon n$-connected component in H'. Assume by contradiction that there is a minimum s-t cut that separates C. Because we preserved all cuts to within a multiplicative $(1 \pm \epsilon)$, the same cut has value at most $(1+\epsilon)f_G \leq f_H + 2\epsilon n$ in H. But because there is an s-t flow of value f_H in $H \setminus H'$, all cuts have value at least f_H in $H \setminus H'$. Therefore, this approximate min s-t cut *must* cut at most $2\epsilon n$ edges in H'. So immediately by definition of k-connectivity, we obtain a contradiction to this cut possibly separating a $3\epsilon n$-connected component in H'. This establishes the correctness of the algorithm, since the exact min s-t cut is not altered in step 2.

Once we contract the $3\epsilon n$-connected components in H', we are left (by Lemma 21) with a total weight of at most $3\epsilon n^2$ in H'.

After applying the same contractions to H, we have that the total remaining weight is at most $3\epsilon n^2 + O(\epsilon n^2) = O(\epsilon n^2)$ (the sum of the flow and H'); and therefore, since the cut around each of the contracted vertices is the same in G and H up to a factor of $(1 \pm \epsilon)$, we have that the number of edges remaining in the contracted graph G' is also $|E'| = O(\epsilon n^2)$.

The total number of queries necessary is $\widetilde{O}(n/\epsilon^2)$ in step 1, and then another $|E'|$ in step 4. Choosing $\epsilon = n^{-1/3}$ balances the terms, so that we have $|E'|, n/\epsilon^2 \leq n^{5/3}$. This concludes the proof. ◀

5.1 Covering s-t min cuts with $O(n^{3/2})$ edges

▶ **Lemma 28** (Flow cover). *In an undirected graph $G = (V, E)$ with integral weights from $[0, W]$, every non-circular s-t flow (for any $s, t \in V$) of value f uses edges of at most $O(n\sqrt{fW})$ total weight.*

Proof. Consider the induced *flow graph*, i.e. the DAG that has an edge from u to v with weight equal to the flow from u to v. Fix a topological sorting of the flow graph. We define the *length* of an edge to be the difference between its endpoints in the sorting.

Bucket all the edges into $O(\log W)$ buckets according to their weights, with bucket B_w containing all the edges of weight in $[w, 2w-1]$. Let d_w denote the (unweighted) average incoming degree when only considering edges from B_w, and let ℓ_w denote the (unweighted) average length of edges in B_w.

For each $i \in [n-1]$, at most f/w edges from B_w cross the cut C_i between the first i vertices and the last $n-i$ vertices in the topological ordering (because each such edge has weight $\geq w$ and the total flow crossing any of these cuts is exactly f). Similarly, the number of cuts each that edge in B_w crosses is exactly equal to its length. We can count the total number of pairs (e, C_i) such that edge $e \in B_w$ crosses cut C_i in two different ways: summing across $(n-1)$ cuts, or summing across $|B_w|$ edges. We therefore have that

$$(n-1) \cdot f/w \geq |B_w| \cdot \ell_w. \tag{2}$$

For each vertex v, each incoming edge has a different length; therefore, the average length among its incoming edges is at least half of its degree. By the Cauchy-Schwartz inequality it follows that this is also true on average across all edges and vertices:[8]

$$\ell_w \geq d_w/2. \tag{3}$$

Observe also that $|B_w| = n \cdot d_w$. Combining this observation with Inequalities (2) and (3), we have that

$$f/w \geq \ell_w \cdot \left(\frac{B_w}{n}\right) \geq d_w^2/2.$$

In particular, for each bucket, the number of edges is bounded by $|B_w| = O(n\sqrt{f/w})$

Therefore, the total weight of all edges is bounded by

$$\sum_w |B_w| \cdot 2w = \sum_w O(n\sqrt{fw}) = O(n\sqrt{fW}).$$

◀

Acknowledgements. The authors thank Robert Krauthgamer, Satish Rao, Aaron Schild, and anonymous reviewers for helpful conversations and suggestions.

References

1 Kook Jin Ahn, Sudipto Guha, and Andrew McGregor. Analyzing graph structure via linear measurements. In Yuval Rabani, editor, *Proceedings of the Twenty-Third Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2012, Kyoto, Japan, January 17-19, 2012*, pages 459–467. SIAM, 2012. URL: `http://portal.acm.org/citation.cfm?id=2095156&CFID=63838676&CFTOKEN=79617016`.

[8] To see this, we can compute the average length ($\ell(e)$ denotes the length of edge e) as $(\sum_v \sum_{e \text{ incoming to } v} \ell(e))/\sum_v d_v \geq (\sum_v d_v^2/2)/\sum_v d_v \geq ((\sum_v d_v)^2/2n)/\sum_v d_v \geq \sum_v d_v/2n = d_w/2$.

2 Alexandr Andoni, Jiecao Chen, Robert Krauthgamer, Bo Qin, David P. Woodruff, and Qin Zhang. On sketching quadratic forms. In Madhu Sudan, editor, *Proceedings of the 2016 ACM Conference on Innovations in Theoretical Computer Science, Cambridge, MA, USA, January 14-16, 2016*, pages 311–319. ACM, 2016. doi:10.1145/2840728.2840753.

3 Joshua D. Batson, Daniel A. Spielman, and Nikhil Srivastava. Twice-ramanujan sparsifiers. *SIAM Review*, 56(2):315–334, 2014. doi:10.1137/130949117.

4 András A. Benczúr and David R. Karger. Randomized approximation schemes for cuts and flows in capacitated graphs. *SIAM J. Comput.*, 44(2):290–319, 2015. doi:10.1137/070705970.

5 Nader H. Bshouty and Hanna Mazzawi. Reconstructing weighted graphs with minimal query complexity. *Theor. Comput. Sci.*, 412(19):1782–1790, 2011. doi:10.1016/j.tcs.2010.12.055.

6 Deeparnab Chakrabarty, Yin Tat Lee, Aaron Sidford, and Sam Chiu-wai Wong. Subquadratic submodular function minimization. *CoRR*, abs/1610.09800, 2016. arXiv:1610.09800.

7 Shiri Chechik, Yuval Emek, Boaz Patt-Shamir, and David Peleg. Sparse reliable graph backbones. *Inf. Comput.*, 210:31–39, 2012. doi:10.1016/j.ic.2011.10.007.

8 Sung-Soon Choi and Jeong Han Kim. Optimal query complexity bounds for finding graphs. In Cynthia Dwork, editor, *Proceedings of the 40th Annual ACM Symposium on Theory of Computing, Victoria, British Columbia, Canada, May 17-20, 2008*, pages 749–758. ACM, 2008. doi:10.1145/1374376.1374484.

9 Efim A. Dinitz, Alexander V. Karzanov, and Micael V. Lomonosov. On the structure of a family of minimum weighted cuts in a graph. *Studies in Discrete Optimization*, pages 290–306, 1976.

10 Andrew Drucker, Fabian Kuhn, and Rotem Oshman. On the power of the congested clique model. In Magnús M. Halldórsson and Shlomi Dolev, editors, *ACM Symposium on Principles of Distributed Computing, PODC '14, Paris, France, July 15-18, 2014*, pages 367–376. ACM, 2014. doi:10.1145/2611462.2611493.

11 Lester Randolph Ford Jr. and Delbert Ray Fulkerson. *Flows in Networks*. Princeton University Press, 1962.

12 Satoru Fujishige. *Submodular Functions and Optimization*. Elsevier Science, 2005.

13 Martin Grötschel, László Lovász, and Alexander Schrijver. The ellipsoid method and its consequences in combinatorial optimization. *Combinatorica*, 1(2):169–197, 1981. doi:10.1007/BF02579273.

14 Nicholas J. A. Harvey. Matroid intersection, pointer chasing, and young's seminormal representation of S_n. In Shang-Hua Teng, editor, *Proceedings of the Nineteenth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2008, San Francisco, California, USA, January 20-22, 2008*, pages 542–549. SIAM, 2008. URL: http://dl.acm.org/citation.cfm?id=1347082.1347142.

15 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. IEEE Computer Society, 2014. doi:10.1109/FOCS.2014.66.

16 David R. Karger. Global min-cuts in rnc, and other ramifications of a simple min-cut algorithm. In Vijaya Ramachandran, editor, *Proceedings of the Fourth Annual ACM/SIGACT-SIAM Symposium on Discrete Algorithms, 25-27 January 1993, Austin, Texas.*, pages 21–30. ACM/SIAM, 1993. URL: http://dl.acm.org/citation.cfm?id=313559.313605.

17 David R. Karger. Random sampling in cut, flow, and network design problems. *Math. Oper. Res.*, 24(2):383–413, 1999. doi:10.1287/moor.24.2.383.

18 David R. Karger and Matthew S. Levine. Fast augmenting paths by random sampling from residual graphs. *SIAM J. Comput.*, 44(2):320–339, 2015. doi:10.1137/070705994.

19 David R. Karger and Clifford Stein. A new approach to the minimum cut problem. *J. ACM*, 43(4):601–640, 1996. doi:10.1145/234533.234534.

20 Ken-ichi Kawarabayashi and Mikkel Thorup. Deterministic global minimum cut of a simple graph in near-linear time. 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 665–674. ACM, 2015. doi:10.1145/2746539.2746588.

21 Dmitry Kogan and Robert Krauthgamer. Sketching cuts in graphs and hypergraphs. In Tim Roughgarden, editor, *Proceedings of the 2015 Conference on Innovations in Theoretical Computer Science, ITCS 2015, Rehovot, Israel, January 11-13, 2015*, pages 367–376. ACM, 2015. doi:10.1145/2688073.2688093.

22 Yin Tat Lee, Aaron Sidford, and Sam Chiu-wai Wong. A faster cutting plane method and its implications for combinatorial and convex optimization. In Venkatesan Guruswami, editor, *IEEE 56th Annual Symposium on Foundations of Computer Science, FOCS 2015, Berkeley, CA, USA, 17-20 October, 2015*, pages 1049–1065. IEEE Computer Society, 2015. doi:10.1109/FOCS.2015.68.

23 Hanna Mazzawi. Optimally reconstructing weighted graphs using queries. In Moses Charikar, editor, *Proceedings of the Twenty-First Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2010, Austin, Texas, USA, January 17-19, 2010*, pages 608–615. SIAM, 2010. doi:10.1137/1.9781611973075.51.

24 Dalit Naor and Vijay V. Vazirani. Representing and enumerating edge connectivity cuts in RNC. In Frank K. H. A. Dehne, Jörg-Rüdiger Sack, and Nicola Santoro, editors, *Algorithms and Data Structures, 2nd Workshop WADS '91, Ottawa, Canada, August 14-16, 1991, Proceedings*, volume 519 of *Lecture Notes in Computer Science*, pages 273–285. Springer, 1991. doi:10.1007/BFb0028269.

25 Robert Nishihara, Stefanie Jegelka, and Michael I. Jordan. On the convergence rate of decomposable submodular function minimization. In Zoubin Ghahramani, Max Welling, Corinna Cortes, Neil D. Lawrence, and Kilian Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pages 640–648, 2014. URL: http://papers.nips.cc/paper/5255-on-the-convergence-rate-of-decomposable-submodular-function-minimization.

26 Atish Das Sarma, Stephan Holzer, Liah Kor, Amos Korman, Danupon Nanongkai, Gopal Pandurangan, David Peleg, and Roger Wattenhofer. Distributed verification and hardness of distributed approximation. *SIAM J. Comput.*, 41(5):1235–1265, 2012. doi:10.1137/11085178X.

27 Peter Stobbe and Andreas Krause. Efficient minimization of decomposable submodular functions. In John D. Lafferty, Christopher K. I. Williams, John Shawe-Taylor, Richard S. Zemel, and Aron Culotta, editors, *Advances in Neural Information Processing Systems 23: 24th Annual Conference on Neural Information Processing Systems 2010. Proceedings of a meeting held 6-9 December 2010, Vancouver, British Columbia, Canada.*, pages 2208–2216. Curran Associates, Inc., 2010. URL: http://papers.nips.cc/paper/4028-efficient-minimization-of-decomposable-submodular-functions.

Approximate Clustering with Same-Cluster Queries*

Nir Ailon[†1], Anup Bhattacharya[‡2], Ragesh Jaiswal[3], and Amit Kumar[4]

1 **Technion, Haifa, Israel**
nailon@cs.technion.ac.il
2 **Department of Computer Science and Engineering, Indian Institute of Technology Delhi India**
anupb@cse.iitd.ac.in
3 **Department of Computer Science and Engineering, Indian Institute of Technology Delhi, India**
rjaiswal@cse.iitd.ac.in
4 **Department of Computer Science and Engineering, Indian Institute of Technology Delhi, India**
amitk@cse.iitd.ac.in

Abstract

Ashtiani *et al.* proposed a Semi-Supervised Active Clustering framework (SSAC), where the learner is allowed to make adaptive queries to a domain expert. The queries are of the kind "*do two given points belong to the same optimal cluster?*", where the answers to these queries are assumed to be consistent with a unique optimal solution. There are many clustering contexts where such *same cluster* queries are feasible. Ashtiani *et al.* exhibited the power of such queries by showing that any instance of the k-means clustering problem, with additional *margin* assumption, can be solved efficiently if one is allowed to make $O(k^2 \log k + k \log n)$ same-cluster queries. This is interesting since the k-means problem, even with the margin assumption, is NP-hard.

In this paper, we extend the work of Ashtiani *et al.* to the approximation setting by showing that a few of such same-cluster queries enables one to get a polynomial-time $(1+\varepsilon)$-approximation algorithm for the k-means problem without any margin assumption on the input dataset. Again, this is interesting since the k-means problem is NP-hard to approximate within a factor $(1+c)$ for a fixed constant $0 < c < 1$. The number of same-cluster queries used by the algorithm is $\text{poly}(k/\varepsilon)$ which is independent of the size n of the dataset. Our algorithm is based on the D^2-sampling technique, also known as the k-means++ seeding algorithm. We also give a conditional lower bound on the number of same-cluster queries showing that if the Exponential Time Hypothesis (ETH) holds, then any such efficient query algorithm needs to make $\Omega\left(\frac{k}{\text{poly}\log k}\right)$ same-cluster queries. Our algorithm can be extended for the case where the query answers are wrong with some bounded probability. Another result we show for the k-means++ seeding is that a small modification of the k-means++ seeding within the SSAC framework converts it to a constant factor approximation algorithm instead of the well known $O(\log k)$-approximation algorithm.

1998 ACM Subject Classification I.5.3 Clustering

Keywords and phrases k-means, semi-supervised learning, query bounds

Digital Object Identifier 10.4230/LIPIcs.ITCS.2018.40

* A full version of the paper is available at [3], https://arxiv.org/abs/1704.01862

† Nir Ailon and Ragesh Jaiswal acknowledge the support of ISF-UGC India-Israel Joint Research Grant 2014.

‡ Anup Bhattacharya acknowledges the support of TCS graduate fellowship at IIT Delhi.

1 Introduction

Clustering is extensively used in data mining and is typically the first task performed when trying to understand large data. Clustering basically involves partitioning given data into groups or clusters such that data points within the same cluster are similar as per some similarity measure. Clustering is usually performed in an unsupervised setting where data points do not have any labels associated with them. The partitioning is done using some measure of similarity/dissimilarity between data elements. In this paper, we extend the work of Ashtiani *et al.* [6] who explored the possibility of performing clustering in a *semi-supervised active learning* setting for center based clustering problems such as k-median/k-means. In this setting, which they call Semi-Supervised Active Clustering framework or SSAC in short, the clustering algorithm is allowed to make adaptive queries of the form: " *do two points from the dataset belong to the same optimal cluster?*", where query answers are assumed to be consistent with a unique optimal solution. Ashtiani *et al.* [6] started the study of understanding the strength of this model. Do hard clustering problems become easy in this model? They explored such questions in the context of center-based clustering problems. Center based clustering problems such as k-means are extensively used to analyze large data clustering problems. Let us define the k-means problem in the Euclidean setting.

▶ **Definition 1** (k-means problem)**.** Given a dataset $X \subseteq \mathbb{R}^d$ containing n points, and a positive integer k, find a set of k points $C \subseteq \mathbb{R}^d$ (called centers) such that the following cost function is minimized: $\Phi(C, X) = \sum_{x \in X} \min_{c \in C} D(x, c)$, where $D(x, c)$ denotes the squared Euclidean distance between x and c. That is, $D(x, c) = ||x - c||^2$.

Note that the k optimal centers $c_1, \ldots, c_k$ of the k-means problem define k clusters of points in a natural manner. All points for which the closest center is c_i belong to the i^{th} cluster. This is also known as the *Voronoi partitioning* and the clusters obtained in this manner using the optimal k centers are called the optimal clusters. Note that the optimal center for the 1-means problem for any dataset $X \subseteq \mathbb{R}^d$ is the centroid of the dataset denoted by $\mu(X) \stackrel{def.}{=} \frac{\sum_{x \in X} x}{|X|}$. This means that if $X_1, ..., X_k$ are the optimal clusters for the k-means problem on any dataset $X \subseteq \mathbb{R}^d$ and $c_1, ..., c_k$ are the corresponding optimal centers, then $\forall i, c_i = \mu(X_i)$. The k-means problem has been widely studied and various facts are known about this problem. The problem is tractable when either the number k of clusters or the dimension d is equal to 1. However, when $k > 1$ or $d > 1$, then the problem is known to be NP-hard [16, 32, 28]. There has been a number of works of *beyond the worst-case* flavour for the k-means problem in which it is typically assumed that the dataset satisfies some *separation* condition, and then the question is whether this assumption can be exploited to design algorithms with better guarantees for the problem. Such studies have led to different definitions of separation and also some interesting results for datasets that satisfy these separation conditions (e.g., [30, 11, 8]). Ashtiani *et al.* [6] explored one such separation notion that they call the γ-margin property.

▶ **Definition 2** (γ-margin property)**.** Let $\gamma > 1$ be a real number. Let $X \subseteq \mathbb{R}^d$ be any dataset and k be any positive integer. Let $P_X = \{X_1, \ldots, X_k\}$ denote k optimal clusters for the k-means problem. Then this optimal partition of the dataset P_X is said to satisfy the γ-margin property iff for all $i \neq j \in \{1, \ldots, k\}$ and $x \in X_i$ and $y \in X_j$, we have:

$$\gamma \cdot ||x - \mu(X_i)|| < ||y - \mu(X_i)||.$$

Qualitatively this means that every point within some cluster is closer to its own cluster center than a point that does not belong to this cluster. This seems to be a very strong separation

property. Ashtiani *et al.* [6] showed that the k-means clustering problem is NP-hard even when restricted to instances that satisfy the γ-margin property for $\gamma = \sqrt{3.4} \approx 1.84$. Here is the formal statement of their hardness result.

▶ **Theorem 3** (Theorem 10 in [6]). *Finding an optimal solution to the k-means objective function is NP-hard when $k = \Theta(n^{\varepsilon})$ for any $\varepsilon \in (0, 1)$, even when there is an optimal clustering that satisfies the γ-margin property for $\gamma = \sqrt{3.4}$.*

In the context of the k-means problem, the *same-cluster* queries within the SSAC framework are decision questions of the form: *Do points x, y such that $x \neq y$ belong to the same optimal cluster?* [1] Following is the main question explored by Ashtiani *et al.* [6]: *Under the γ-margin assumption, for a fixed $\gamma \in (1, \sqrt{3.4}]$, how many queries must be made in the SSAC framework for k-means to become tractable?*

Ashtiani *et al.* [6] addressed the above question and gave a query algorithm. Their algorithm, in fact, works for a more general setting where the clusters are not necessarily optimal. In the more general setting, there is a *target* clustering $\bar{X} = \bar{X}_1, \ldots, \bar{X}_k$ of the given dataset $X \subseteq \mathbb{R}^d$ such that these clusters satisfy the γ-margin property (i.e., for all $i, x \in \bar{X}_i$, and $y \notin \bar{X}_i, \gamma \cdot ||x - \mu(\bar{X}_i)|| < ||y - \mu(\bar{X}_i)||$), and the goal of the query algorithm is to output the clustering $\bar{X}$. Here is the main result of Ashtiani *et al.*

▶ **Theorem 4** (Theorems 7,8 in [6]). *Let $\delta \in (0, 1)$, $\gamma > 1$. Let $X \subseteq \mathbb{R}^d$ be any dataset with n points, k be a positive integer, and $X_1, \ldots, X_k$ be any target clustering of X that satisfies the γ-margin property. Then there is a query algorithm A that makes $O\left(k \log n + k^2 \frac{\log k + \log 1/\delta}{(\gamma-1)^4}\right)$ same-cluster queries and with probability at least $(1 - \delta)$, outputs the clustering $X_1, \ldots, X_k$. The running time of algorithm A is $O\left(kn \log n + k^2 \frac{\log k + \log 1/\delta}{(\gamma-1)^4}\right)$.*

The above result is a witness to the power of the SSAC framework. We extend this line of work by examining the power of same-cluster queries in the *approximation algorithms* domain. Our results do not assume any separation condition on the dataset (such as γ-margin as in [6]) and they hold for *any* dataset.

Since the k-means problem is NP-hard, an important line of research work has been to obtain approximation algorithms for the problem. There are many efficient approximation algorithms for the k-means problem, for example [25, 26]. Ahmadian *et al.* [2] gave a 6.357-approximation algorithm for the k-means problem. A simple approximation algorithm that gives an $O(\log k)$ approximation guarantee in expectation is the k-means++ seeding algorithm (also known as D^2-sampling algorithm) by Arthur and Vassilvitskii [5]. This algorithm is commonly used in solving the k-means problem in practice. As far as *approximation schemes* or in other words $(1 + \varepsilon)$-approximation algorithms (for arbitrary small $\varepsilon < 1$) are concerned, the following is known: It was shown by Awasthi *et al.* [9] that there is some fixed constant $0 < c < 1$ such that there cannot exist an efficient $(1 + c)$ factor approximation unless P = NP. This result was improved by Lee *et al.* [27] where it was shown that it is NP-hard to approximate the k-means problem within a factor of 1.0013. However, when either k or d is a fixed constant, then there are Polynomial Time Approximation Schemes (PTAS) for the k-means problem.[2] Addad *et al.* [15] and Friggstad *et al.* [19] gave PTASs for the k-means problem in constant dimension. For fixed constant k, various PTASs are known [26, 18, 23, 24]. Following is the main question that we explore in this work:

[1] If the optimal solution is not unique, the same-cluster query answers are assumed to be consistent with respect to any fixed optimal clustering.

[2] This basically means an algorithm that runs in time polynomial in the input parameters but may run in time exponential in $1/\varepsilon$.

For arbitrary small $\varepsilon > 0$, how many same-cluster queries must be made in an efficient $(1+\varepsilon)$-approximation algorithm for k-means in the SSAC framework? The running time should be polynomial in all input parameters such as n, k, d and also in $1/\varepsilon$.

Note that this is a natural extension of the main question explored by Ashtiani *et al.* [6]. Moreover, we have removed the separation assumption on the data. We provide an algorithm that makes $\text{poly}(k/\varepsilon)$ same-cluster queries and runs in time $O(nd \cdot \text{poly}(k/\varepsilon))$. More specifically, here is the formal statement of our main result:

▶ **Theorem 5** (Main result: query algorithm)**.** *Let $0 < \varepsilon \leq 1/2$. Let $X \subseteq \mathbb{R}^d$, and k be any positive integer. Then there is a query algorithm* A *that runs in time $O(ndk^9/\varepsilon^4)$, and with probability at least 0.99, outputs a center set C such that $\Phi(C, X) \leq (1+\varepsilon) \cdot \Delta_k(X)$. Moreover, the number of same-cluster queries used by* A *is $O(k^9/\varepsilon^4)$. Here $\Delta_k(X)$ denotes the optimal value of the k-means objective function.*

Note that unlike Theorem 4, our bound on the number of same-cluster queries is independent of the size of the dataset. We find this interesting and the next natural question we ask is whether this bound on the number of same-cluster queries is *tight* in some sense. In other words, does there exist a query algorithm in the SSAC setting that gives $(1+\varepsilon)$-approximation in time polynomial in n, k, d and makes significantly fewer queries than the one given in the theorem above? We answer this question in the negative by establishing a conditional lower bound on the number of same-cluster queries under the assumption that ETH (Exponential Time Hypothesis) [20, 21] holds. The formal statement of our result is given below.

▶ **Theorem 6** (Main result: query lower bound)**.** *If the Exponential Time Hypothesis (ETH) holds, then there exists a constant $c > 1$ such that any c-approximation query algorithm for the k-means problem that runs in time $\text{poly}(n, d, k)$ makes at least $\frac{k}{\text{poly}\log k}$ queries.*

1.1 Faulty query setting

The existence of a same-cluster oracle that answers such queries perfectly may be too strong an assumption. A more reasonable assumption is the existence of a *faulty* oracle that can answer incorrectly but only with bounded probability. Our query approximation algorithm can be extended to the setting where answers to the same-cluster queries are *faulty*. More specifically, we can get wrong answers to queries independently but with probability at most some constant $q < 1/2$. Also note that in our model the answer for a same-cluster query does not change with repetition. This means that one cannot ask the same query multiple times and amplify the probability of correctness. We obtain $(1+\varepsilon)$-approximation guarantee for the k-means clustering problem in this setting. The main result is given as follows.

▶ **Theorem 7.** *Let $0 < \varepsilon \leq 1/2$. Let $X \subseteq \mathbb{R}^d$, and k be any positive integer. Consider a faulty SSAC setting where the response to every same-cluster query is incorrect with probability at most some constant $q < 1/2$. In such a setting, there is a query algorithm A^E that with probability at least 0.99, outputs a center set C such that $\Phi(C, X) \leq (1+\varepsilon) \cdot \Delta_k(X)$. Moreover, the number of same-cluster queries used by A^E is $O(k^{15}/\varepsilon^8)$.*

The previous theorems summarise the main results of this work which basically explores the power of same-cluster queries in designing fast $(1+\varepsilon)$-approximation algorithms for the k-means problem. We will give the proofs of the above theorems in Sections 3, 4, and 5. There are some other simple and useful contexts, where the SSAC framework gives extremely nice results. One such context is the popular k-means++ seeding algorithm. This is an

extremely simple sampling based algorithm for the k-means problem that samples k centers in a sequence of k iterations. We show that within the SSAC framework, a small modification of this sampling algorithm converts it to one that gives constant factor approximation instead of $O(\log k)$-approximation [5] that is known. This is another witness to the power of same-cluster queries. We discuss this result in Section 2. Some of the basic techniques involved in proving our main results will be introduced while discussing this simpler context.

1.2 Other related work

Clustering problems have been studied in various semi-supervised settings. Basu *et al.* [12] explored *must-link* and *cannot-link* constraints in their semi-supervised clustering formulation. In their framework, must-link and cannot-link constraints were provided explicitly as part of the input along with the cost of violating these constraints. They gave an active learning formulation for clustering in which an oracle answers whether two query points belong to the same cluster or not, and gave a clustering algorithm using these queries. However, they work with a different objective function and there is no discussion on theoretical bounds on the number of queries. In contrast, in our work we consider the k-means objective function and provide bounds on approximation guarantee, required number of adaptive queries, and the running time. Balcan and Blum [10] proposed an interactive framework for clustering with *split/merge* queries. Given a clustering $C = \{C_1, \ldots\}$, a user provides feedback by specifying that some cluster C_l should be split, or clusters C_i and C_j should be merged. Awasthi *et al.* [7] studied a local interactive algorithm for clustering with split and merge feedbacks. Voevodski *et al.* [34] considered *one versus all* queries where query answer for a point $s \in X$ returns distances between s to all points in X. For a k-median instance satisfying (c, ε)-approximation stability property [11], the authors found a clustering close to true clustering using only $O(k)$ one versus all queries. Vikram and Dasgupta [33] designed an interactive bayesian hierarchical clustering algorithm. Given dataset X, the algorithm starts with a candidate hierarchy T, and an initially empty set C of constraints. The algorithm queries user with a subtree $T|_S$ of hierarchy T restricted to constant sized set $S \subset X$ of leaves. User either accepts $T|_S$ or provides a counterexample triplet $(\{a, b\}, c)$ which the algorithm adds to its set of constraints C, and updates T. They consider both random and adaptive ways to select S to query $T|_S$.

1.3 Our Techniques

We now give a brief outline of the new ideas needed for our results. Many algorithms for the k-means problem proceed by iteratively finding approximations to the optimal centers. One such popular algorithm is the k-means++ seeding algorithm [5]. In this algorithm, one builds a set of potential centers iteratively. We start with a set C initialized to the empty set. At each step, we sample a point with probability proportional to the square of the distance from C, and add it to C. Arthur and Vassilvitskii [5] showed that if we continue this process till $|C|$ reaches k, then the corresponding k-means solution has expected cost $O(\log k)$ times the optimal k-means cost. Aggarwal *et al.* [1] showed that if we continue this process till $|C|$ reaches βk, for some constant $\beta > 1$, then the corresponding k-means solution (where we actually open all the centers in C) has cost which is within constant factor of the optimal k-means cost with high probability. Ideally, one would like to stop when size of C reaches k and obtain a constant factor approximation guarantee. We know from previous works [5, 14, 13] that this is not possible in the classical (unsupervised) setting. In this work, we show that one can get such a result in the SSAC framework. A high-level

way of analysing the k-means++ seeding algorithm is as follows. We first observe that if we randomly sample a point from a cluster, then the expected cost of assigning all points of this cluster to the sampled point is within a constant factor of the cost of assigning all the points to the mean of this cluster. Therefore, it suffices to select a point chosen uniformly at random from each of the clusters. Suppose the set C contains such samples for the first i clusters (of an optimal solution). If the other clusters are far from these i clusters, then it is likely that the next point added to C belongs to a new cluster (and perhaps is close to a uniform sample). However to make this more probable, one needs to add several points to C. Further, the number of samples that needs to be added to C starts getting worse as the value of i increases. Therefore, the algorithm needs to build C till its size becomes $O(k \log k)$. In the SSAC framework, we can *tell* if the next point added in C belongs to a new cluster or not. Therefore, we can always ensure that $|C|$ does not exceed k. To make this idea work, we need to extend the induction argument of Arthur and Vassilvitskii [5] – details are given in Section 2.

We now explain the ideas for the PTAS for k-means. We consider the special case of $k = 2$. Let X_1 and X_2 denote the optimal clusters with X_1 being the larger cluster. Inaba *et al.* [22] showed that if we randomly sample about $O(1/\varepsilon)$ points from a cluster, and let μ' denote the mean of this subset of sampled points, then the cost of assigning all points in the cluster to μ' is within $(1 + \varepsilon)$ of the cost of assigning all these points to their actual mean (whp). Therefore, it is enough to get uniform samples of size about $O(1/\varepsilon)$ from each of the clusters. Jaiswal *et al.* [23] had the following approach for obtaining a $(1 + \varepsilon)$-approximation algorithm for k-means (with running time being $nd \cdot f(k, \varepsilon)$, where f is an exponential function of k/ε) – suppose we sample about $O(1/\varepsilon^2)$ points from the input, call this sample S. It is likely to contain at least $O(1/\varepsilon)$ from X_1, but we do not know which points in S are from X_1. Jaiswal *et al.* addressed this problem by cycling over all subsets of S. In the SSAC model, we can directly partition S into $S \cap X_1$ and $S \cap X_2$ using $|S|$ same-cluster queries. Having obtained such a sample S, we can get a close approximation to the mean of X_1. So assume for sake of simplicity that we know μ_1, the mean of X_1. Now we are faced with the problem of obtaining a uniform sample from X_2. The next idea of Jaiswal *et al.* is to sample points with probability proportional to square of distance from μ_1. This is known as D^2-sampling. Suppose we again sample about $O(1/\varepsilon^2)$ such points, call this sample S'. Assuming that the two clusters are far enough (otherwise the problem only gets easier), they show that S' will contain about $O(1/\varepsilon^2)$ points from X_2 (with good probability). Again, in the SSAC model, we can find this subset by $|S'|$ queries – call this set S''. However, the problem is that S'' may not represent a uniform sample from X_2. For any point $e \in X_2$, let p_e denote the conditional probability of sampling e given that a point from X_2 is sampled when sampled using D^2-sampling. They showed p_e is at least $\frac{\varepsilon}{m}$, where m denotes the size of X_2. In order for the sampling lemma of Inaba *et al.* [22] to work, we cannot work with approximately uniform sampling. The final trick of Jaiswal *et al.* was to show that one can in fact get a uniform sample of size about $O(\varepsilon|S''|) = O(1/\varepsilon)$ from S''. The idea is as follows – for every element $e \in S''$, we *retain* it with probability $\frac{\varepsilon}{p_e m}$ (which is at most 1), otherwise we remove it from S''. It is not difficult to see that this gives a uniform sample from X_2. The issue is that we do not know m. Jaiswal *et al.* again cycle over all subsets of S' – we know that there is a (large enough) subset of S' which will behave like a uniform sample from X_2. In the SSAC framework, we first identify the subset of S' which belongs to X_2, call this S'' (as above). Now we prune some points from S'' such that the remaining points behave like a uniform sample. This step is non-trivial because as indicated above, we do not know the value m. Instead, we first show that p_e lies between $\frac{\varepsilon}{m}$ and $\frac{2}{m}$ for most of the

Table 1 k-means++ seeding algorithm (left) and its adaptation in the SSAC setting (right).

k-means++(X,k)	Query-k-means++(X,k)
- Randomly sample a point $x \in X$ - $C \leftarrow \{x\}$ - for $i = 2$ to k - Sample $x \in X$ using distribution D, where $D(x) = \frac{\Phi(C,\{x\})}{\Phi(C,X)}$ - $C \leftarrow C \cup \{x\}$ - return(C)	- Randomly sample a point $x \in X$ - $C \leftarrow \{x\}$ - for $i = 2$ to k - for $j = 1$ to $\lceil \log k \rceil$ - Sample $x \in X$ using distribution D, where $D(x) = \frac{\Phi(C,\{x\})}{\Phi(C,X)}$ - if (NewCluster(C,x)) {$C \leftarrow C \cup \{x\}$; break} - return$(C)$
	NewCluster(C,x) - If ($\exists c \in C$ s.t. SameCluster(c,x)) return(false) - else return(true)

points of X_2. Therefore, S'' is likely to contain such a *nice* point, call it v. Now, for every point $e \in S''$, we retain it with probability $\frac{\varepsilon p_e}{2p_v}$ (which we know is at most 1). This gives a uniform sample of sufficiently large size from X_2. For k larger than 2, we generalize the above ideas using a non-trivial induction argument.

2 k-means++ within SSAC framework

The k-means++ seeding algorithm, also known as the D^2-sampling algorithm, is a simple sampling procedure that samples k centers in k iterations. The description of this algorithm is given below.

The algorithm picks the first center randomly from the set X of points and after having picked the first $(i-1)$ centers denoted by C_{i-1}, it picks a point $x \in X$ to be the i^{th} center with probability proportional to $\min_{c \in C_{i-1}} ||x-c||^2$. The running time of k-means++ seeding algorithm is clearly $O(nkd)$. Arthur and Vassilvitskii [5] showed that this simple sampling procedure gives an $O(\log k)$ approximation in expectation for any dataset. Within the SSAC framework where the algorithm is allowed to make same-cluster queries, we can make a tiny addition to the k-means++ seeding algorithm to obtain a query algorithm that gives constant factor approximation guarantee and makes only $O(k^2 \log k)$ same-cluster queries. The description of the query algorithm is given in Table 1 (see right). In iteration $i > 1$, instead of simply accepting the sampled point x as the i^{th} center (as done in k-means++ seeding algorithm), the sampled point x is accepted only if it belongs to a cluster other than those to which centers in C_{i-1} belong (if this does not happen, the sampling is repeated for at most $\lceil \log k \rceil$ times). Here is the main result that we show for the query-k-means++ algorithm.

▶ **Theorem 8.** *Let $X \subseteq \mathbb{R}^d$ be any dataset containing n points and $k > 1$ be a positive integer. Let C denote the output of the algorithm* Query-k-means++(X,k). *Then*

$$\mathbf{E}[\Phi(C,X)] \leq 24 \cdot \Delta_k(X),$$

where $\Delta_k(X)$ denotes the optimal k-means cost on dataset X. Furthermore, the algorithm makes $O(k^2 \log k)$ same-cluster queries and runs in time $O(nkd + k \log k \log n + k^2 \log k)$.

The bound on the number of same-cluster queries is trivial from the algorithm description. For the running time, it takes $O(nd)$ time to update the distribution D which is updated k times. This accounts for the $O(nkd)$ term in the running time. Sampling an element from a distribution D takes $O(\log n)$ time (if we maintain the cumulative distribution etc.) and at most $O(k \log k)$ points are sampled. Moreover, determining whether a sampled point belongs to an uncovered cluster takes $O(k)$ time. So, the overall running time of the algorithm is $O(nkd + k \log k \log n + k^2 \log k)$. The proof of the above theorem may be found in the full version of the paper available at [3].

3 Query Approximation Algorithm (proof of Theorem 5)

As mentioned in the introduction, our query algorithm is based on the D^2-sampling based algorithm of Jaiswal *et al.* [23, 24]. The algorithm in these works give a $(1+\varepsilon)$-factor approximation for arbitrary small $\varepsilon > 0$. The running time of these algorithms are of the form $nd \cdot f(k, \varepsilon)$, where f is an exponential function of k/ε. We now show that it is possible to get a running time which is polynomial in $n, k, d, 1/\varepsilon$ in the SSAC model. The main ingredient in the design and analysis of the sampling algorithm is the following lemma by Inaba *et al.* [22].

▶ **Lemma 9** ([22]). *Let S be a set of points obtained by independently sampling M points uniformly at random with replacement from a point set $X \subset \mathbb{R}^d$. Then for any $\delta > 0$,*

$$\mathbf{Pr}\left[\Phi(\{\mu(S)\}, X) \leq \left(1 + \frac{1}{\delta M}\right) \cdot \Delta_1(X)\right] \geq (1 - \delta).$$

Here $\mu(S)$ denotes the geometric centroid of the set S. That is $\mu(S) = \frac{\sum_{s \in S} s}{|S|}$

Our algorithm `Query-`k`-means` is described in Table 2. It maintains a set C of potential centers of the clusters. In each iteration of step **(3)**, it adds one more candidate center to the set C (whp), and so, the algorithm stops when $|C|$ reaches k. For sake of explanation, assume that optimal clusters are $X_1, X_2, \ldots, X_k$ with means $\mu_1, \ldots, \mu_k$ respectively. Consider the i^{th} iteration of step **(3)**. At this time $|C| = i - 1$, and it has good approximations to means of $i-1$ clusters among $X_1, \ldots . X_k$. Let us call these clusters *covered*. In Step **(3.1)**, it samples N points, each with probability proportional to square of distance from C (D^2-sampling). Now, it partitions this set, S, into $S \cap X_1, \ldots, S \cap X_k$ in the procedure `UncoveredCluster`, and then picks the partition with the largest size such that the corresponding optimal cluster X_j is not one of the $(i-1)$ covered clusters. Now, we would like to get a uniform sample from X_j – recall that $S \cap X_j$ does not represent a uniform sample. However, as mentioned in the introduction, we need to find an element s of X_j for which the probability of being in sampled is small enough. Therefore, we pick the element in $S \cap X_j$ for which this probability is smallest (and we will show that it has the desired properties). The procedure `UncoveredCluster` returns this element s. Finally, we choose a subset T of $S \cap X_j$ in the procedure `UniformSample`. This procedure is designed such that each element of X_j has the same probability of being in T. In step **(3.4)**, we check whether the multi-set T is of a desired minimum size. We will argue that the probability of T not containing sufficient number of points is very small. If we have T of the desired size, we take its mean and add it to C in Step **(3.6)**.

We now formally prove the approximation guarantee of the `Query-`k`-means` algorithm.

Table 2 Approximation algorithm for k-means(top-left frame). Note that $\mu(T)$ denotes the centroid of T and D^2-sampling w.r.t. empty center set C means just uniform sampling. The algorithm `UniformSample`(X, C, s) (bottom-left) returns a uniform sample of size $\Omega(1/\varepsilon)$ (w.h.p.) from the optimal cluster to which point s belongs.

Constants: $N = \frac{(2^{12})k^3}{\varepsilon^2}$, $M = \frac{64k}{\varepsilon}$, $L = \frac{(2^{23})k^2}{\varepsilon^4}$

`Query-`k`-means`(X, k, ε)

(1) $R \leftarrow \emptyset$

(2) $C \leftarrow \emptyset$

(3) for $i = 1$ to k

(3.1) D^2-sample a multi-set S of N points from X with respect to center set C

(3.2) $s \leftarrow$ `UncoveredCluster`(C, S, R)

(3.3) $T \leftarrow$ `UniformSample`(X, C, s)

(3.4) If $(|T| < M)$ continue

(3.5) $R \leftarrow R \cup \{s\}$

(3.6) $C \leftarrow C \cup \mu(T)$

(4) return(C)

`UniformSample`(X, C, s)

- $T \leftarrow \emptyset$
- For $i = 1$ to L:
  - D^2-sample a point x from X with respect to center set C
  - If (`SameCluster`(s, x))
    - With probability $\left(\frac{\varepsilon}{128} \cdot \frac{\Phi(C,\{s\})}{\Phi(C,\{x\})}\right)$ add x in multi-set T
- return(T)

`UncoveredCluster`(C, S, R)

- For all $i \in \{1, ..., k\}$: $S_i \leftarrow \emptyset$
- $i \leftarrow 1$
- For all $y \in R$: $\{S_i \leftarrow y;\ i{+}{+}\}$
- For all $s \in S$:
  - If $(\exists j, y$ s.t. $y \in S_j$ & `SameCluster`$(s, y))$
    - $S_j \leftarrow S_j \cup \{s\}$
  - else
    - Let i be any index s.t. S_i is empty
    - $S_i \leftarrow \{s\}$.
- Let S_i be the largest set such that $i > |R|$.
- Let $s \in S_i$ be the element with smallest
  - value of $\Phi(C, \{s\})$ in S_i.
- return(s)

▶ **Theorem 10.** *Let $0 < \varepsilon \leq 1/2$. Let $X \subseteq \mathbb{R}^d$, and k be any positive integer. There exists an algorithm that runs in time $O(ndk^9/\varepsilon^4)$, and with probability at least $\frac{1}{4}$, outputs a center set C such that $\Phi(C, X) \leq (1 + \varepsilon) \cdot \Delta_k(X)$. Moreover, the algorithm makes $O(k^9/\varepsilon^4)$ same-cluster queries.*

Note that the success probability of the algorithm may be boosted by repeating it a constant number of times. This will also prove our main theorem (that is, Theorem 5). We will assume that the dataset X satisfies (k, ε)-irreducibility property defined next. We will later drop this assumption using a simple argument and show that the result holds for *all* datasets. This property was also used in some earlier works [26, 23].

▶ **Definition 11** (Irreducibility)**.** Let k be a positive integer and $0 < \varepsilon \leq 1$. A given dataset $X \subseteq \mathbb{R}^d$ is said to be (k, ε)-irreducible iff

$$\Delta_{k-1}(X) \geq (1 + \varepsilon) \cdot \Delta_k(X).$$

Qualitatively, what the irreducibility assumption implies is that the optimal solution for the $(k - 1)$-means problem does not give a $(1 + \varepsilon)$-approximation to the k-means problem.

The following useful lemmas are well known facts.

▶ **Lemma 12.** *For any dataset $X \subseteq \mathbb{R}^d$ and a point $c \in \mathbb{R}^d$, we have:*

$$\Phi(\{c\}, X) = \Phi(\mu(X), X) + |X| \cdot ||c - \mu(X)||^2.$$

▶ **Lemma 13** (Approximate Triangle Inequality)**.** *For any three points $p, q, r \in \mathbb{R}^d$, we have*

$$||p - q||^2 \leq 2(||p - r||^2 + ||r - q||^2)$$

Let $X_1, ..., X_k$ be optimal clusters of the dataset X for the k-means objective. Let $\mu_1, ..., \mu_k$ denote the corresponding optimal k centers. That is, $\forall i, \mu_i = \mu(X_i)$. For all i, let $m_i = |X_i|$. Also, for all i, let $r_i = \frac{\sum_{x \in X_i} ||x - \mu_i||^2}{m_i}$. The following useful lemma holds due to irreducibility. The lemma is the same as Lemma 4 from [23].

▶ **Lemma 14.** *For all $1 \leq i < j \leq k$, $||\mu_i - \mu_j||^2 \geq \varepsilon \cdot (r_i + r_j)$.*

Consider the algorithm `Query-k-means` in Figure 2. Let $C_i = \{c_1, ..., c_i\}$ denote the set of centers at the end of the i^{th} iteration of the for loop. That is, C_i is the same as variable C at the end of iteration i. We will prove Theorem 10 by inductively arguing that for every i, there are i distinct clusters for which centers in C_i are good in some sense. Consider the following invariant:

P(i): There exists a set of i distinct clusters $X_{j_1}, X_{j_2}, ..., X_{j_i}$ such that

$$\forall r \in \{1, ..., i\}, \Phi(\{c_r\}, X_{j_r}) \leq \left(1 + \frac{\varepsilon}{16}\right) \cdot \Delta_1(X_{j_r}).$$

Note that a trivial consequence of $P(i)$ is $\Phi(C_i, X_{j_1} \cup ... \cup X_{j_i}) \leq (1 + \frac{\varepsilon}{16}) \cdot \sum_{r=1}^{i} \Delta_1(X_{j_r})$. We will show that for all i, $P(i)$ holds with probability at least $(1 - 1/k)^i$. Note that the theorem follows if $P(k)$ holds with probability at least $(1 - 1/k)^k$. We will proceed using induction.

The base case $P(0)$ holds since C_0 is the empty set. For the inductive step, assuming that $P(i)$ holds with probability at least $(1 - 1/k)^i$ for some arbitrary $i \geq 0$, we will show that $P(i + 1)$ holds with probability at least $(1 - 1/k)^{i+1}$. We condition on the event $P(i)$ (that is true with probability at least $(1 - 1/k)^i$). Let C_i and $X_{j_1}, ..., X_{j_i}$ be as guaranteed by the invariant $P(i)$. For ease of notation and without loss of generality, let us assume that the index j_r is r. So, C_i gives a good approximation w.r.t. points in the set $X_1 \cup X_2 \cup .... \cup X_i$ and these clusters may be thought of as "covered" clusters (in the approximation sense). Suppose we D^2-sample a point p w.r.t. center set C_i. The probability that p belongs to some "uncovered cluster" X_r where $r \in [i + 1, k]$ is given as $\frac{\Phi(C_i, X_r)}{\Phi(C_i, X)}$. If this quantity is small, then the points sampled using D^2 sampling in subsequent iterations may not be good representatives for the uncovered clusters. This may cause the analysis to break down. However, we argue that since our data is (k, ε)-irreducible, this does not occur. The following lemma is the same as Lemma 5 from [23].

▶ **Lemma 15.** $\frac{\Phi(C_i, X_{i+1} \cup ... \cup X_k)}{\Phi(C_i, X)} \geq \frac{\varepsilon}{4}$.

The following simple corollary of the above lemma will be used in the analysis later.

▶ **Corollary 16.** *There exists an index $j \in \{i + 1, ..., k\}$ such that $\frac{\Phi(C_i, X_j)}{\Phi(C_i, X)} \geq \frac{\varepsilon}{4k}$.*

The above corollary says that there is an uncovered cluster which will have a non-negligible representation in the set S that is sampled in iteration $(i+1)$ of the algorithm `Query-`k`-means`. The next lemma shows that conditioned on sampling from an uncovered cluster $l \in \{i+1, ..., k\}$, the probability of sampling a point x from X_l is at least $\frac{\varepsilon}{64}$ times its sampling probability if it were sampled uniformly from X_l (i.e., with probability at least $\frac{\varepsilon}{64} \cdot \frac{1}{m_l}$). [3]

▶ **Lemma 17.** *For any $l \in \{i+1, ..., k\}$ and $x \in X_l$, $\frac{\Phi(C_i, \{x\})}{\Phi(C_i, X_l)} \geq \frac{\varepsilon}{64} \cdot \frac{1}{m_l}$.*

Proof. Let $t \in \{1, ..., i\}$ be the index such that x is closest to c_t among all centers in C_i. We have:

$$\begin{aligned}
\Phi(C_i, X_l) &= m_l \cdot r_l + m_l \cdot ||\mu_l - c_t||^2 \quad \text{(using Lemma 12)} \\
&\leq m_l \cdot r_l + 2m_l \cdot (||\mu_l - \mu_t||^2 + ||\mu_t - c_t||^2) \quad \text{(using Lemma 13)} \\
&\leq m_l \cdot r_l + 2m_l \cdot (||\mu_l - \mu_t||^2 + \frac{\varepsilon}{16} \cdot r_t) \quad \text{(using invariant and Lemma 12)}
\end{aligned}$$

Also, we have:

$$\begin{aligned}
\Phi(C_i, \{x\}) = ||x - c_t||^2 &\geq \frac{||x - \mu_t||^2}{2} - ||\mu_t - c_t||^2 \quad \text{(using Lemma 13)} \\
&\geq \frac{||\mu_l - \mu_t||^2}{8} - ||\mu_t - c_t||^2 \quad \text{(since } ||x - \mu_t|| \geq ||\mu_l - \mu_t||/2\text{)} \\
&\geq \frac{||\mu_l - \mu_t||^2}{8} - \frac{\varepsilon}{16} \cdot r_t \quad \text{(using invariant and Lemma 12)} \\
&\geq \frac{||\mu_l - \mu_t||^2}{16} \quad \text{(using Lemma 14)}
\end{aligned}$$

Combining the inequalities obtained above, we get the following:

$$\begin{aligned}
\frac{\Phi(C_i, \{x\})}{\Phi(C_i, X_l)} &\geq \frac{||\mu_l - \mu_t||^2}{16 \cdot m_l \cdot \left(r_l + 2||\mu_l - \mu_t||^2 + \frac{\varepsilon}{8} \cdot r_t\right)} \\
&\geq \frac{1}{16 \cdot m_l} \cdot \frac{1}{(1/\varepsilon) + 2 + (1/8)} \geq \frac{\varepsilon}{64} \cdot \frac{1}{m_l}
\end{aligned}$$

This completes the proof of the lemma. ◀

With the above lemmas in place, let us now get back to the inductive step of the proof. Let $J \subseteq \{i+1, ..., k\}$ denote the subset of indices (from the uncovered cluster indices) such that $\forall j \in J, \frac{\Phi(C_i, X_j)}{\Phi(C_i, X)} \geq \frac{\varepsilon}{8k}$. For any index $j \in J$, let $Y_j \subseteq X_j$ denote the subset of points in X_j such that $\forall y \in Y_j, \frac{\Phi(C_i, \{y\})}{\Phi(C_i, X_j)} \leq \frac{2}{m_j}$. That is, Y_j consists of all the points such that the conditional probability of sampling any point y in Y_j, given that a point is sampled from X_j, is upper bounded by $2/m_j$. Note that from Lemma 17, the conditional probability of sampling a point x from X_j, given that a point is sampled from X_j, is lower bounded by $\frac{\varepsilon}{64} \cdot \frac{1}{m_j}$. This gives the following simple and useful lemma.

▶ **Lemma 18.** *For all $j \in \{i+1, ..., k\}$ the following two inequalities hold:*
1. *$\frac{\Phi(C_i, Y_j)}{\Phi(C_i, X)} \geq \frac{\varepsilon}{128} \cdot \frac{\Phi(C_i, X_j)}{\Phi(C_i, X)}$, and*
2. *For any $y \in Y_j$ and any $x \in X_j$, $\frac{\varepsilon}{128} \cdot \Phi(C_i, \{y\}) \leq \Phi(C_i, \{x\})$.*

[3] This is Lemma 6 from [23]. We give the proof for self-containment.

Proof. Inequality (1) follows from the fact that $|Y_j| \geq m_j/2$, and $\frac{\Phi(C_i,\{y\})}{\Phi(C_i,X_j)} \geq \frac{\varepsilon}{64} \cdot \frac{1}{m_j}$ for all $y \in X_j$. Inequality (2) follows from the fact that for all $x \in X_j$, $\frac{\Phi(C_i,\{x\})}{\Phi(C_i,X_j)} \geq \frac{\varepsilon}{64} \cdot \frac{1}{m_j}$ and for all $y \in Y_j$, $\frac{\Phi(C_i,\{y\})}{\Phi(C_i,X_j)} \leq \frac{2}{m_j}$. ◀

Let us see the outline of our plan before continuing with our formal analysis. What we hope to get in line (3.2) of the algorithm is a point s that belongs to one of the uncovered clusters with index in the set J. That is, s belongs to an uncovered cluster that is likely to have a good representation in the D^2-sampled set S obtained in line (3.1). In addition to s belonging to X_j for some $j \in J$, we would like s to belong to Y_j. This is crucial for the uniform sampling in line (3.3) to succeed. We will now show that the probability of s returned in line (3.2) satisfies the above conditions is large.

▶ **Lemma 19.** *Let S denote the D^2-sample obtained w.r.t. center set C_i in line (3.1) of the algorithm.*

$$\mathbf{Pr}[\exists j \in J \text{ such that } S \text{ does not contain any point from } Y_j] \leq \frac{1}{4k}.$$

Proof. We will first get bound on the probability for a fixed $j \in J$ and then use the union bound. From property (1) of Lemma 18, we have that for any $j \in J$, $\frac{\Phi(C_i,Y_j)}{\Phi(C_i,X)} \geq \frac{\varepsilon}{128} \cdot \frac{\varepsilon}{8k} = \frac{\varepsilon^2}{(2^{10})k}$. Since the number of sampled points is $N = \frac{(2^{12})k^3}{\varepsilon^2}$, we get that the probability that S has no points from Y_j is at most $\frac{1}{4k^2}$. Finally, using the union bound, we get the statement of the lemma. ◀

▶ **Lemma 20.** *Let S denote the D^2-sample obtained w.r.t. center set C_i in line (3.1) of the algorithm and let S_j denote the representatives of X_j in S. Let $max = \arg\max_{j \in \{i+1,\ldots,k\}} |S_j|$. Then $\mathbf{Pr}[max \notin J] \leq \frac{1}{4k}$.*

Proof. From Corollary 16, we know that there is an index $j \in \{i+1,\ldots,k\}$ such that $\frac{\Phi(C_i,X_j)}{\Phi(C_i,X)} \geq \frac{\varepsilon}{4k}$. Let $\alpha = N \cdot \frac{\varepsilon}{4k}$. The expected number of representatives from X_j in S is at least α. So, from Chernoff bounds, we have:

$$\mathbf{Pr}[|S_j| \leq 3\alpha/4] \leq e^{-\alpha/32}$$

On the other hand, for any $r \in \{i+1,\ldots,k\} \setminus J$, the expected number of points in S from X_r is at most $\frac{\varepsilon}{8k} \cdot N = \alpha/2$. So, from Chernoff bounds, we have:

$$\mathbf{Pr}[|S_r| > 3\alpha/4] \leq e^{-\alpha/24}$$

So, the probability that there exists such an r is at most $k \cdot e^{-\alpha/24}$ by union bound. Finally, the probability that $max \notin J$ is at most $\mathbf{Pr}[|S_j| \leq 3\alpha/4] + \mathbf{Pr}[\exists r \in \{i+1,\ldots,k\} \setminus J | |S_r| > 3\alpha/4]$ which is at most $\frac{1}{4k}$ due to our choice of $N = \frac{(2^{12})k^3}{\varepsilon^2}$. ◀

From the previous two lemmas, we get that with probability at least $(1 - \frac{1}{2k})$, the s returned in line (3.2) belongs to Y_j for some $j \in J$. Finally, we will need the following claim to argue that the set T returned in line (3.3) is a uniform sample from one of the sets X_j for $j \in \{i+1,\ldots,k\}$.

▶ **Lemma 21.** *Let S denote the D^2-sample obtained w.r.t. center set C_i in line (3.1) and s be the point returned in line (3.2) of the algorithm. Let j denote the index of the cluster to which s belongs. If $j \in J$ and $s \in Y_j$, then with probability at least $(1 - \frac{1}{4k})$, T returned in line (3.3) is a uniform sample from X_j with size at least $\frac{64k}{\varepsilon}$.*

Proof. Consider the call to sub-routine `UniformSample`(X, C_i, s) with s as given in the statement of the lemma. If j is the index of the cluster to which s belongs, then $j \in J$ and $s \in Y_j$. Let us define L random variables $Z_1, ..., Z_L$ one for every iteration of the sub-routine. These random variables are defined as follows: for any $r \in [1, L]$, if the sampled point x belongs to the same cluster as s and it is picked to be included in multi-set S, then $Z_r = x$, otherwise $Z_r = \perp$. We first note that for any r and any $x, y \in X_j$, $\mathbf{Pr}[Z_r = x] = \mathbf{Pr}[Z_r = y]$. This is because for any $x \in X_j$, we have $\mathbf{Pr}[Z_r = x] = \frac{\Phi(C_i,\{x\})}{\Phi(C_i,X)} \cdot \frac{\frac{\varepsilon}{128}\cdot\Phi(C_i,\{s\})}{\Phi(C_i,\{x\})} = \frac{\varepsilon}{128} \cdot \frac{\Phi(C_i,\{s\})}{\Phi(C_i,X)}$. It is important to note that $\frac{\frac{\varepsilon}{128}\cdot\Phi(C_i,\{s\})}{\Phi(C_i,\{x\})} \leq 1$ from property (2) of Lemma 18 and hence valid in the probability calculations above.

Let us now obtain a bound on the size of T. Let $T_r = I(Z_r)$ be the indicator variable that is 1 if $Z_r \neq \perp$ and 0 otherwise. Using the fact that $j \in J$, we get that for any r:

$$\mathbf{E}[T_r] = \mathbf{Pr}[T_r = 1] = \frac{\varepsilon}{128} \cdot \frac{\sum_{x \in X_j} \Phi(C_i, \{s\})}{\Phi(C_i, X)} \geq \frac{\varepsilon}{128} \cdot \frac{\varepsilon}{8k} \cdot \frac{\varepsilon}{64} = \frac{\varepsilon^3}{(2^{16})k}.$$

Given that $L = \frac{2^{23}k^2}{\varepsilon^4}$, applying Chernoff bounds, we get the following:

$$\mathbf{Pr}\left[|T| \geq \frac{64k}{\varepsilon}\right] = 1 - \mathbf{Pr}\left[|T| \leq \frac{64k}{\varepsilon}\right] \geq \left(1 - \frac{1}{4k}\right)$$

This completes the proof of the lemma. ◀

Since a suitable s (as required by the above lemma) is obtained in line (3.2) with probability at least $(1 - \frac{1}{2k})$, the probability that T obtained in line (3.3) is a uniform sample from some uncovered cluster X_j is at least $(1 - \frac{1}{2k}) \cdot (1 - \frac{1}{4k})$. Finally, the probability that the centroid $\mu(T)$ of the multi-set T that is obtained is a good center for X_j is at least $(1 - \frac{1}{4k})$ using Inaba's lemma. Combining everything, we get that with probability at least $(1 - \frac{1}{k})$ an uncovered cluster will be covered in the i^{th} iteration. This completes the inductive step and hence the approximation guarantee of $(1 + \varepsilon)$ holds for any dataset that satisfies the (k, ε)-irreducibility assumption. For the number of queries and running time, note that every time sub-routine `UncoveredCluster` is called, it uses at most kN same cluster queries. For the subroutine `UniformSample`, the number of same-cluster queries made is L. So, the total number of queries is $O(k(kN + L)) = O(k^5/\varepsilon^4)$. More specifically, we have proved the following theorem.

▶ **Theorem 22.** *Let $0 < \varepsilon \leq 1/2$, k be any positive integer, and $X \subseteq \mathbb{R}^d$ such that X is (k, ε)-irreducible. Then* `Query-k-means`(X, k, ε) *runs in time $O(ndk^5/\varepsilon^4)$ and with probability at least $(1/4)$ outputs a center set C such that $\Phi(C, X) \leq (1+\varepsilon) \cdot \Delta_k(X)$. Moreover, the number of same-cluster queries used by* `Query-k-means`(X, k, ε) *is $O(k^5/\varepsilon^4)$.*

To complete the proof of Theorem 10, we must remove the irreducibility assumption in the above theorem. We do this by considering the following two cases:

1. Dataset X is $(k, \frac{\varepsilon}{(4+\varepsilon/2)k})$-irreducible.
2. Dataset X is not $(k, \frac{\varepsilon}{(4+\varepsilon/2)k})$-irreducible.

In the former case, we can apply Theorem 22 to obtain Theorem 10. Now, consider the latter case. Let $1 < i \leq k$ denote the largest index such that X is $(i, \frac{\varepsilon}{(1+\varepsilon/2)k})$-irreducible, otherwise $i = 1$. Then we have:

$$\Delta_i(X) \leq \left(1 + \frac{\varepsilon}{(4 + \varepsilon/2)k}\right)^{k-i} \cdot \Delta_k(X) \leq \left(1 + \frac{\varepsilon}{4}\right) \cdot \Delta_k(X).$$

This means that a $(1+\varepsilon/4)$-approximation for the i-means problem on the dataset X gives the desired approximation for the k-means problem. Note that our approximation analysis works for the i-means problem with respect to the algorithm being run only for i steps in line (3) (instead of k). That is, the centers sampled in the first i iterations of the algorithm give a $(1+\varepsilon/16)$-approximation for the i-means problem for any fixed i. This simple observation is sufficient for Theorem 10.

Note since Theorem 22 is used with value of the error parameter as $O(\varepsilon/k)$, the bounds on the query and running time get multiplied by a factor of k^4.

4 Query Lower Bound (proof of Theorem 6)

In this section, we will obtain a conditional lower bound on the number of same-cluster queries assuming the Exponential Time Hypothesis (ETH). This hypothesis has been used to obtain lower bounds in various different contexts (see [29] for reference). We start by stating the Exponential Time Hypothesis (ETH).

▶ **Hypothesis 23** (Exponential Time Hypothesis (ETH)[20, 21])**.** *There does not exist an algorithm that can decide whether any 3-SAT formula with m clauses is satisfiable with running time $2^{o(m)}$.*

Since we would like to obtain lower bounds in the approximation domain, we will need a *gap* version of the above ETH hypothesis. The following version of the PCP theorem will be very useful in obtaining a gap version of ETH.

▶ **Theorem 24** (Dinur's PCP Theorem [17])**.** *For some constants $\varepsilon, d > 0$, there exists a polynomial time reduction that takes a 3-SAT formula ψ with m clauses and produces another 3-SAT formula ϕ with $m' = O(m \text{ polylog } m)$ clauses such that:*

- *If ψ is satisfiable, then ϕ is satisfiable,*
- *if ψ is unsatisfiable, then $val(\phi) \leq 1-\varepsilon$, and*
- *each variable of ϕ appears in at most d clauses.*

Here $val(\phi)$ denotes the maximum fraction of clauses of ϕ that are satisfiable by any assignment.

The following new hypothesis follows from ETH and will be useful in our analysis.

▶ **Hypothesis 25.** *There exists constants $\varepsilon, d > 0$ such that the following holds: There does not exist an algorithm that, given a 3-SAT formula ψ with m clauses with each variable appearing in at most d clauses, distinguishes whether ψ is satisfiable or $val(\psi) \leq (1-\varepsilon)$, runs in time $2^{\Omega(\frac{m}{poly \log m})}$.*

The following simple lemma follows from Dinur's PCP theorem given above.

▶ **Lemma 26.** *If Hypothesis 23 holds, then so does Hypothesis 25.*

We now see a reduction from the gap version of 3-SAT to the gap version of the Vertex Cover problem that will be used to argue the hardness of the k-means problem. The next result is a standard reduction and can be found in a survey by Luca Trevisan [31].

▶ **Lemma 27.** *Let $\varepsilon, d > 0$ be some constants. There is a polynomial time computable function mapping 3-SAT instances ψ with m variables and where each variable appears in at most d clauses, into graphs G_ψ with $3m$ vertices and maximum degree $O(d)$ such that if ψ is satisfiable, then G_ψ has a vertex cover of size at most $2m$ and if $val(\psi) \leq (1-\varepsilon)$, then every vertex cover of G_ψ has size at least $2m(1+\varepsilon/2)$.*

We formulate the following new hypothesis that holds given that hypothesis 25 holds. Eventually, we will chain all these hypothesis together.

▶ **Hypothesis 28.** *There exists constants $\varepsilon, d > 0$ such that the following holds: There does not exist an algorithm that, given a graph G with n vertices and maximum degree d, distinguishes between the case when G has a vertex cover of size at most $2n/3$ and the case when G has a vertex cover of size at least $\frac{2n}{3} \cdot (1+\varepsilon)$, runs in time $2^{\Omega(\frac{n}{poly \log n})}$.*

The following lemma is a simple implication of Lemma 27

▶ **Lemma 29.** *If Hypothesis 25 holds, then so does Hypothesis 28.*

We are getting closer to the k-means problem that has a reduction from the vertex cover problem on triangle-free graphs [9]. So, we will need reductions from vertex cover problem to vertex cover problem on triangle-free graphs and then to the k-means problem. These two reductions are given by Awasthi *et al.* [9].

▶ **Lemma 30** (Follows from Theorem 21 [9])**.** *Let $\varepsilon, d > 0$ be some constants. There is a polynomial-time computable function mapping any graph $G = (V, E)$ with maximum degree d to a triangle-free graph $\hat{G} = (\hat{V}, \hat{E})$ such that the following holds:*

- *$|\hat{V}| = poly(d, 1/\varepsilon) \cdot |V|$ and maximum degree of vertices in $\hat{G}$ is $O(d^3/\varepsilon^2)$, and*
- *$\left(1 - \frac{|VC(G)|}{|V|}\right) \leq \left(1 - \frac{|VC(\hat{G})|}{|\hat{V}|}\right) \leq (1+\varepsilon) \cdot \left(1 - \frac{|VC(G)|}{|V|}\right)$.*

Here $VC(G)$ denote the size of the minimum vertex cover of G.

We can formulate the following hypothesis that will follow from Hypothesis 28 using the above lemma.

▶ **Hypothesis 31.** *There exists constants $\varepsilon, d > 0$ such that the following holds: There does not exist an algorithm that, given a triangle-free graph G with n vertices and maximum degree d, distinguishes between the case when G has a vertex cover of size at most $\frac{2n}{3}$ and the case when G has a vertex cover of size at least $\frac{2n}{3} \cdot (1+\varepsilon)$, runs in time $2^{\Omega(\frac{n}{poly \log n})}$.*

The next claim is a simple application of Lemma 30.

▶ **Lemma 32.** *If Hypothesis 28 holds, then so does Hypothesis 31.*

Finally, we use the reduction from the vertex cover problem in triangle-free graphs to the k-means problem to obtain the hardness result for the k-means problem. We will use the following reduction from Awasthi *et al.* [9].

▶ **Lemma 33** (Theorem 3 [9])**.** *There is an efficient reduction from instances of Vertex Cover (in triangle free graphs) to those of k-means that satisfies the following properties:*

- *if the Vertex Cover instance has value k, then the k-means instance has cost at most $(m-k)$*
- *if the Vertex Cover instance has value at least $k(1+\varepsilon)$, then the optimal k-means cost is at least $m - (1 - \Omega(\varepsilon))k$. Here ε is some fixed constant > 0.*

Here m denotes the number of edges in the vertex cover instance.

The next hypothesis follows from Hypothesis 31 due to the above lemma.

▶ **Hypothesis 34.** *There exists constant $c > 1$ such that the following holds: There does not exist an algorithm that gives an approximation guarantee of c for the k-means problem that runs in time $poly(n, d) \cdot 2^{\Omega(\frac{k}{poly \log k})}$.*

▶ **Lemma 35.** *If Hypothesis 31 holds, then so does Hypothesis 34.*

Now using Lemmas 26, 29, 32, and 35, get the following result.

▶ **Lemma 36.** *If the Exponential Time Hypothesis (ETH) holds then there exists a constant $c > 1$ such that any c-approximation algorithm for the k-means problem cannot have running time better than $poly(n, d) \cdot 2^{\Omega(\frac{k}{poly \log k})}$.*

This proves Theorem 6 since if there is a query algorithm that runs in time $poly(n, d, k)$ and makes $\frac{k}{poly \log k}$ same-cluster queries, then we can convert it to a non-query algorithm that runs in time $poly(n, d) \cdot 2^{\frac{k}{poly \log k}}$ in a brute-force manner by trying out all possible answers for the queries and then picking the best k-means solution.

5 Query Approximation Algorithm with Faulty Oracle

In this section, we describe how to extend our approximation algorithm for k-means clustering in the SSAC framework when the query oracle is *faulty*. That is, the answers to the same-cluster queries may be incorrect. Let us denote the faulty same-cluster oracle as $\mathcal{O}^E$. We consider the following error model: for a query with points u and v, the query answer $\mathcal{O}^E(u, v)$ is wrong independently with probability at most q where q is a constant strictly less than $1/2$. More specifically, if u and v belong to the same optimal cluster, then $\mathcal{O}^E(u, v) = 1$ with probability at least $(1 - q)$ and $\mathcal{O}^E(u, v) = 0$ with probability at most q. Similarly, if u and v belong to different optimal clusters, then $\mathcal{O}^E(u, v) = 1$ with probability at most q and $\mathcal{O}^E(u, v) = 0$ with probability at least $(1 - q)$.

The modified algorithm giving $(1 + \varepsilon)$-approximation for k-means with faulty oracle $\mathcal{O}^E$ is given in Table 3. Let $X_1, \ldots, X_k$ denote the k optimal clusters for the dataset X. Let $C = \{c_1, \ldots, c_i\}$ denote the set of i centers chosen by the algorithm at the end of iteration i. Let S denote the sample obtained using D^2-sampling in the $(i + 1)^{\text{st}}$ iteration. The key idea for an efficient $(1 + \varepsilon)$-approximation algorithm for k-means in the SSAC framework with a *perfect* oracle was the following. Given a sample S, we could compute using at most $k|S|$ same-cluster queries the partition $S_1, \ldots, S_k$ of S among the k optimal clusters such that $S_j = S \cap X_j$ for all j. In the following, we discuss how to achieve this partitioning of S among k optimal clusters when the oracle $\mathcal{O}^E$ is faulty.

We reduce the problem of finding the partitions of S among the optimal clusters to the problem of recovering dense (graph) clusters in a *stochastic block model* (SBM). An instance of an SBM is created as follows. Given any arbitrary partition $V_1, \ldots, V_k$ of a set V of vertices, an edge is added between two vertices belonging to the same partition with probability at least $(1 - q)$ and between two vertices in different partitions with probability at most q. We first construct an instance I of an SBM using the sample S. By querying the oracle $\mathcal{O}^E$ with all pairs of points u, v in S, we obtain a graph I on $|S|$ vertices, where vertices in I correspond to the points in S, and an edge exists in I between vertices u and v if $\mathcal{O}^E(u, v) = 1$. Since oracle $\mathcal{O}^E$ errs with probability at most q, for any $u, v \in S_j$ for some $j \in [k]$, there is an edge between u and v with probability at least $(1 - q)$. Similarly, there is an edge $(u, v) \in I$ for any two points $u \in S_y$ and $v \in S_z, y \neq z$ belonging to different optimal clusters with probability at most q. Note that the instance I created in this manner would be an instance of an SBM. Since $q < 1/2$, this procedure, with high probability, creates more edges between vertices belonging to the same partition than the number of edges between vertices in different partitions. Intuitively, the partitions of S would correspond to the dense (graph) clusters in the SBM instance I, and if there were no errors, then each partition would form a clique in I. One way to figure out the partitions $S_1, \ldots, S_k$ would be to retrieve the

Table 3 Approximation algorithm for k-means (top-left frame) using faulty oracle. Note that $\mu(T)$ denotes the centroid of T and D^2-sampling w.r.t. empty center set C means just uniform sampling. The algorithm `UniformSample`(X, C, s) (bottom-left) returns a uniform sample of size $\Omega(1/\varepsilon)$ (w.h.p.) from the optimal cluster in which point s belongs.

Constants: $N = \frac{(2^{13})k^3}{\varepsilon^2}$, $M = \frac{64k}{\varepsilon}$, $L = \frac{(2^{23})k^2}{\varepsilon^4}$

`Faulty-Query-`k`-means`(X, k, ε)

(1) $R \leftarrow \emptyset$

(2) $C \leftarrow \emptyset$

(3) for $i = 1$ to k

(3.1) D^2-sample a multi-set S of N points from X with respect to center set C

(3.2) $s \leftarrow$ `UncoveredCluster`(C, S, R)

(3.3) $T \leftarrow$ `UniformSample`(X, C, s)

(3.4) If $(|T| < M)$ continue

(3.5) $R \leftarrow R \cup \{s\}$

(3.6) $C \leftarrow C \cup \mu(T)$

(4) return(C)

`UncoveredCluster`(C, S, R)

- For all $i \in \{1, ..., k\}$: $S_i \leftarrow \emptyset$
- $i \leftarrow 1$
- For all $y \in R$: $\{S_i \leftarrow y$; i++$\}$
- $T_1, \ldots, T_l =$ `PartitionSample`(S)
- for $j = 1, \ldots, l$
  - if IsCovered(C, T_j) is FALSE
    - if $\exists t$ such that $S_t = \emptyset$, then $S_t = T_j$
- Let S_i be the largest set such that $i > |R|$
- Let $s \in S_i$ be the element with smallest value of $\Phi(C, \{s\})$ in S_i
- return(s)

`UniformSample`(X, C, s)

- $S \leftarrow \emptyset$
- For $i = 1$ to L:
  - D^2-sample point $x \in X$ w.r.t center set C
  - $U = U \cup \{x\}$
- $T_1, \ldots, T_l =$ `PartitionSample`(U)
- for $j = 1, \ldots, l$
  - If (`IsCovered`(s, T_j) is TRUE)
    - $\forall x \in T_j$, with probability $\left(\frac{\varepsilon}{128} \cdot \frac{\Phi(C,\{s\})}{\Phi(C,\{x\})}\right)$ add x in multi-set S
- return (S)

`PartitionSample`(S)

- Construct SBM instance I by querying $\mathcal{O}^E(s, t)$ $\forall s, t \in S$
- Run cluster recovery algorithm of Ailon et al. [4] on I
- Return $T_1, \ldots, T_l$ for $l < k$

`IsCovered(C,U)`

- for $c \in C$
  - if for majority of $u \in U$, $\mathcal{O}^E(c, u) = 1$
    - Return TRUE
- Return FALSE

dense (graph) clusters from the instance I. Ailon et al. [4] gave a randomized algorithm to retrieve all large clusters of any SBM instance. Their algorithm on a graph of n vertices retrieves all clusters of size at least $\sqrt{n}$ with high probability. Their main result in our context is given as follows.

▶ **Lemma 37** ([4]). *There exists a polynomial time algorithm that, given an instance of a stochastic block model on n vertices, retrieves all clusters of size t least $\Omega(\sqrt{n})$ with high probability, provided $q < 1/2$.*

We use Lemma 37 to retrieve the large clusters from our SBM instance I. We also need to make sure that the sample S is such that its overlap with at least one uncovered optimal

cluster is large, where an optimal cluster S_j for some j is *uncovered* if $C \cap S_j = \emptyset$. More formally, we would require the following: $\exists j \in [k]$ such that $|S_j| \geq \Omega(\sqrt{|S|})$, and X_j is uncovered by C. From Corollary 16, given a set of centers C with $|C| < k$, there exists an uncovered cluster such that any point sampled using D^2-sampling would belong to that uncovered cluster with probability at least $\frac{\varepsilon}{4k}$. Therefore, in expectation, D^2-sample S would contain at least $\frac{\varepsilon}{4k}|S|$ points from one such uncovered optimal cluster. In order to ensure that this quantity is at least as large as $\sqrt{|S|}$, we need $|S| = \Omega(\frac{16k^2}{\varepsilon^2})$. Our bounds for N and L, in the algorithm, for the size of D^2-sample S satisfy this requirement with high probability. This follows from a simple application of Chernoff bounds.

▶ **Lemma 38.** *For D^2-sample S of size at least $\frac{2^{12}k^2}{\varepsilon^2}$, there is at least one partition $S_j = S \cap X_j$ among the partitions returned by the sub-routine* `PartitionSample` *corresponding to an uncovered cluster X_j with probability at least $(1 - \frac{1}{16k})$.*

Proof. From Corollary 16, for any point p sampled using D^2-sampling, the probability that point p belongs to some uncovered cluster X_j is at least $\frac{\varepsilon}{4k}$. In expectation, the number of points sampled from uncovered cluster X_j is $\mathbf{E}[|S_j|] = \frac{\varepsilon|S|}{4k} = \frac{2^{10}k}{\varepsilon}$. Exact recovery using Lemma 37 requires $|S_j|$ to be at least $\frac{2^6 k}{\varepsilon}$. Using Chernoff bounds, the probability of this event is at least $(1 - \frac{1}{16k})$. ◀

Following Lemma 38, we condition on the event that there is at least one partition corresponding to an uncovered cluster among the partitions returned by the sub-routine `PartitionSample`. Next, we figure out using the sub-routine `IsCovered` which of the partitions returned by `PartitionSample` are covered and which are uncovered. Let $T_1, \ldots, T_l$ be the partitions returned by `PartitionSample` where $l < k$. Sub-routine `IsCovered` determines whether a cluster is covered or uncovered in the following manner. For each $j \in [l]$, we check whether T_j is covered by some $c \in C$. We query oracle $\mathcal{O}^E$ with pairs (v, c) for $v \in T_j$ and $c \in C$. If majority of the query answers for some $c \in C$ is 1, we say cluster T_j is covered by C. If for all $c \in C$ and some T_j, the majority of the query answers is 0, then we say T_j is uncovered by C. Using Chernoff bounds, we show that with high probability uncovered clusters would be detected.

▶ **Lemma 39.** *With probability at least $(1 - \frac{1}{16k})$, all covered and uncovered clusters are detected correctly by the sub-routine* `IsCovered`*.*

Proof. We find the probability that any partition T_j for $j \in [l]$ is detected correctly as covered or uncovered. Then we use union bound to bound the probability that all clusters are detected correctly. Recall that each partition returned by `PartitionSample` has size at least $|T_j| \geq \frac{2^6 k}{\varepsilon}$ for $j \in [l]$. We first compute for one such partition T_j and some center $c \in C$, the probability that majority of the queries $\mathcal{O}^E(v, c)$ where $v \in T_j$ are wrong. Since each query answer is wrong independently with probability $q < 1/2$, in expectation the number of wrong query answers would be $q|T_j|$. Using Chernoff bound, the probability that majority of the queries is wrong is at most $e^{-\frac{2^6 k}{3\varepsilon}(1-\frac{1}{2q})^2}$. The probability that the majority of the queries is wrong for at least one center $c \in C$ is at most $ke^{-\frac{2^6 k}{3\varepsilon}(1-\frac{1}{2q})^2}$. Again using union bound all clusters are detected correctly with probability at least $(1 - k^2 e^{-\frac{2^6 k}{3\varepsilon}(1-\frac{1}{2q})^2}) \geq (1 - \frac{1}{16k})$. ◀

With probability at least $(1 - \frac{1}{8k})$, given a D^2-sample S, we can figure out the largest uncovered optimal cluster using the sub-routines `PartitionSample` and `IsCovered`. The analysis of Algorithm 3 follows the analysis of Algorithm 2. For completeness, we compute the probability of success, and the query complexity of the algorithm. Note that s in line (3.2)

of the Algorithm 3 is chosen correctly with probability $(1-\frac{1}{4k})(1-\frac{1}{8k})$. The uniform sample in line (3.3) is chosen properly with probability $(1-\frac{1}{4k})(1-\frac{1}{8k})$. Since, given the uniform sample, success probability using Inaba's lemma is at least $(1-\frac{1}{4k})$, overall the probability of success becomes $(1-\frac{1}{k})$. For query complexity, we observe that `PartitionSample` makes $O(\frac{k^6}{\varepsilon^8})$ same-cluster queries to oracle $\mathcal{O}^E$, query complexity of `IsCovered` is $O(\frac{k^4}{\varepsilon^4})$. Since `PartitionSample` is called at most k times, total query complexity would be $O(\frac{k^7}{\varepsilon^8})$. Note that these are bounds for dataset that satisfies (k,ε)-irreducibility condition. For general dataset, we will use $O(\varepsilon/k)$ as the error parameter. This causes the number of same-cluster queries to be $O(k^{15}/\varepsilon^8)$. This proves the main result in Theorem 7.

6 Conclusion and Open Problems

This work explored the power of the SSAC framework defined by Ashtiani *et al.* [6] in the approximation algorithms domain. We showed how a simple modification of k-means++ seeding algorithm in SSAC framework gives a constant factor approximation for k-means. Furthermore, we obtained an efficient $(1+\varepsilon)$-approximation algorithm for the k-means problem within the SSAC framework. This is interesting because it is known that such an efficient algorithm is not possible in the classical model unless $\mathsf{P}=\mathsf{NP}$.

Our results encourage us to formulate similar query models for other hard problems. If the query model is reasonable (as is the SSAC framework for center-based clustering), then it may be worthwhile to explore its powers and limitations as it may be another way of circumventing the hardness of the problem. For instance, the problem closest to center-based clustering problems such as k-means is the *correlation clustering* problem. The query model for this problem may be similar to the SSAC framework. It will be interesting to see if same-cluster queries allow us to design efficient approximation algorithms for correlation clustering problem for which hardness results similar to that of k-means are known.

References

1 Ankit Aggarwal, Amit Deshpande, and Ravi Kannan. Adaptive sampling for k-means clustering. In *APPROX-RANDOM*, pages 15–28. Springer, 2009.

2 Sara Ahmadian, Ashkan Norouzi-Fard, Ola Svensson, and Justin Ward. Better guarantees for k-means and euclidean k-median by primal-dual algorithms. *CoRR*, abs/1612.07925, 2016. `arXiv:1612.07925`.

3 Nir Ailon, Anup Bhattacharya, Ragesh Jaiswal, and Amit Kumar. Approximate clustering with same-cluster queries. *CoRR*, abs/1704.01862, 2017. `arXiv:1704.01862`.

4 Nir Ailon, Yudong Chen, and Huan Xu. Iterative and active graph clustering using trace norm minimization without cluster size constraints. *Journal of Machine Learning Research*, 16:455–490, 2015.

5 David Arthur and Sergei Vassilvitskii. k-means++: The advantages of careful seeding. In *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 1027–1035. Society for Industrial and Applied Mathematics, 2007.

6 Hassan Ashtiani, Shrinu Kushagra, and Shai Ben-David. Clustering with same-cluster queries. In *Advances in neural information processing systems*, pages 3216–3224, 2016.

7 Pranjal Awasthi, Maria-Florina Balcan, and Konstantin Voevodski. Local algorithms for interactive clustering. In *ICML*, pages 550–558, 2014.

8 Pranjal Awasthi, Avrim Blum, and Or Sheffet. Center-based clustering under perturbation stability. *Information Processing Letters*, 112(1):49–54, 2012.

9 Pranjal Awasthi, Moses Charikar, Ravishankar Krishnaswamy, and Ali Kemal Sinop. The Hardness of Approximation of Euclidean k-Means. In *31st International Symposium on Computational Geometry (SoCG 2015)*, volume 34, pages 754–767, 2015.

10 Maria-Florina Balcan and Avrim Blum. Clustering with interactive feedback. In *ALT*, pages 316–328. Springer, 2008.

11 Maria-Florina Balcan, Avrim Blum, and Anupam Gupta. Approximate clustering without the approximation. In *Proceedings of the twentieth annual ACM-SIAM symposium on Discrete algorithms*, pages 1068–1077, 2009.

12 Sugato Basu, Arindam Banerjee, and Raymond J Mooney. Active semi-supervision for pairwise constrained clustering. In *Proceedings of the 2004 SIAM international conference on data mining*, pages 333–344. SIAM, 2004.

13 Anup Bhattacharya, Ragesh Jaiswal, and Nir Ailon. Tight lower bound instances for k-means++ in two dimensions. *Theoretical Computer Science*, 634:55–66, 2016.

14 Tobias Brunsch and Heiko Röglin. A bad instance for k-means++. *Theoretical Computer Science*, 505:19–26, 2013.

15 Vincent Cohen-Addad, Philip N. Klein, and Claire Mathieu. Local search yields approximation schemes for k-means and k-median in euclidean and minor-free metrics. 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 353–364. IEEE Computer Society, 2016. doi:10.1109/FOCS.2016.46.

16 Sanjoy Dasgupta. The hardness of k-means clustering. Technical report, Department of Computer Science and Engineering, University of California, San Diego, 2008.

17 Irit Dinur. The pcp theorem by gap amplification. *Journal of the ACM (JACM)*, 54(3):12, 2007.

18 Dan Feldman, Morteza Monemizadeh, and Christian Sohler. A ptas for k-means clustering based on weak coresets. In *Proceedings of the twenty-third annual symposium on Computational geometry*, pages 11–18. ACM, 2007.

19 Zachary Friggstad, Mohsen Rezapour, and Mohammad R Salavatipour. Local search yields a ptas for k-means in doubling metrics. In *Foundations of Computer Science (FOCS), 2016 IEEE 57th Annual Symposium on*, pages 365–374. IEEE, 2016.

20 Russell Impagliazzo and Ramamohan Paturi. On the complexity of k-sat. *Journal of Computer and System Sciences*, 62:367–375, 2001.

21 Russell Impagliazzo, Ramamohan Paturi, and Francis Zane. Which problems have strongly exponential complexity? *Journal of Computer and System Sciences*, 63:512–530, 2001.

22 Mary Inaba, Naoki Katoh, and Hiroshi Imai. Applications of weighted voronoi diagrams and randomization to variance-based k-clustering. In *Proceedings of the tenth annual symposium on Computational geometry*, pages 332–339. ACM, 1994.

23 Ragesh Jaiswal, Amit Kumar, and Sandeep Sen. A simple d2-sampling based ptas for k-means and other clustering problems. *Algorithmica*, 70(1):22–46, 2014.

24 Ragesh Jaiswal, Mehul Kumar, and Pulkit Yadav. Improved analysis of d^2-sampling based ptas for k-means and other clustering problems. *Information Processing Letters*, 115(2):100–103, 2015.

25 Tapas Kanungo, David M Mount, Nathan S Netanyahu, Christine D Piatko, Ruth Silverman, and Angela Y Wu. A local search approximation algorithm for k-means clustering. In *Proceedings of the eighteenth annual symposium on Computational geometry*, pages 10–18. ACM, 2002.

26 Amit Kumar, Yogish Sabharwal, and Sandeep Sen. Linear-time approximation schemes for clustering problems in any dimensions. *Journal of the ACM (JACM)*, 57(2):5, 2010.

27 Euiwoong Lee, Melanie Schmidt, and John Wright. Improved and simplified inapproximability for k-means. *Information Processing Letters*, 120:40–43, 2017.

28 Meena Mahajan, Prajakta Nimbhorkar, and Kasturi Varadarajan. The planar k-means problem is np-hard. *Theoretical Computer Science*, 442:13–21, 2012.

29 Pasin Manurangsi. Almost-polynomial ratio eth-hardness of approximating densest k-subgraph. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*, pages 954–961. ACM, 2017.

30 Rafail Ostrovsky, Yuval Rabani, Leonard J Schulman, and Chaitanya Swamy. The effectiveness of lloyd-type methods for the k-means problem. *Journal of the ACM (JACM)*, 59(6):28, 2012.

31 Luca Trevisan. Inapproximability of combinatorial optimization problems. *CoRR*, cs.CC/0409043, 2004. URL: `http://arxiv.org/abs/cs.CC/0409043`.

32 Andrea Vattani. The hardness of k-means clustering in the plane. Technical report, Department of Computer Science and Engineering, University of California San Diego, 2009.

33 Sharad Vikram and Sanjoy Dasgupta. Interactive bayesian hierarchical clustering. In *International Conference on Machine Learning*, pages 2081–2090, 2016.

34 Konstantin Voevodski, Maria-Florina Balcan, Heiko Röglin, Shang-Hua Teng, and Yu Xia. Efficient clustering with limited distance information. *CoRR*, abs/1408.2045, 2014. `arXiv:1408.2045`.

Graph Clustering using Effective Resistance*

Vedat Levi Alev[†1], Nima Anari[2], Lap Chi Lau[‡3], and Shayan Oveis Gharan[§4]

1 **University of Waterloo, Waterloo, Canada**
vlalev@uwaterloo.ca

2 **Simons Institute, Berkeley, USA**
anari@stanford.edu

3 **University of Waterloo, Waterloo, Canada**
lapchi@uwaterloo.ca

4 **University of Washington, Seattle, USA**
shayan@cs.uwaterloo.ca

Abstract

We design a polynomial time algorithm that for any weighted undirected graph $G = (V, E, \boldsymbol{w})$ and sufficiently large $\delta > 1$, partitions V into subsets $V_1, \ldots, V_h$ for some $h \geq 1$, such that

- at most δ^{-1} fraction of the weights are between clusters, i. e.

$$w(E - \cup_{i=1}^{h} E(V_i)) \lesssim \frac{w(E)}{\delta};$$

- the effective resistance diameter of each of the induced subgraphs $G[V_i]$ is at most δ^3 times the inverse of the average weighted degree, i. e.

$$\max_{u,v \in V_i} \mathsf{Reff}_{G[V_i]}(u, v) \lesssim \delta^3 \cdot \frac{|V|}{w(E)} \quad \text{for all } i = 1, \ldots, h.$$

In particular, it is possible to remove one percent of weight of edges of any given graph such that each of the resulting connected components has effective resistance diameter at most the inverse of the average weighted degree.

Our proof is based on a new connection between effective resistance and low conductance sets. We show that if the effective resistance between two vertices u and v is large, then there must be a low conductance cut separating u from v. This implies that very mildly expanding graphs have constant effective resistance diameter. We believe that this connection could be of independent interest in algorithm design.

1998 ACM Subject Classification G.2.2 Graph Theory

Keywords and phrases Electrical Flows, Effective Resistance, Conductance, Graph Partitioning

Digital Object Identifier 10.4230/LIPIcs.ITCS.2018.41

* A full version a the paper is available at [3], https://arxiv.org/abs/1711.06530.
† Supported by the David R. Cheriton Graduate Scholarship.
‡ Supported by NSERC Discovery Grant 2950-120715 and NSERC Accelerator Supplement 2950-120719.
§ Supported by NSF Career Award and ONR Young Investigator Award

1 Introduction

Graph decomposition is a useful algorithmic primitive with various applications. The general framework is to remove few edges so that the remaining components have nice properties, and then specific problems are solved independently in each component. Several types of graph decomposition results have been studied in the literature. The most relevant to this work are low diameter graph decompositions and expander decompositions. We refer the reader to Section 2 for notation and definitions.

Low Diameter Graph Decompositions: Given a weighted undirected graph $G = (V, E, \boldsymbol{w})$ and a parameter $\Delta > 0$, a low diameter graph decomposition algorithm seeks to partition the vertex set V into sets $V_1, \ldots, V_h$ with the following two properties:

- Each component $G[V_i]$ has bounded shortest path diameter, i. e. $\max_{u,v \in V_i} \text{dist}_w(u, v) \leq \Delta$, where $\text{dist}_w(u, v)$ is the shortest path distance between u and v using the edge weight $\boldsymbol{w}$.
- There are not too many edges between the sets V_i, i. e. $\left|E - \bigcup_{i=1}^{h} E(V_i)\right| \leq \frac{D(G)}{\Delta} \cdot |E|$, where $D(G)$ is the "distortion" that depends on the input graph.

This widely studied [39, 31, 11, 38, 1] primitive (and its generalization to decomposition into padded partitions) has been very useful in designing approximation algorithms [17, 15, 21, 22, 35, 9, 37]. This approach is particularly effective when the input graph is of bounded genus g or K_r-minor free, in which case $D(G) = \mathcal{O}(\log g)$ [38] and $D(G) = \mathcal{O}(r)$ [1]. For these special graphs, this primitive can be used to proving constant flow-cut gaps [31], proving tight bounds on the Laplacian spectrum [12, 26], and obtaining constant factor approximation algorithms for NP-hard problems [9, 4]. However, there are graphs for which $D(G)$ is necessarily $\Omega(\log n)$ where n is the number of vertices, and this translates to a $\Omega(\log n)$ factor loss in applying this approach to general graphs. For example, in a hypercube, if we only delete a small constant fraction of edges, some remaining components will have diameter $\Omega(\log n)$.

Expander Decompositions: Given an undirected graph $G = (V, E)$ and a parameter $\phi > 0$, an expander decomposition algorithm seeks to partition the vertex set V into sets $V_1, \ldots, V_h$ with the following two properties.

- Each component $G[V_i]$ is a ϕ-expander, i. e. $\Phi(G[V_i]) \geq \phi$, where $\Phi(G[V_i])$ is the conductance of the induced subgraph $G[V_i]$; see Section 2 for the definition of conductance.
- There are not too many edges between the sets V_i, i. e. $\left|E - \bigcup_{i=1}^{h} E(V_i)\right| \leq \delta(G, \phi) \cdot |E|$, where $\delta(G, \phi)$ is a parameter depending on the graph G and ϕ.

This decomposition is also well studied [25, 44, 7, 42], and is proved useful in solving Laplacian equations, approximating Unique Games, and designing clustering algorithms. It is of natural interest to minimize the parameter $\delta(G, \phi)$. Similar to the low diameter partitioning case, there are graphs where $\delta(G, \phi) \geq \Omega(\phi \cdot \log(n))$. For example, in a hypercube, if we delete a small constant fraction of edges, some remaining components will have conductance $\mathcal{O}(1/\log n)$.

Motivations: In some applications, we could not afford to have an $\Omega(\log n)$ factor loss in the approximation ratio. One motivating example is the Unique Games problem. It is known that Unique Games can be solved effectively in graphs with constant conductance [8] and more generally in graphs with low threshold rank [32, 24, 10], and in graphs with constant diameter [23]. Some algorithms for Unique Games on general graphs are based on graph

decomposition results that remove a small constant fraction of edges so that the remaining components are of low threshold rank [7] or of low diameter [4], but the $\Omega(\log n)$ factor loss in the decomposition is the bottleneck of these algorithms. This leads us to the question of finding a property that is closely related to low diameter and high expansion, so that every graph admits a decomposition into components with such a property without an $\Omega(\log n)$ factor loss.

Effective Resistance Diameter: The property that we consider in this paper is having low effective resistance diameter. We interpret the graph $G = (V, E, \boldsymbol{w})$ as an electrical circuit by viewing every edge $e \in E$ as a resistor with resistance $1/w(e)$. The effective resistance distance $\mathsf{Reff}(u, v)$ between the vertices u and v is then the potential difference between u and v when injecting a unit of electric flow into the circuit from the vertex u and removing it out of the circuit from the vertex v. We define

$$\mathcal{R}_{diam}(G) := \max_{u,v \in V} \mathsf{Reff}(u, v)$$

as the effective resistance diameter of G. Both the properties of low diameter and of high expansion have the property of low effective resistance diameter as a common denominator: The effective resistance distance $\mathsf{Reff}(u, v)$ is upper bounded by the shortest path distance for any graph, and so every low diameter component has low effective resistance diameter. Also, a d-regular graph with constant expansion has effective resistance diameter $\mathcal{O}(1/d)$ [14, 16], and so an expander graph also has low effective resistance diameter. See Section 2 for more details.

In this paper, we study the connection between effective resistance and graph conductance. Roughly speaking, we show if all sets have mild expansion (see Theorem 1), then the effective resistance diameter is small. We use this observation to design a graph partitioning algorithm to decompose a graph into clusters with effective resistance diameter at most the inverse of the average degree (up to constant losses) while removing only a constant fraction of edges. This shows that although we cannot partition a graph into $\Omega(1)$-expanders by removing a constant fraction of edges, we can partition it into components that satisfy the "electrical properties" of expanders.

Applications of Effective Resistance: Besides the motivation from the Unique Games problem, we believe that effective resistance is a natural property to be investigated on its own. The effective resistance distance between two vertices $u, v \in V$ has many useful probabilistic interpretations, such as the commute time [16], the cover time [41], and the probability of an edge being in a random spanning tree [29]. See Section 2 for more details. Recently, the concept of effective resistance has found surprising applications in spectral sparsification [43], in computing maximum flows [19], in finding thin trees [6], and in generating random spanning trees [27, 40, 20]. The recent algorithms in generating a random spanning tree are closely related to our work. Madry and Kelner [27] showed how to sample a random spanning tree in time $\widetilde{\mathcal{O}}(m \cdot \sqrt{n})$ where m is the number of edges, faster than the worst case cover time $\widetilde{\mathcal{O}}(m \cdot n)$ (see Section 2). A cruicial ingredient of their algorithm is the low diameter graph decomposition technique, which they use to ensure that the resulting components have small cover time. In subsequent work, Madry, Straszak and Tarnawski [40] have improved the time complexity of their algorithm to $\widetilde{\mathcal{O}}(m^{4/3})$ by working with the effective resistance metric instead of the shortest path metric. Indeed, their technique of reducing the effective resistance diameter is similar to our technique – even though it cannot recover our result.

1.1 Our Results

Our main technical result is the following connection between effective resistance and graph partitioning.

▶ **Theorem 1.** *Let $G = (V, E)$ be a weighted graph with weights $\boldsymbol{w} \in \mathbb{R}^E_{\geq 0}$. Suppose for any set $S \subseteq V$ with $\mathrm{vol}(S) \leq \mathrm{vol}(G)/2$ we have*

$$\Phi(S) \geq \frac{c}{\mathrm{vol}(S)^{1/2-\varepsilon}} \qquad \text{(mild expansion)}$$

for some $c > 0$ and $1/2 \geq \varepsilon \geq 0$. Then, for any pair of vertices $s, t \in V$, we have

$$\mathsf{Reff}(s,t) \lesssim \left(\frac{1}{\deg(s)^{2\varepsilon}} + \frac{1}{\deg(t)^{2\varepsilon}}\right) \cdot \frac{1}{\varepsilon \cdot c^2}, \qquad \text{(resistance bound)}$$

where $\deg(v) = \sum_{u:uv\in E} w(u,v)$ is the weighted degree of v.

In [16], Chandra et al. proved that a d-regular graph with constant expansion has effective resistance diameter $\mathcal{O}(1/d)$. They also proved that the effective resistance diameter of a d-dimensional grid is $\mathcal{O}(1/d)$ when $d > 2$ even though it is a poor expander. Theorem 1 can be seen as a common generalization of these two results, using the mild expansion condition as a unifying assumption. Chandra et al. [16] also showed that the effective resistance diameter of a 2-dimensional grid is $\Theta(\log n)$. Note that for a $\sqrt{n} \times \sqrt{n}$ grid, $\Phi(S) \approx 1/\mathrm{vol}(S)^{1/2}$ for any $k \times k$ square. This shows that the mild expansion assumption of the theorem cannot be weakened in the sense that if $\varepsilon = 0$ for some sets S, then $\mathsf{Reff}(s,t)$ may grow as a function of $|V|$.

The proof of Theorem 1 also provides an efficient algorithm to find such a sparse cut. The high-level idea is to prove that if all level sets of the st electric potential vector satisfy the mild expansion condition, then the potential difference between s and t must be small, i. e., $\mathsf{Reff}(s,t)$ is small. Combining with a fast Laplacian solver [45], we show that the existence of a pair of vertices $u, v \in V$ with high effective resistance distance implies the existence of a sparse cut which can be found in nearly linear time.

▶ **Corollary 2.** *Let $G = (V, E, \boldsymbol{w})$ be a weighted undirected graph. If $\deg(v) \geq 1/\alpha$ for all $v \in V$, then for any $0 < \varepsilon < 1/2$, there is a subset of vertices $U \subseteq V$ such that*

$$\Phi(U) \lesssim \frac{\alpha^\varepsilon}{\sqrt{\mathcal{R}_{diam} \cdot \varepsilon}} \cdot \mathrm{vol}(U)^{\varepsilon - 1/2}.$$

Furthermore, the set U can be found in time $\widetilde{\mathcal{O}}\Big(m \cdot \log\Big(\frac{w(E)}{\min_e w(e)}\Big)\Big)$.

Using Corollary 2 repeatedly, we can prove the following graph decomposition result.

▶ **Theorem 3** (Main)**.** *Given a weighted undirected graph $G = (V, E, \boldsymbol{w})$, and a large enough parameter $\delta > 1$, there is an algorithm with time complexity $\widetilde{\mathcal{O}}\Big(m \cdot n \cdot \log\Big(\frac{w(E)}{\min_e w(e)}\Big)\Big)$ that finds a partition $V = \bigcup_{i=1}^h V_i$ satisfying*

$$w\left(E - \bigcup_{i=1}^{h} E(V_i)\right) \lesssim \frac{w(E)}{\delta} \qquad \text{(loss bound)}$$

and

$$\mathcal{R}_{diam}(G[V_i]) \lesssim \delta^3 \cdot \frac{n}{w(E)} \qquad \text{(resistance bound)}$$

for all $i = 1, \ldots, h$.

Let G be a d-regular unweighted graph. Theorem 3 implies that it is possible to remove a constant fraction of the edges of G and decompose G into components with effective resistance diameter at most $\mathcal{O}(1/d)$. Note that d-regular $\Omega(1)$-expanders with $\mathcal{R}_{diam} = \mathcal{O}(1/d)$ have the least effective resistance diameter among all d-regular graphs. So, even though it is impossible to decompose d-regular graphs into graphs with $\Omega(1)$-expansion while removing only a constant fraction of edges, we can find a decomposition with analogous "electrical properties".

We can also view Theorem 3 as a generalization of the following result: Any d-regular graph can be decomposed into $\Omega(d)$-edge connected subgraphs by removing only a constant fraction of edges. This is because if the effective resistance diameter of an unweighted graph G is ϵ, then G must be $1/\epsilon$-edge connected. Recall that a graph is k-edge connected, if the size of every cut in that graph is at least k.

2 Preliminaries

In this section, we will first define the notations used in this paper, and then we will review the background in effective resistances, Laplacian solvers, and graph expansions in the following subsections.

Given an undirected graph $G = (V, E)$ and a subset of vertices $U \subseteq V$, we use the notation $E_G(U)$ for the set of edges with both endpoints in U, i.e. $E_G(U) = \{\{u, v\} \in E(G) : u, v \in U\}$. We write U^c for the complement of U with respect to $V(G)$, i. e. $U^c = V \backslash U$. The variables n and m stand for the number of vertices and the edges of the graph respectively, i. e. $n = |V|$ and $m = |E|$. We use the notation $\partial_G U$ for the edge boundary of $U \subseteq V$, i.e. $\partial_G U = E_G(U, U^c) = \{\{u, v\} \in E : u \in U, v \in U^c\}$. For a graph $G = (V, E)$ with weights $\boldsymbol{w} \in \mathbb{R}^E_{\geq 0}$, we write $\deg_G(v) = \sum_{u:uv\in E} w(u, v)$ for the weighted degree of v. For $S \subseteq V$, the volume $\mathrm{vol}_G(S)$ of S is defined to be $\mathrm{vol}_G(S) = \sum_{s\in S} \deg(s)$. When the graph is clear in the context we may drop the subscript in all aforementioned notation.

Scalar functions and vectors are typed in bold, i.e. $\boldsymbol{x} \in \mathbb{R}^V$, or $\boldsymbol{w} \in \mathbb{R}^E$. For a subset $A \subseteq E$, the notation $w(A)$ stands for the sum of the weights of all edges in A, i. e. $w(A) = \sum_{e\in A} w(e)$. The j-th canonical basis vector is denoted by $\boldsymbol{e}_j \in \mathbb{R}^V$. Matrices are typed in serif, i. e. $\mathsf{A} \in \mathbb{R}^{V\times V}$.

Time complexities are given in asymptotic notation. We employ the notation $\widetilde{\mathcal{O}}(f(n))$ to hide polylogarithmic factors in n, i.e. $\widetilde{\mathcal{O}}(f(n)) = \mathcal{O}(f(n) \cdot \mathrm{polylog}(n))$. We use the notation $f \lesssim g$ for asymptotic inequalities, i. e. $f = \mathcal{O}(g)$; and the notation $f \asymp g$ for asymptotic equalities, i. e. $f = \Theta(g)$.

2.1 Electric Flow, Electric Potential, and Effective Resistance

Let $G = (V, E)$ be a given graph with non-negative edge weights $\boldsymbol{w} \in \mathbb{R}^E_{\geq 0}$. The notion of an electric flow arises when one interprets the graph G as an electrical network where every edge $e \in E$ represents a resistor with resistance $1/w(e)$.

We fix an arbitrary orientation $E^{\pm}$ of the edges E and define a unit st flow in this network as a function $\boldsymbol{f} \in \mathbb{R}^{E^{\pm}}_{\geq 0}$ (where for $e \notin E^{\pm}$ we define $f(e) = -f(-e)$) satisfying the following:

$$\sum_{w\in\delta^+(v)} f(vw) - \sum_{u\in\delta^-(v)} f(uv) = \begin{cases} 1 & \text{if } v = s \\ -1 & \text{if } v = t \\ 0 & \text{otherwise,} \end{cases} \qquad \text{(flow conservation)}$$

where $\delta^+(v)$ is the set of edges having v as the head in our orientation, and $\delta^-(v)$ is the set of edges having v as tail. Let $e = uv \in E^{\pm}$ be an oriented edge. The flow $\boldsymbol{f}$ has to obey Ohm's law

$$f(e) = w(e) \cdot \Delta_e \boldsymbol{p} = w(e) \cdot (p(u) - p(v)) \qquad \text{(Ohm's law)}$$

for some vector $\boldsymbol{p} \in \mathbb{R}^V$ which we call the potential vector. The electrical flow between the vertices s and t is the unit st flow that satisfies flow conservation and Ohm's law.

The electrical energy $\mathcal{E}(\boldsymbol{f})$ of a flow $\boldsymbol{f}$ is defined as the following quantity,

$$\mathcal{E}(\boldsymbol{f}) = \sum_{e \in E^{\pm}} \frac{f(e)^2}{w(e)}. \qquad \text{(electrical energy)}$$

It is known that the electric flow between s and t is the unit st flow with minimal electrical energy. The effective resistance $\mathsf{Reff}(s,t)$ between the vertices s and t is the potential difference between the vertices s and t induced by this flow, i.e. $\mathsf{Reff}(s,t) = p(s) - p(t)$. It is known that the potential difference between s and t equals the energy $\mathcal{E}(\boldsymbol{f}_{st})$ of this flow. This is often referred as Thomson's principle.

The electric potential vector and the effective resistance are known to have the following closed form expressions: Let $\mathsf{W} \in \mathbb{R}^{V \times V}$ be the weighted adjacency matrix of G, i. e. the matrix satisfying $W(u,v) = 1[uv \in E] \cdot w(u,v)$, and $\mathsf{D} \in \mathbb{R}^{V \times V}$ the weighted degree matrix, i. e. the diagonal matrix satisfying $D(v,v) = \deg(v) = \sum_{u:uv \in E} w(u,v)$. The (weighted) Laplacian $\mathsf{L}_G \in \mathbb{R}^{V \times V}$ is defined to be the matrix

$$\mathsf{L}_G = \mathsf{D} - \mathsf{W}. \qquad \text{(weighted Laplacian)}$$

It is well-known that this is a symmetric positive semi-definite matrix. We will take

$$\mathsf{L}_G = \sum_{i=2}^{n} \lambda_i \boldsymbol{v}_i \boldsymbol{v}_i^\top$$

as the spectral decomposition of L_G, where $\lambda_1 = 0 \leq \lambda_2 \leq \cdots \leq \lambda_n$ are the eigenvalues of L_G sorted in increasing order. It is easy to verify $\mathsf{L}_G \mathbf{1} = 0$ and further it can be shown that this is the only vector (up to scaling) satisfying this when G is connected. This means if G is connected, the matrix L_G is invertible in the subspace perpendicular to $\mathbf{1}$. This inversion will be done by the matrix $\mathsf{L}_G^\dagger$, the so-called Moore-Penrose pseudo-inverse of L_G defined by

$$\mathsf{L}_G^\dagger = \sum_{j=2}^{n} \frac{1}{\lambda_j} \boldsymbol{v}_i \boldsymbol{v}_i^\top. \qquad \text{(pseudo-inverse of } \mathsf{L}_G\text{)}$$

Let $\boldsymbol{f}^\star \in \mathbb{R}^E$ be the st unit electric flow vector. It can be verified that the st electric potential $\boldsymbol{p}^\star$ – i.e. the vector satisfying $w(uv) \cdot (p^\star(u) - p^\star(v)) = f^\star(uv)$ for all $uv \in E^{\pm}$ – satisfies the equation

$$\mathsf{L}_G \boldsymbol{p}^\star = \boldsymbol{e}_s - \boldsymbol{e}_t \iff \boldsymbol{p}^\star = \mathsf{L}_G^\dagger (\boldsymbol{e}_s - \boldsymbol{e}_t). \tag{2.1}$$

In particular, this implies the following closed form expression for $\mathsf{Reff}(s,t)$

$$\mathsf{Reff}(s,t) = \langle \boldsymbol{e}_s - \boldsymbol{e}_t, \mathsf{L}_G^\dagger (\boldsymbol{e}_s - \boldsymbol{e}_t) \rangle. \qquad (st \text{ effective resistance})$$

It can be verified that this defines a (ℓ_2^2) metric on the set vertices V of G [30], as we have

1. $\mathsf{Reff}(u,v) = 0$ if and only if $u = v$.

2. $\mathsf{Reff}(u,v) = \mathsf{Reff}(v,u)$ for all $u,v \in V$.
3. $\mathsf{Reff}(u,v) + \mathsf{Reff}(v,w) \geq \mathsf{Reff}(u,w)$ for all $u,v,w \in V$.

Further, by routing the unit st flow along the st shortest path we see that the shortest path metric dominates the effective resistance metric, i. e. $\mathsf{Reff}(u,v) \leq \mathrm{dist}(u,v)$ for all the pairs of vertices $u,v \in V$.

It is known that the commute time distance $\kappa(u,v)$ between u and v – the expected number of steps a random walk starting from the vertex u needs to visit the vertex v and then return to u – is $\mathrm{vol}(G)$ times the effective resistance distance $\mathsf{Reff}(u,v)$ [16]. Also, the effective resistance $\mathsf{Reff}(u,v)$ of an edge $uv \in E$ corresponds to the probability of this edge being contained in a uniformly sampled random spanning tree [29]. A well-known result of Matthews [41] relates the effective resistance diameter to the cover time of the graph – the expected number of steps a random walk needs to visit all the vertices of G. Aldous [2] and Broder [13] have shown that simulating a random walk until every vertex has been visited allows one to sample a uniformly random spanning tree of the graph.

2.2 Solving Laplacian Systems

For our algorithmic results, it will be important to be able to compute electric potentials, and effective resistances quickly. We will do this by appealing to Equation (2.1) and the definition of the st effective resistance. Both of these equations require us to solve a Laplacian system. Fortunately, it is known that these systems can be solved in nearly linear time [45, 33, 34, 28, 36].

▶ **Lemma 4** (The Spielman-Teng Solver, [45]). *Let a (weighted) Laplacian matrix $\mathsf{L} \in \mathbb{R}^{V\times V}$, a right-hand side vector $\boldsymbol{b} \in \mathbb{R}^V$, and an accuracy parameter $\zeta > 0$ be given. Then, there is a randomized algorithm which takes time $\widetilde{\mathcal{O}}(m \cdot \log(1/\zeta))$ and produces a vector $\widehat{\boldsymbol{x}}$ that satisfies*

$$\left\|\widehat{\boldsymbol{x}} - \mathsf{L}^{\dagger}\boldsymbol{b}\right\|_{\mathsf{L}} \leq \zeta \cdot \|\mathsf{L}^{\dagger}\boldsymbol{b}\|_{\mathsf{L}} \qquad \text{(accuracy guarantee)}$$

with constant probability, where $\|\boldsymbol{x}\|_{\mathsf{A}}^2 = \langle \boldsymbol{x}, \mathsf{A}\boldsymbol{x}\rangle$.

For our purposes it will suffice to pick ζ inversely polynomial in the size of the graph in the unweighted case, and $1/\operatorname{poly}(w(E)/\min_e w(e), 1/m)$ in the weighted case.

Extending the ideas of Kyng and Sachdeva [36], Durfee et al. [20] show that it is possible to compute approximations for effective resistances between a set of given pairs $S \subseteq V \times V$ efficiently.

▶ **Lemma 5.** *Let $G = (V,E,\boldsymbol{w})$ be a weighted graph, $\beta > 0$ an accuracy parameter, and $S \subseteq V \times V$. There is an $\widetilde{\mathcal{O}}(m + (n + |S|)/\beta^2)$-time algorithm which returns numbers $A(u,v)$ for all $(u,v) \in V$ satisfying*

$$e^{-\beta}\,\mathsf{Reff}(u,v) \leq A(u,v) \leq e^{\beta}\,\mathsf{Reff}(u,v).$$

This lemma will aid us in computing fast approximations for furthest points in the effective resistance metric. For our purposes, we only need to pick β as a small enough constant, i. e. $\beta = \ln(3/2)$. Similar guarantees can also be obtained using the ideas of Spielman and Srivastava [43].

2.3 Conductance

For a graph $G = (V,E)$ with non-negative edge weights $\boldsymbol{w} \in \mathbb{R}^E_{\geq 0}$, we define the conductance of a set $S \subseteq V$ as

$$\Phi(S) = \frac{w(\partial S)}{\mathrm{vol}(S)}. \qquad \text{(conductance of a set)}$$

The conductance of the graph G is then defined as

$$\Phi(G) = \min\{\Phi(S) : S \subseteq V \text{ and } 2\operatorname{vol}(S) \leq \operatorname{vol}(G)\}. \qquad \text{(conductance of a graph)}$$

It is well-known [18, 5] that the conductance of the graph G is controlled by the spectral gap (second smallest eigenvalue) $\tilde{\lambda}_2$ of the normalised Laplacian matrix $\mathsf{D}^{-1/2}\mathsf{L}_G\mathsf{D}^{-1/2}$, i. e.

$$\tilde{\lambda}_2 \lesssim \Phi(G) \lesssim \sqrt{\tilde{\lambda}_2}. \qquad \text{(Cheeger's inequality)}$$

Appealing to the closed form formula for the *st* effective resistance it can be verified that the spectral gap λ_2 of the (unnormalised) Laplacian controls the effective resistance distance, i. e.

$$\max_{s,t\in V} \mathsf{Reff}(s,t) \lesssim \frac{1}{\lambda_2}.$$

By an easy application of Cheeger's inequality we see that the expansion controls the effective resistance as well, i. e.

$$\max_{s,t\in V} \mathsf{Reff}(s,t) \lesssim \frac{1}{\Phi(G)^2}.$$

Indeed, Theorem 1 and Corollary 2 will improve upon this bound.

3 From Well Separated Points to Sparse Cuts

In this section, we are going to prove Theorem 1 and Corollary 2. As previously mentioned, we will prove that if all the level sets of the potential vector have mild expansion, the effective resistance cannot be high.

▶ **Theorem 1.** *Let $G = (V, E)$ be a weighted graph with weights $\boldsymbol{w} \in \mathbb{R}^E_{\geq 0}$. Suppose for any set $S \subseteq V$ with $\operatorname{vol}(S) \leq \operatorname{vol}(G)/2$ we have*

$$\Phi(S) \geq \frac{c}{\operatorname{vol}(S)^{1/2-\varepsilon}} \qquad \text{(mild expansion)}$$

for some $c > 0$ and $1/2 \geq \varepsilon \geq 0$. Then, for any pair of vertices $s, t \in V$, we have

$$\mathsf{Reff}(s,t) \lesssim \left(\frac{1}{\deg(s)^{2\varepsilon}} + \frac{1}{\deg(t)^{2\varepsilon}}\right) \cdot \frac{1}{\varepsilon \cdot c^2}, \qquad \text{(resistance bound)}$$

where $\deg(v) = \sum_{u:uv\in E} w(u,v)$ is the weighted degree of v.

Proof. In the following let $\boldsymbol{f} \in \mathbb{R}^E$ be a unit electric flow from s to t, and $\boldsymbol{p} \in \mathbb{R}^V$ be the corresponding vector of potentials where we assume without loss of generality that $p(t) = 0$. We direct our attention to the following threshold sets

$$S_p = \{v \in V : \boldsymbol{p}(v) \geq p\}.$$

Then, we have

$$\sum_{e\in\partial S_p} |f(e)| = 1.$$

Using Ohm's law, we can rewrite this into

$$\sum_{e\in\partial S_p} w(e) \cdot |\Delta_e \boldsymbol{p}| = 1, \qquad (3.1)$$

where $\Delta_e \boldsymbol{p}$ is the potential difference along the endpoints of the edge e. Normalizing this, we get

$$\sum_{e \in \partial S_p} \frac{w(e)}{w(\partial S_p)} \cdot |\Delta_e \boldsymbol{p}| = \frac{1}{w(\partial S_p)}. \tag{3.2}$$

Now, set $\mu(e) = w(e)/w(\partial S_p)$. Restricted over the set of edges ∂S_p, μ is a probability distribution and the LHS of (3.2) corresponds to the expected potential drop when edges $e \in \partial S_p$ are sampled with respect to the probability distribution μ, i. e. we have

$$\mathbb{E}_\mu |\Delta_e \boldsymbol{p}| = \frac{1}{w(\partial S_p)}.$$

Then, by Markov's inequality, we get a set $F \subseteq \partial S_p$ such that

- all edges $f \in F$ satisfy

$$|\Delta_f \boldsymbol{p}| \leq \frac{2}{w(\partial S_p)};$$

- $\mathbb{P}_\mu(e \in F) \geq 1/2$, equivalently

$$w(F) = \sum_{e \in F} w(e) = \sum_{e \in F} w(\partial S_p) \cdot \mu(e) = w(\partial S_p) \cdot \mu(F) \geq \frac{w(\partial S_p)}{2}.$$

Using the observation that the endpoint of an edge $f \in F$ that is not contained in S_p should have potential at least $p - 2/w(\partial S_p)$, we obtain

$$\operatorname{vol}(S_{p-2/w(\partial S_p)}) \geq \operatorname{vol}(S_p) + w(F) \geq \operatorname{vol}(S_p) + \frac{w(\partial S_p)}{2}.$$

Assuming $\operatorname{vol}(\partial S_p) \leq \operatorname{vol}(G)/2$, using the mild expansion property, we have $w(\partial S_p) \geq c \operatorname{vol}(S_p)^{1/2+\varepsilon}$. So, from above we get

$$\operatorname{vol}(S_{p-2/c \operatorname{vol}(S_p)^{1/2+\varepsilon}}) \geq \operatorname{vol}(S_{p-2/w(\partial S_p)}) \geq \operatorname{vol}(S_p) + \frac{c \operatorname{vol}(S_p)^{1/2+\varepsilon}}{2},$$

where in the first inequality we also used that $\operatorname{vol}(S_p)$ increases as p decreases. Now, iterating this procedure $2 \operatorname{vol}(S_p)^{1/2-\varepsilon}/c$ times we obtain

$$\operatorname{vol}\left(S_{p-\frac{4}{c^2 \operatorname{vol}(S_p)^{2\varepsilon}}}\right) = \operatorname{vol}\left(S_{p-\frac{2}{c \operatorname{vol}(S_p)^{1/2+\varepsilon}} \cdot \frac{2 \operatorname{vol}(S_p)^{1/2-\varepsilon}}{c}}\right) \geq 2 \operatorname{vol}(S_p), \tag{3.3}$$

as $\operatorname{vol}(S_p)$ increases as p decreases. We set $p_0 = p(s)$, then $\operatorname{vol}(S_{p_0}) = \deg(s)$. Inductively define

$$p_{k+1} = p_k - \frac{4}{c^2 \operatorname{vol}(S_{p_k})^{2\varepsilon}}.$$

Then, using the inequality (3.3), we have

$$\operatorname{vol}(S_{p_{k+1}}) \geq 2 \cdot \operatorname{vol}(S_{p_k}). \tag{3.4}$$

Note that we can run the above procedure as long as $\operatorname{vol}(S_p) \leq \operatorname{vol}(G)/2$. Therefore, for some $k^\star \lesssim \log \frac{\operatorname{vol}(G)}{\deg(s)}$, we must have

$$\operatorname{vol}(G) \geq 2 \cdot \operatorname{vol}(S_{p_{k^\star}}) \geq \operatorname{vol}(G)/2.$$

Therefore,

$$p_0 - p_{k^\star} \leq 4 \cdot \sum_{j=0}^{k^\star} \frac{1}{c^2 \operatorname{vol}(S_{p_i})^{2\varepsilon}}.$$

Using (3.4) we get

$$p_0 - p_{k^\star} \lesssim \frac{1}{c^2 \operatorname{vol}(S_0)^{2\varepsilon}} \cdot \sum_{j=0}^{k^\star} \frac{1}{2^{2j\varepsilon}} \lesssim \frac{1}{\deg(s)^{2\varepsilon} \cdot c^2 \cdot \varepsilon},$$

where the last inequality is a geometric sum with ratio $\approx 1/(1+\epsilon)$.

By a similar argument (sending flow from t to s), we see that more than half of the vertices have potential smaller than

$$\frac{1}{\deg(t)^{2\varepsilon}} \cdot \frac{1}{\varepsilon \cdot c^2}.$$

Combining these two bounds, we obtain

$$\mathsf{Reff}(s,t) = p(s) \lesssim \left(\frac{1}{\deg(s)^{2\varepsilon}} + \frac{1}{\deg(t)^{2\varepsilon}} \right) \cdot \frac{1}{\varepsilon \cdot c^2},$$

where the equality follows since the flow is a unit flow. ◀

▶ Remark. For our proof to go through, we do not need the mild expansion condition to be satisfied by all cuts. It suffices to have this condition satisfied by electric potential threshold cuts (S_p, S_p^c) only.

For computational purposes, it will be important to show that our argument is robust to small perturbations in the potentials, i. e. we need to show that the proof will still go through when we are working with threshold cuts with respect to a vector $\hat{p}$ which is close to the electric potential vector $\boldsymbol{p}$, rather than working with the potential vector $\boldsymbol{p}$ directly.

This is shown in Appendix A of the arxiv version of our paper [3].

3.1 Finding the Sparse Cuts Algorithmically

Next we prove Corollary 2.

▶ **Corollary 6.** *Let $G = (V, E, \boldsymbol{w})$ be a weighted undirected graph. If $\deg(v) \geq 1/\alpha$ for all $v \in V$, then for any $0 < \varepsilon < 1/2$, there is a subset of vertices $U \subseteq V$ such that*

$$\Phi(U) \lesssim \frac{\alpha^\varepsilon}{\sqrt{\mathcal{R}_{diam} \cdot \varepsilon}} \cdot \operatorname{vol}(U)^{\varepsilon - 1/2}.$$

Furthermore, the set U can be found in time $\widetilde{\mathcal{O}}\Big(m \cdot \log\Big(\frac{w(E)}{\min_e w(e)}\Big)\Big)$.

Proof. First, we prove the existence of U. Let $u, v \in V$ such that

$$\mathsf{Reff}(u,v) = \mathcal{R}_{diam}. \tag{3.5}$$

The choice of

$$c \asymp \sqrt{\frac{\frac{1}{\deg(u)^{2\varepsilon}} + \frac{1}{\deg(v)^{2\varepsilon}}}{\mathsf{Reff}(u,v) \cdot \varepsilon}} \quad \text{ensures} \quad \mathsf{Reff}(u,v) > \left(\frac{1}{\deg(s)^{2\varepsilon}} + \frac{1}{\deg(t)^{2\varepsilon}} \right) \cdot \frac{1}{\varepsilon \cdot c^2}. \tag{3.6}$$

Then, by Theorem 1, there must be a threshold set S_p of the potential vector $\boldsymbol{p}$ corresponding to sending one unit of electrical flow from u to v such that

$$\Phi(U) \lesssim \frac{c}{\mathrm{vol}(U)^{1/2-\varepsilon}} \lesssim \frac{\alpha^{\varepsilon}}{\sqrt{\varepsilon \cdot \mathcal{R}_{diam}}} \cdot \mathrm{vol}(U)^{\varepsilon-1/2},$$

where the last inequality follows from our assumption that $\deg(v) \geq 1/\alpha$ for all $v \in V$. This proves the first part of the corollary.

It remains to devise a near linear time algorithm to find the set U. First, suppose that we are given the optimum pair of vertices u, v satisfying (3.5). Using the Spielman-Teng solver (Lemma 4), we can compute the potential vector $\boldsymbol{p}$ corresponding to sending one unit of electrical flow from to u to v in time $\widetilde{\mathcal{O}}\Big(m \cdot \log\Big(\frac{w(E)}{\min_e w(e)}\Big)\Big)$. We can then sort the vertices by their potential values in time $\mathcal{O}(n \log n) = \widetilde{\mathcal{O}}(m)$. Finally, we simply go over the sorted list and find the least expanding level set. This can be done in $\mathcal{O}(m)$ time in total, since getting $\partial S_{p(v_i)}$ from $\partial S_{p(v_{i+1})}$ (resp. $\mathrm{vol}(S_{p(v_i)})$ from $\mathrm{vol}(S_{p(v_{i+1})})$) can be done by considering the $\deg(v_i)$ edges $e \in \partial(v_i)$ incident to v_i.

It remains to find such an optimal pair of vertices u, v satisfying (3.5). Instead, we find a pair of vertices u', v' such that $\mathsf{Reff}(u', v') \geq \mathcal{R}_{diam}/3$, which is enough for our purposes as this only causes a constant factor loss in the conductance of U.

▶ **Lemma 7.** *Let G be a weighted graph. In time $\widetilde{\mathcal{O}}(m)$, one can compute a pair of vertices $u, v \in V$ satisfying*

$$\mathsf{Reff}(u, v) \geq \mathcal{R}_{diam}/3.$$

Proof. By the triangle inequality for effective resistances, we have the following inequality for any $u \in V$:

$$\mathcal{R}_{diam} \leq 2 \max_{v \in V} \mathsf{Reff}(u, v). \tag{3.7}$$

Thus, we fix a $u \in V$. Appyling Lemma 5 (with $S = \{u\} \times V$), we get the numbers $A(u, v)$ which multiplicatively approximate $\mathsf{Reff}(u, v)$ within a factor e^{β}. Let $v^* = \arg\max_{v \in V} A(u, v)$. By combining the inequality (3.7) with

$$\max_{v \in V} \mathsf{Reff}(u, v) \leq e^{\beta} \max_{v \in V} A(u, v) = e^{\beta} A(u, v^*) \leq e^{2\beta}\, \mathsf{Reff}(u, v^*),$$

we obtain $\mathsf{Reff}(u, v^*) \geq \mathcal{R}_{diam}/3$ for some $\beta = \Theta(1)$. The algorithm consists of an application of Lemma 5 with $|S| = n$, and a linear scan for finding the maximum. Hence, the time bound follows. ◀

So, Corollary 2 follows by first using Lemma 7 to find u', v' with $\mathsf{Reff}(u', v') \geq R/3$, and then apply Theorem 1 with the choice of c as described in (3.6). ◀

▶ Remark. We have avoided treating the issues caused by working with an approximate potential vector for the sake of clarity. This issue is addressed in Appendix A of the arxiv version of our paper [3].

4 Low Effective Resistance Diameter Graph Decomposition

In this section we prove Theorem 3.

▶ **Theorem 3** (Main). *Given a weighted undirected graph $G = (V, E, \boldsymbol{w})$, and a large enough parameter $\delta > 1$, there is an algorithm with time complexity $\widetilde{\mathcal{O}}\Big(m \cdot n \cdot \log\Big(\frac{w(E)}{\min_e w(e)}\Big)\Big)$ that finds a partition $V = \bigcup_{i=1}^{h} V_i$ satisfying*

$$w\left(E - \bigcup_{i=1}^{h} E(V_i)\right) \lesssim \frac{w(E)}{\delta} \qquad \text{(loss bound)}$$

and

$$\mathcal{R}_{diam}(G[V_i]) \lesssim \delta^3 \cdot \frac{n}{w(E)} \qquad \text{(resistance bound)}$$

for all $i = 1, \ldots, h$.

Proof. Let R be the target effective resistance diameter and W be the target sum of the weights of edges that we are going to cut. We will write the algorithm in terms of R, W, and we will optimize for these parameters later in the proof. Note that $n = |V|$ is the number of vertices of the original graph G, and it is fixed throughout the execution of the following algorithm.

Algorithm 1 Effective Resistance Partitioning

Input A graph H, and parameters R, W, n.
Output A partition $\mathcal{P} = \{V_i \mid i = 1, \ldots, h\}$ of $V(H)$.

1. If there is a vertex $v \in V(H)$ such that $\deg_H(v) \le W/(2n)$, then delete all the edges incident to v. Repeat this step until there are no such vertices in the remaining graph H.
2. Use Lemma 7 to find vertices u, v such that $\mathsf{Reff}(u, v) \ge \mathcal{R}_{diam}(H)/3$.
3. If $\mathsf{Reff}(u, v) \le R$, return $\{V(H)\}$.
4. Otherwise, find the cut (U, U^c) with $\Phi_H(U) \lesssim \frac{(n/W)^\varepsilon}{\sqrt{\varepsilon \cdot R}} \cdot \mathrm{vol}_H(U)^{\varepsilon - 1/2}$ by invoking Corollary 2, with minimum degree at least $W/(2n)$ and $\varepsilon = 1/4$.
5. Call the algorithm recursively on $H[U]$ and $H[U^c]$.
6. Return the union of the outputs of both recursive calls.

First of all, by construction, every set V_i in the output partition satisfies $\mathcal{R}_{diam}(G[V_i]) \le 3R$. It is not hard to see that the running time is $\widetilde{\mathcal{O}}(n \cdot m \cdot \log(w(E)/\min_e w(e)))$, as the most expensive of the above algorithm takes time $\widetilde{\mathcal{O}}(m \cdot \log(w(E)/\min_e w(e)))$, and we make at most n recursive calls.

It remains to calculate the sum of the weights of all edges that we cut. Note that we cut edges either when a vertex has a low degree or when we find a low conductance set U. We classify the cut edges into two types as follows:

- **(i)** Edges e where e is cut as an incident edge of a vertex v with $\deg_H(v) \le W/2n$.
- **(ii)** The rest of the edges, i. e., edges e where $e \in \partial_H(U)$ for some U where $\Phi_H(U) \lesssim \frac{(n/W)^\varepsilon}{\sqrt{\varepsilon \cdot R}} \mathrm{vol}_H(U)^{\varepsilon - 1/2}$.

We observe that we are going to remove edges of type (i) for at most n times, because each such removal isolates a vertex of G. So, the sum of the weights of edges of type (i) that we cut is at most $n \cdot W/2n \le W/2$. It remains to bound the sum of the weight of edges of type (ii) that we cut.

We use an amortization argument: Let $\Psi(e)$ stand for the tokens charged from an edge. We assume that for each edge $e \in E$, the number of tokens $\Psi(e)$ is initially set to 0. Every

time we make a cut of type (ii), we assume without loss of generality that $\mathrm{vol}_H(U) \leq \mathrm{vol}(H)/2$ and we modify the number of tokens as follows

$$\Psi(e) := \begin{cases} \Psi(e) + \frac{w(\partial_H U)}{w(E_H(U))} & \text{if } e \in E_H(U) \\ \Psi(e) & \text{otherwise.} \end{cases} \tag{4.1}$$

By definition, after the termination of the algorithm, we have

$$w(\text{set of cut edges of type (ii)}) = \sum_{e \in E} \Psi(e) \cdot w(e). \tag{4.2}$$

Therefore, to bound the total weight of type (ii) edges that are cut, it is enough to show that no edge is charged with too many tokens provided R is large enough.

▶ **Claim 8.** *If $R \gtrsim n/(\varepsilon W)$, we will have $\Psi(e) \lesssim \frac{4}{\sqrt{R \cdot W/8n - 1}}$ for all edges $e \in E$ after the termination of the algorithm.*

Proof. Fix an edge $e \in E$. Let $\Delta\,\Psi(e)$ be the increment of $\Psi(e)$ due to a cut (U, U^c). We have

$$\Delta\,\Psi(e) = \frac{w(\partial_H U)}{w(E_H(U))} = 2 \cdot \frac{w(\partial_H U)}{\mathrm{vol}_H(U) - w(\partial_H U)} = 2 \cdot \frac{1}{\frac{1}{\Phi_H(U)} - 1} \lesssim \frac{2c}{\mathrm{vol}_H(U)^{1/2-\varepsilon} - c}, \tag{4.3}$$

where c is chosen as in (3.6) in the proof of Corollary 2 so that $\Phi(U) \leq c/\,\mathrm{vol}(U)^{1/2-\varepsilon}$ for the last inequality to hold. Since the minimum degree is at least $W/2n$ by Step (1) of the algorithm, we have

$$c \asymp \frac{(2n/W)^{\varepsilon}}{\sqrt{\varepsilon \cdot R}}.$$

The minimum degree condition also implies that $\mathrm{vol}_H(U) \geq W/(2n)$. Note that the denominator of the rightmost term of (4.3) is non-negative as long as $\mathrm{vol}_H(U)^{1/2-\epsilon} \geq (W/2n)^{1/2-\varepsilon} \geq c$, which holds when $R \gtrsim n/(\varepsilon W)$.

Let $U_0 \subseteq V(H_0)$ be the set for which e was charged for the last time, and in general $U_k \subseteq V(H_k)$ be the k-th last set for which e was charged. We write $\Delta_k \Psi(e)$ to denote the increment in $\Psi(e)$ due to U_k.

Note that by (4.1) we have $e \in E_{H_i}(U_i)$ for all i. Furthermore, since $\mathrm{vol}_{H_i}(U_i) \leq \mathrm{vol}(H_i)/2 \leq \mathrm{vol}_{H_{i+1}}(U_{i+1})/2$ for all i, we have

$$\mathrm{vol}_{H_k}(U_k) \geq 2^k\,\mathrm{vol}_{H_0}(U_0) \tag{4.4}$$

for all $k \geq 0$. Therefore, using (4.3) and (4.4), we can write

$$\begin{aligned} \Psi(e) = \sum_{k \geq 0} \Delta_k\,\Psi(e) &\leq \sum_{k \geq 0} \frac{2c}{\mathrm{vol}_{H_i}(U_i)^{1/2-\varepsilon} - c} \\ &\leq \sum_{k \geq 0} \frac{2c}{(2^k\,\mathrm{vol}_{H_0}(U_0))^{1/2-\varepsilon} - c} \\ &\leq \frac{2c}{\mathrm{vol}_{H_0}(U_0)^{1/2-\varepsilon} - c} \cdot \sum_{k \geq 0} \frac{1}{(2^{1/2-\varepsilon})^k}, \end{aligned}$$

where the last inequality assumes that $\varepsilon < 1/2$. As argued before, the minimum degree condition implies that every vertex is of degree at least $W/2n$ and thus $\mathrm{vol}_{H_0}(U_0) \geq W/(2n)$. Therefore, by the geometric sum formula, we have

$$\Psi(e) \leq \frac{2}{\frac{1}{c}(W/2n)^{1/2-\varepsilon} - 1} \cdot \frac{1}{1 - 2^{\varepsilon - 1/2}}.$$

Plugging the value of c and setting $\varepsilon = 1/4 < 1/2$, we conclude that

$$\Psi(e) \lesssim \frac{2}{\sqrt{\varepsilon \cdot R \cdot W/2n} - 1} \leq \frac{4}{\sqrt{R \cdot W/8n} - 1}. \quad \blacktriangleleft$$

Setting $R \asymp \delta^2 \cdot n/W$ for a sufficiently large $\delta^2 > 1$ so that the assumption of Claim 8 is satisfied, it follows from (4.2) that the sum of the weights of all cut edges is at most

$$W/2 + \sum_e \Psi(e) \cdot w(e) \lesssim W/2 + \frac{w(E)}{\delta}.$$

Setting $W = w(E)/\delta$ proves the theorem. This completes the proof of Theorem 3. $\blacktriangleleft$

5 Conclusions and Open Problems

We have shown that we can decompose a graph into components of bounded effective resistance diameter while losing only a small number of edges. There are few questions which arise naturally from this work.

1. Can the decomposition in Theorem 3 be computed in near linear time? Is this decomposition useful in generating a random spanning tree?
2. For the Unique Games Conjecture, Theorem 3 implies that we can restrict our attention to graphs with bounded effective resistance diameter. Can we solve Unique Games instances better in such graphs? More generally, are there some natural and nontrivial problems that can be solved effectively in graphs of bounded effective resistance diameter?
3. Is there a generalization of Theorem 1 to multi-partitioning, i. e. does the existence of k-vertices with high pairwise effective resistance distance help us in finding a k-partitioning of the graph where every cut is very sparse?
4. Theorem 1 says that a small-set expander has bounded effective resistance diameter. Is it possible to strengthen Theorem 3 to show that every graph can be decomposed into small-set expanders? This may be used to show that the Small-Set Expansion Conjecture and the Unique Games Conjecture are equivalent, depending on the quantitative bounds.

Acknowledgements. We would like to thank Hong Zhou for helpful discussions and anonymous referees for their useful suggestions.

References

1 Ittai Abraham, Cyril Gavoille, Anupam Gupta, Ofer Neiman, and Kunal Talwar. Cops, robbers, and threatening skeletons: padded decomposition for minor-free graphs. In *46th Annual Symposium on Theory of Computing*, pages 79–88, 2014.

2 David Aldous. The random walk construction of uniform spanning trees and uniform labelled trees. *SIAM J. Discrete Math.*, 3(4):450–465, 1990.

3 Vedat Levi Alev, Nima Anari, Lap Chi Lau, and Shayan Oveis Gharan. Graph clustering using effective resistance. *ArXiv e-prints*, 2017. `arXiv:1711.06530`.

4 Vedat Levi Alev and Lap Chi Lau. Approximating unique games using low diameter graph decomposition. In *APPROX/RANDOM 2017*, pages 18:1–18:15, 2017.

5 Noga Alon and V. D. Milman. lambda$_1$, isoperimetric inequalities for graphs, and superconcentrators. *J. Comb. Theory, Ser. B*, 38(1):73–88, 1985.

6 Nima Anari and Shayan Oveis Gharan. Effective-resistance-reducing flows, spectrally thin trees, and asymmetric TSP. In *56th Annual Symposium on Foundations of Computer Science*, pages 20–39, 2015.

7 Sanjeev Arora, Boaz Barak, and David Steurer. Subexponential algorithms for unique games and related problems. In *51th Annual Symposium on Foundations of Computer Science*, pages 563–572, 2010.

8 Sanjeev Arora, Subhash Khot, Alexandra Kolla, David Steurer, Madhur Tulsiani, and Nisheeth K. Vishnoi. Unique games on expanding constraint graphs are easy. In *40th Annual Symposium on Theory of Computing*, pages 21–28, 2008.

9 Nikhil Bansal, Uriel Feige, Robert Krauthgamer, Konstantin Makarychev, Viswanath Nagarajan, Joseph Naor, and Roy Schwartz. Min-max graph partitioning and small set expansion. In *52nd Annual Symposium on the Theory of Computation*, pages 17–26. IEEE, 2011.

10 Boaz Barak, Prasad Raghavendra, and David Steurer. Rounding semidefinite programming hierarchies via global correlation. In *52nd Annual Symposium on Foundations of Computer Science*, pages 472–481, 2011.

11 Yair Bartal. Probabilistic approximations of metric spaces and its algorithmic applications. In *37th Annual Symposium on Foundations of Computer Science*, pages 184–193, 1996.

12 Punyashloka Biswal, James R. Lee, and Satish Rao. Eigenvalue bounds, spectral partitioning, and metrical deformations via flows. *J. ACM*, 57(3):13:1–13:23, 2010.

13 Andrei Z. Broder. Generating random spanning trees. In *30th Annual Symposium on Foundations of Computer Science*, pages 442–447, 1989.

14 Andrei Z Broder and Anna R Karlin. Bounds on the cover time. *Journal of Theoretical Probability*, 2(1):101–120, 1989.

15 Gruia Călinescu, Howard J. Karloff, and Yuval Rabani. Approximation algorithms for the 0-extension problem. In *12th Annual Symposium on Discrete Algorithms*, pages 8–16, 2001.

16 Ashok K. Chandra, Prabhakar Raghavan, Walter L. Ruzzo, Roman Smolensky, and Prasoon Tiwari. The electrical resistance of a graph captures its commute and cover times. *Computational Complexity*, 6(4):312–340, 1997.

17 Moses Charikar, Chandra Chekuri, To-Yat Cheung, Zuo Dai, Ashish Goel, Sudipto Guha, and Ming Li. Approximation algorithms for directed steiner problems. In *9th Annual Symposium on Discrete Algorithms*, pages 192–200, 1998.

18 Jeff Cheeger. A lower bound for the smallest eigenvalue of the laplacian. *Problems in analysis*, pages 195–199, 1970.

19 Paul Christiano, Jonathan A. Kelner, Aleksander Madry, Daniel A. Spielman, and Shang-Hua Teng. Electrical flows, laplacian systems, and faster approximation of maximum flow in undirected graphs. In *43rd Symposium on Theory of Computing*, pages 273–282, 2011.

20 David Durfee, Rasmus Kyng, John Peebles, Anup B. Rao, and Sushant Sachdeva. Sampling random spanning trees faster than matrix multiplication. In *49th Annual Symposium on Theory of Computing*, pages 730–742, 2017.

21 Jittat Fakcharoenphol, Chris Harrelson, Satish Rao, and Kunal Talwar. An improved approximation algorithm for the 0-extension problem. In *14th Annual Symposium on Discrete Algorithms*, pages 257–265, 2003.

22 Uriel Feige, MohammadTaghi Hajiaghayi, and James R. Lee. Improved approximation algorithms for minimum weight vertex separators. *SIAM J. Comput.*, 38(2):629–657, 2008.

23 Anupam Gupta and Kunal Talwar. Approximating unique games. In *17th AnnualSymposium on Discrete Algorithms*, pages 99–106, 2006.

24 Venkatesan Guruswami and Ali Kemal Sinop. Lasserre hierarchy, higher eigenvalues, and approximation schemes for graph partitioning and quadratic integer programming with PSD objectives. In *52nd Annual Symposium on Foundations of Computer Science*, pages 482–491, 2011.

25 Ravi Kannan, Santosh Vempala, and Adrian Vetta. On clusterings: Good, bad and spectral. *J. ACM*, 51(3):497–515, 2004. doi:10.1145/990308.990313.

26 Jonathan A. Kelner, James R. Lee, Gregory N. Price, and Shang-Hua Teng. Higher eigenvalues of graphs. In *50th Annual Symposium on Foundations of Computer Science*, pages 735–744, 2009.

27 Jonathan A. Kelner and Aleksander Madry. Faster generation of random spanning trees. In *50th Annual Symposium on Foundations of Computer Science*, pages 13–21, 2009.

28 Jonathan A. Kelner, Lorenzo Orecchia, Aaron Sidford, and Zeyuan Allen Zhu. A simple, combinatorial algorithm for solving SDD systems in nearly-linear time. In *45th Annual Symposium on Theory of Computing*, pages 911–920, 2013.

29 Gustav Kirchhoff. Ueber die auflösung der gleichungen, auf welche man bei der untersuchung der linearen vertheilung galvanischer ströme geführt wird. *Annalen der Physik*, 148(12):497–508, 1847.

30 Douglas J Klein and Milan Randić. Resistance distance. *Journal of mathematical chemistry*, 12(1):81–95, 1993.

31 Philip Klein, Serge A Plotkin, and Satish Rao. Excluded minors, network decomposition, and multicommodity flow. In *Proceedings of the twenty-fifth annual ACM symposium on Theory of computing*, pages 682–690. ACM, 1993.

32 Alexandra Kolla. Spectral algorithms for unique games. *Computational Complexity*, 20(2):177–206, 2011.

33 Ioannis Koutis, Gary L. Miller, and Richard Peng. Approaching optimality for solving SDD linear systems. In *51th Annual Symposium on Foundations of Computer Science*, pages 235–244, 2010.

34 Ioannis Koutis, Gary L. Miller, and Richard Peng. A nearly-m log n time solver for SDD linear systems. In *52nd Annual Symposium on Foundations of Computer Science*, pages 590–598, 2011.

35 Robert Krauthgamer and Tim Roughgarden. Metric clustering via consistent labeling. *Theory of Computing*, 7(1):49–74, 2011.

36 Rasmus Kyng and Sushant Sachdeva. Approximate gaussian elimination for laplacians - fast, sparse, and simple. In *57th Annual Symposium on Foundations of Computer Science*, pages 573–582, 2016.

37 James R. Lee, Shayan Oveis Gharan, and Luca Trevisan. Multiway spectral partitioning and higher-order cheeger inequalities. *J. ACM*, 61(6):37:1–37:30, 2014.

38 James R. Lee and Anastasios Sidiropoulos. Genus and the geometry of the cut graph. In *21st Annual Symposium on Discrete Algorithms*, pages 193–201, 2010.

39 Nathan Linial and Michael E. Saks. Low diameter graph decompositions. *Combinatorica*, 13(4):441–454, 1993.

40 Aleksander Madry, Damian Straszak, and Jakub Tarnawski. Fast generation of random spanning trees and the effective resistance metric. In *26th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 2019–2036, 2015.

41 Peter Matthews. Covering problems for brownian motion on spheres. *The Annals of Probability*, pages 189–199, 1988.

42 Shayan Oveis Gharan and Luca Trevisan. Partitioning into expanders. In *25th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1256–1266, 2014.

43 Daniel A. Spielman and Nikhil Srivastava. Graph sparsification by effective resistances. *SIAM J. Comput.*, 40(6):1913–1926, 2011.

44 Daniel A. Spielman and Shang-Hua Teng. Spectral sparsification of graphs. *SIAM J. Comput.*, 40(4):981–1025, 2011.

45 Daniel A. Spielman and Shang-Hua Teng. Nearly linear time algorithms for preconditioning and solving symmetric, diagonally dominant linear systems. *SIAM J. Matrix Analysis Applications*, 35(3):835–885, 2014.

Lattice-based Locality Sensitive Hashing is Optimal*

Karthekeyan Chandrasekaran[1], Daniel Dadush[2], Venkata Gandikota[3], and Elena Grigorescu[4]

1 **University of Illinois, Urbana-Champaign, USA**
karthe@illinois.edu
2 **Centrum Wiskunde & Informatica, Amsterdam, The Netherlands**
dndadush@gmail.com
3 **Purdue University, West Lafayette, USA**
vgandiko@purdue.edu
4 **Purdue University, West Lafayette, USA**
elena-g@purdue.edu

Abstract

Locality sensitive hashing (LSH) was introduced by Indyk and Motwani (STOC '98) to give the first sublinear time algorithm for the c-approximate nearest neighbor (ANN) problem using only polynomial space. At a high level, an LSH family hashes "nearby" points to the same bucket and "far away" points to different buckets. The quality of measure of an LSH family is its LSH exponent, which helps determine both query time and space usage.

In a seminal work, Andoni and Indyk (FOCS '06) constructed an LSH family based on *random ball partitionings* of space that achieves an LSH exponent of $1/c^2$ for the ℓ_2 norm, which was later shown to be optimal by Motwani, Naor and Panigrahy (SIDMA '07) and O'Donnell, Wu and Zhou (TOCT '14). Although optimal in the LSH exponent, the ball partitioning approach is computationally expensive. So, in the same work, Andoni and Indyk proposed a simpler and more practical hashing scheme based on *Euclidean lattices* and provided computational results using the 24-dimensional Leech lattice. However, no theoretical analysis of the scheme was given, thus leaving open the question of finding the exponent of lattice based LSH.

In this work, we resolve this question by showing the existence of lattices achieving the optimal LSH exponent of $1/c^2$ using techniques from the geometry of numbers. At a more conceptual level, our results show that optimal LSH space partitions can have *periodic structure*. Understanding the extent to which additional structure can be imposed on these partitions, e.g. to yield low space and query complexity, remains an important open problem.

1998 ACM Subject Classification E.1 Data Structures

Keywords and phrases Locality Sensitive Hashing, Approximate Nearest Neighbor Search, Random Lattices

Digital Object Identifier 10.4230/LIPIcs.ITCS.2018.42

1 Introduction

Nearest neighbor search (NNS) is a fundamental problem in data structure design. Here, we are given a database P of n points in a metric space X, and the goal is to build a data structure that can quickly return a closest point in the database to any queried target. In

* A full version of the paper is available at https://arxiv.org/abs/1712.08558.

its exact form, the problem is known to suffer from the curse of dimensionality, where data structures that beat brute force search (i.e. a linear scan through the data points) require either space or query time exponential in the dimension of the space X. To circumvent this issue, Indyk and Motwani [20] studied a relaxed version of NNS which allowed for both *approximation* and *randomization*. In (c, r)-approximate nearest neighbor search (ANN), we are given an approximation factor $c \geq 1$ and distance threshold $r > 0$, where we must guarantee that for a query q, if $d_X(q, P) \leq r$ then the data structure returns $p \in P$ such that $d_X(q, p) \leq cr$. When we allow randomization, we only require that any fixed query succeeds with good probability over the randomness used to construct the data structure.

In order to address ANN, Indyk and Motwani introduced the concept of Locality Sensitive Hashing (LSH). A locality sensitive hash function maps "nearby" points together and "far away" points apart. Indyk and Motwani showed that such LSH function families can be used to build data structures with both sublinear query time and subquadratic space for ANN. LSH is now one of the most popular methods for solving ANN and has found many applications in areas such as cryptanalysis [23, 10], information retrieval and machine learning (see [29] for a survey). Important metric spaces for LSH include $\{0, 1\}^d$ or $\mathbb{R}^d$ under ℓ_1 or ℓ_2-norms, and the sphere S^{d-1} under angular distance. In this work, we focus on $\mathbb{R}^d$ under the ℓ_2-norm.

Let $\mathcal{H}$ be a family of functions with an associated probability distribution. An LSH family $\mathcal{H}$ is (c, r, p_1, p_2)-sensitive for X if a randomly chosen hash function h from $\mathcal{H}$ maps any two points in X at distance at most r to the same bucket with probability at least p_1 and any two points in X at distance at least cr to the same bucket with probablity at most p_2. The measure of quality of the LSH family is the so-called LSH exponent $\rho := \ln(1/p_1)/\ln(1/p_2)$. If $X = (\mathbb{R}^d, \ell_2)$ and the maximum computational time for evaluating the hash function $h(x)$ at any point $x \in X$ for any element $h \in \mathcal{H}$ is at most κ, then one can build a randomized (c, r)-ANN data structure that answers queries in $O((d + \kappa)n^{\rho(c)} \log_{1/p_2}(n))$ time using $O(dn + n^{1+\rho(c)})$ space [20, 19]. Similar results hold for other d-dimensional metric spaces. Consequently, much research effort has been directed at constructing LSH families with both low LSH exponent and fast evaluation times.

For the ℓ_2-norm, the first results [20, 18] gave constructions achieving an exponent $1/c \pm o(1)$ for X being the hypercube $\{0, 1\}^d$, which was later extended to all of $\mathbb{R}^d$ in [14]. For the ℓ_2-norm over $X = \mathbb{R}^d$, Andoni and Indyk [4] gave the first construction of an LSH hash family achieving a limiting exponent of $1/c^2$, which was later shown to be optimal in [25, 26]. We note that optimality here holds only for "classical" LSH, in which the LSH family depends only on the ambient metric space and not on the database itself, and that these lower bounds have been recently circumvented using more sophisticated data dependent approaches [6, 8], which we discuss later.

While achieving the optimal exponent, the hash functions from Andoni and Indyk's work [4] are unfortunately quite expensive to evaluate. Their hash function family can be described as follows: For a design dimension k, a function from the family corresponds to $k^{O(k)}$ random shifts $t_1, t_2, \ldots$ of the integer lattice $\mathbb{Z}^k$ which satisfy that every point in $\mathbb{R}^k$ is at ℓ_2 distance at most $1/4$ from at least one shift. To map the database and the queries into $\mathbb{R}^k$, the hash function uses a Gaussian random projection G mapping $\mathbb{R}^d$ to $\mathbb{R}^k$. The hash value on query q then equals the closest vector to Gq in $\mathbb{Z}^k + t_i$, where i is the first index such that Gq is at distance at most $1/4$ from some point in $\mathbb{Z}^k + t_i$. For this family they prove an upper bound on the LSH exponent of $1/c^2 + O(\log k/\sqrt{k})$, which tends to $1/c^2$ as $k \to \infty$. Note that storing the description of this hash function requires $k^{O(k)}$ space and evaluating it requires iterating over all shifts which takes $k^{O(k)}$ time. This prohibitive space usage and running time restricted the use of these hash functions to only very low

dimensions in the context of ANN (i.e. k is restricted to be a very slow growing function of the number of points n in the database), yielding a rather slow convergence to the optimal $1/c^2$ exponent.

Lattice based LSH. Motivated by the above-mentioned drawbacks, Andoni and Indyk [4] proposed a simpler and more practical LSH scheme based on *Euclidean Lattices*. A k-dimensional lattice $L \subset \mathbb{R}^k$ given by a collection of basis vectors $B = (b_1, \ldots, b_k)$ is defined to be all integer linear combinations of $b_1, \ldots, b_k$. The *determinant* of L is defined as $|\det(B)|$, which we note is invariant to the choice of basis. In lattice based LSH, one simply replaces the $k^{O(k)}$ shifts of $\mathbb{Z}^k$ by a single random shift $t \in \mathbb{R}^k$ of a lattice L, and the hash value on query q now becomes the closest vector to Gq in $L + t$.

We note that the last step of the hashing algorithm corresponds to solving the *closest vector problem* (CVP) on L, i.e. given a target point q one must compute a closest vector to x in L under the ℓ_2 norm. While this problem is NP-Hard in the worst case [22], in analogy to coding, one has complete freedom to *design the lattice*. Thus the main potential benefit of lattice based LSH is that one may hope to find "LSH-good" lattices (i.e., lattices with good LSH exponent) for which CVP can be solved quickly (at least much faster than enumerating over a ball partition). A secondary benefit is that the corresponding hash functions require very little storage compared to the ball partitions, namely just a single shift vector t together with the projection matrix G are sufficient (note that the lattice is shared across all instantiations of the hash function). To evaluate lattice based LSH, Andoni and Indyk [4] provided experimental results for L being the 24 dimensional Leech lattice equipped with the decoder of [3]. A version of this scheme with the 8 dimensional E8 lattice has also been implemented and tested in [21], and a parallelized GPU implementation of the Leech lattice scheme was tested in [11].

The following natural question was left open in the work of Andoni and Indyk: can the space partitions induced by lattices achieve the optimal LSH constant for the ℓ_2-norm? Note that for a lattice L, the associated space partition corresponds to a random shift of the tiling of space $\{y + \mathcal{V}_L : y \in L\}$, where $\mathcal{V}_L$ is the *Voronoi cell* of the lattice, i.e. the set of all points closer to the origin than to any other lattice point.

Our Contribution. As our main result, we resolve this question in the affirmative. We show that for any fixed approximation factor $c > 1$, there exists a sequence of lattices $\{L_{k,c} \subset \mathbb{R}^k : k \geq 1\}$, where $L_{k,c}$ has an associated LSH exponent for ℓ_2-norm bounded by $1/c^2 + O(1/\sqrt{k})$. We note that this is slightly better than the rate of convergence to optimality proven by Andoni and Indyk in [4] for the ball partitioning approach. To prove this result, we rely on the probabilistic method, using a delicate averaging argument over the space of all lattices of determinant 1.

Our result is currently non-constructive, as we lack the appropriate concentration results for the LSH collision probabilities, though we believe this should be achievable. A simple and efficient sampling algorithm for the random lattice distribution that we employ – known as the Siegel measure over lattices – was given by Ajtai [2], and we expect that a lattice sampled from this distribution should be "LSH-good" (in terms of the LSH exponent) with high probability. Perhaps a more significant issue is that for the same dimension k, the probabilistic argument may produce different lattices for different approximation factors. Resolving this issue would require a much finer understanding of the shape of the collision probability curve (currently, we can only control the curve at two points), and we leave this as as an open problem. We note however, that if one allows for sampling a different random

lattice for each hash function instantiation, as opposed to a single lattice shared by all instantiations, then our methods are indeed constructive. We find this approach somewhat less appealing however, since in general the cost of preprocessing a lattice in the context of CVP, say computing a short basis, the Voronoi cell, etc., is substantial, and hence it is desirable to only have to perform such preprocessing once. Furthermore, since the end goal is eventually to find a class of LSH good lattices with fast decoding algorithms, our main contribution here is to show that LSH good lattices do in fact exist.

From the perspective of the complexity of ANN, LSH-good lattices (when given as advice to an ANN algorithm) provide a slight improvement over [4] when using any of the recent $2^{k+o(k)}$-time and $2^{k+o(k)}$-space algorithms for the closest vector problem [13, 1] to implement the hash queries. In particular, for (c, r)-ANN on an n element database in $\mathbb{R}^d$, by choosing the dimension of the lattice to be $k = \log^{2/3}(n)$, we get query time dn^{ρ} using $dn + n^{1+\rho}$ space where $\rho = 1/c^2 + O(1/\log^{1/3}(n))$. These complexity results for ANN are however superseded by the more recent approaches using *data dependent* LSH [6, 8], which achieve $\rho = 1/(2c^2 - 1) + o(1)$. While more sophisticated, these approaches still depend on rather impractical and expensive random space partitions – with query complexity $2^{O(\sqrt{d})}$ instead of 2^d – and hence there is still room for progress.

Given this, we view our contribution mainly as a conceptual one, namely that *structured space partitions* can be optimal. We hope that this provides additional motivation for developing space partitions which admit fast query algorithms, and in particular for finding novel classes of "spherical" lattices (LSH-good or otherwise) admitting fast CVP solvers. We note that up to present, the only known general classes of lattices for which CVP is solvable in polynomial time are lattices of Voronoi's first kind (VFK) [24] and tensor products of two root lattices [15], whose geometry is still rather restrictive (see [33] section 2.3 for an exposition of VFK lattices).

1.1 Techniques and High Level Proof Plan

The main techniques we use come from the theory of random lattices in the geometry of numbers. While getting precise estimates on an LSH collision probability for a generic high dimensional lattice seems very difficult, it turns out to be much easier to estimate the average collision probability for *random lattices*. The distribution on lattices we use is known as the Siegel measure on lattices, which is an invariant probability measure on the space of lattices of determinant 1 whose existence was established by Siegel [30] (the invariance is with respect to linear transformations of determinant 1).

A powerful point of leverage when using random lattices drawn from the Siegel distribution is that one can compute expected lattice point counts using volumes. In particular, for any Borel set $S \subseteq \mathbb{R}^k$, we have the useful identity $\mathbf{E}_L[|(L \cap S) \setminus \{0\}|] = \mathrm{vol}(S)$, i.e. the expected number of non-zero lattice points in S is equal to its volume. We will need more refined tools than this however, and in particular, we shall rely heavily on powerful probabilistic estimates of Schmidt [28] and Rogers [27] developed for the Siegel measure. More specifically, Schmidt [28] provides extremely precise estimates on the probability that a Borel set of small volume does not intersect a random lattice, while Rogers [27] gives similarly precise estimates for the relative fraction of cosets of a random lattice not intersecting a Borel set.

Using these estimates, we quickly derive clean and tight integral expressions for the average collision probabilities. From then on, the strategy is simple if rather tedious, namely, to get precise enough estimates for these integrals to be able to show that the average "near" collision probability to the power $c^2 + o(1)$ is larger than average "far" collision probability. With this inequality in hand, we immediately deduce the existence of an LSH-good lattice

from the probabilitic method. To prove that a random lattice is in fact LSH-good with high probability (making our proof constructive) it would suffice to show concentration for the relevant collision probabilities. While this seems very plausible, we leave it for future work.

Estimating the Collision Probabilities and the LSH Constant. We now give a more detailed geometric explanation of what the collision probabilities represent, how the computations for lattices differ from those for a random ball partition, and how the random lattice estimates mentioned above come into play.

We recall the lattice LSH family going from $\mathbb{R}^d$ to $\mathbb{R}^k$ induced by a lattice $L \subset \mathbb{R}^k$. We shall assume here that L has determinant 1 and hence that the Voronoi cell $\mathcal{V}_L$ of L has volume 1 (any region that tiles space with respect to L has the same volume). A function from the hash family $\mathcal{H}$ is generated as follows. First, pick a uniform random coset $t \leftarrow \mathbb{R}^k/L$ and a matrix $M \in \mathbb{R}^{k\times d}$ with i.i.d. $N(0, 1/k)$ entries (i.e. Gaussian with mean 0 and variance $1/k$). On query q, we define the hash value as $CV_L(Mq + t)$, namely the closest vector in L to $Mq + t$. Note that M is normalized here to approximately preserve distances, since $\mathbf{E}[\|Mq\|^2] = \|q\|^2$. For $x, y \in \mathbb{R}^d$, $\|x-y\|_2 = \Delta$, we wish to estimate the collision probability

$$p_\Delta := \Pr_{h\leftarrow\mathcal{H}}[h(x) = h(y)] = \Pr_{M,t}[CV_L(t + Mx) = CV_L(t + My)] \ , \tag{1}$$

where M, t are as above. We will show shortly that the right hand side indeed only depends on Δ. Using the above hash family, showing that L achieves the optimal LSH exponent for an approximation factor $c > 0$ corresponds to showing

$$\min_{\Delta>0} \ln(1/p_\Delta)/\ln(1/p_{c\Delta}) \le 1/c^2 + o(1) \ . \tag{2}$$

Note that for any desired distance threshold $r > 0$, we can always scale the database so that the scaled distance threshold becomes the minimizer above. Clearly, to be able to get a good upper bound on the LHS of (2), we have to be able to derive tight estimates for the collision probability curve p_Δ over a reasonably large range.

To understand p_Δ, we now show that the collision probability can be expressed as the probability that a uniformly sampled point in $\mathcal{V}_L$ stays inside $\mathcal{V}_L$ after a Gaussian perturbation of size Δ. Let x, y, M, t be as in (1). A first easy observation is that conditioned on any realization of $M(y - x)$, the distribution of $Mx + t$ is still uniform over cosets of $\mathbb{R}^n/L$ since t is uniform. Therefore,

$$\begin{aligned}
\Pr_{M,t}[CV_L(t + Mx) = CV_L(t + My)] &= \Pr_{M,t}[CV_L(t) = CV_L(t + M(y - x))] \\
&= \Pr_{t,g\leftarrow N(0,I_k/k)}[CV_L(t) = CV_L(t + \Delta g)] \\
&\quad \big(\text{ since } M(y-x) \text{ has distribution } N(0, \Delta^2 I_k/k) \big) \\
&= \Pr_{v\leftarrow\mathcal{V}_L, g\leftarrow N(0,I_k/k)}[v + \Delta g \in \mathcal{V}_L] \ .
\end{aligned}$$

For the last equality, note first that the Voronoi cell contains exactly one element from every coset of $\mathbb{R}^k/L$ and hence a uniformly chosen point v from $\mathcal{V}_L$ is also uniform over cosets. Lastly, by construction $CV_L(v) = 0$ and hence $CV_L(v) = CV_L(v + \Delta g) \Leftrightarrow v + \Delta g \in \mathcal{V}_L$.

At this point, without any extra information about $\mathcal{V}_L$, the task of bounding the delicate function of collision probabilities seems daunting if not intractable (note that generically $\mathcal{V}_L$ is a polytope with $2(2^k - 1)$ facets). To compare with the ball partitioning approach, it is

not hard to show that up to a factor 2, the collision probabilities there are in correspondance with the quantities

$$q_\Delta := \Pr_{u \leftarrow r_k B_2^k, g \leftarrow N(0, I_k/k)}[u + \Delta g \in r_k B_2^k],$$

where $r_k \approx \sqrt{k/(2\pi e)}$ is the radius of a ball of volume 1 in $\mathbb{R}^k$. We use the volume 1 ball here to make the correspondance to $\mathcal{V}_L$ which also has volume 1. Thus, to match the collision probabilities of the ball, which we know yield the right exponent, one would like $\mathcal{V}_L$ to "look like" a ball. Unfortunately, even seemingly strong notions of sphericality, such as assuming that $\mathcal{V}_L$ is within a factor 2 scaling of a ball (which random lattices in fact satisfy, see [16] for an exposition), do not seem to suffice to estimate these delicate collision probabilities at the right ranges. Note that to make the effects of the inevitable estimation errors and dimensionality effects small in the minimization of (2), we will want both p_Δ and $p_{c\Delta}$ to be quite small when we estimate the ratio of their logarithms. For the ball, the function q_Δ has the form $e^{-\alpha\Delta^2}$, where $\alpha := \alpha(\Delta)$ varies slowly within a constant range for $\Delta = O(\sqrt{k})$. Note that if α were in fact constant, then $\ln(1/q_\Delta)/\ln(1/q_{c\Delta})$ would equal $1/c^2$ for every Δ. The region where α is the most stable turns out to be around $\Delta = k^{1/4}$, where q_Δ is quite small, i.e. around $e^{-\Omega(\sqrt{k})}$.

Fortunately, while computing precise estimates for a fixed L is hard, computing the average collision probability over the Siegel measure on the space of lattices of determinant 1 is much easier. Note that the expected collision probability curve $\mathbf{E}_L[p_\Delta]$, where L is chosen from the Siegel measure, corresponds exactly to the collision probability curve associated with a slight modification of the LSH family examined above, namely, where instead of using a fixed lattice, we simply sample a new lattice L from the Siegel measure for each hash function instantiation. We now argue that to show existence of a good LSH lattice one can simply replace the collision probability curve above p_Δ by the expected collision probability curve $\mathbf{E}_L[p_\Delta]$. To see this, assume that (2) holds for the expected curve. By rearranging, this implies that that there exists $\Delta > 0$ such that $\mathbf{E}_L[p_\Delta]^{c^2-o(1)} \geq \mathbf{E}_L[p_{c\Delta}]$. Since $c^2 - o(1) \geq 1$, by Jensen's inequality

$$\mathbf{E}_L[p_\Delta^{c^2-o(1)}] \geq \mathbf{E}_L[p_\Delta]^{c^2-o(1)} \geq \mathbf{E}_L[p_{c\Delta}] \ . \tag{3}$$

Thus, by the probabilistic method, there must exist a lattice L' such that $p_\Delta^{c^2-o(1)} \geq p_{c\Delta}$ holds for L', which shows that L' achieves an LSH constant of $1/c^2 + o(1)$, as needed.

We now explain how one can compute the expected collision probabilities using the estimates of Schmidt and Rogers. For a fixed Δ, a direct computation reveals

$$\begin{aligned}
\mathbf{E}_L[p_\Delta] &= \mathbf{E}_{L, u \leftarrow \mathcal{V}_L, g \leftarrow N(0, I_k/k)}[u + \Delta g \in \mathcal{V}_L] \\
&= \mathbf{E}_{L, g \leftarrow N(0, I_k/k)}\left[\int_{\mathbb{R}^n} I[u \in \mathcal{V}_L, u + \Delta g \in \mathcal{V}_L]\mathrm{du}\right] \quad (\text{ since } \mathcal{V}_L \text{ has volume 1 }) \\
&= \int_{\mathbb{R}^n} \Pr_{L, g \leftarrow N(0, I_k/k)}[u \in \mathcal{V}_L, u + \Delta g \in \mathcal{V}_L]\mathrm{du} \ .
\end{aligned} \tag{4}$$

Define B_x for $x \in \mathbb{R}^k$ to be the open ball around x of radius $\|x\|$. Note that for a fixed g and u, the event that both u and $\Delta g + u$ are in $\mathcal{V}_L$, can be directly expressed as $(B_u \cup B_{\Delta g+u}) \cap L = \emptyset$, i.e. that there is no lattice point closer to u and $\Delta g + u$ than 0. Thus, one can express (4) as

$$\int_{\mathbb{R}^n} \Pr_{L, g \leftarrow N(0, I_k/k)}[(B_u \cup B_{\Delta g+u}) \cap L = \emptyset]\mathrm{du} \ . \tag{5}$$

From here, for fixed g and u, the inner expression is exactly the probability that a random lattice L doesn't intersect a Borel set and hence we may apply Schmidt's estimates. Here Schmidt shows that as long as the $B_u \cup B_{\Delta g+u}$ has volume less than $k-1$, then under a mild technical assumption, we can estimate

$$\Pr_L[(B_u \cup B_{\Delta g+u}) \cap L = \emptyset] \approx e^{-V_{u,\Delta g}}$$

where $V_{u,\Delta g}$ is the volume of $B_u \cup B_{\Delta g+u}$. This estimate is only useful when u has norm roughly r_k, since otherwise the volume of B_u is too large to usefully apply Schmidt's estimate. However, one would expect that for large u, the probability that u is in the Voronoi cell is already quite small. This is formalized by Roger's estimate, which gives that the fraction of cosets of L that are not covered by the ball of volume k around the origin (i.e. again radius roughly r_k) is approximately e^{-k}. In particular, this implies that at most an e^{-k} expected fraction of the Voronoi cell (since points in the Voronoi cell are in one to one correspondance with cosets) lies outside a ball of radius $\approx r_k$, and hence we can truncate the integral expression (5) at roughly this radius without losing much.

After these reductions, we get that the collision probabilities can be tightly approximated by the following explicit integral:

$$\int_{\mathbb{R}^n} \mathbf{E}_g[e^{-V_{u,\Delta g}}]\mathrm{du}. \tag{6}$$

The proof now continues with an unfortunately very long and tedious calculation, which shows that the above estimate closely matches the corresponding collision probability q_Δ for the ball, thus yielding the desired LSH constant.

1.2 Related Work

As mentioned earlier, the works [6, 8] show how to use a data dependent version of LSH to give an improved ANN exponent of $1/(2c^2-1)$, which was shown to be optimal under an appropriate formalization of data dependence in [9]. These works reduce ANN in ℓ_2 to ANN on the sphere via a recursive clustering approach, where the base case of the recursion roughly corresponds to the clustered vectors being embedded as nearly orthogonal vectors on the sphere. A generic reduction from ℓ_2 ANN to spherical ANN (without the exact base case guarantee as above) was also given in earlier work of Valiant [32]. We note that the above clustering style reductions to the sphere remain relatively impractical, and thus there still seems to be room for more direct and practical ℓ_2 methods. For a different vein, the works [10, 12, 7] studied the achievable tradeoffs between query time and space usage, where the optimal tradeoff for hashing based approaches was achieved in [7].

With respect to structured and practical LSH hash functions, [5] computed the collision probabilities for cross-polytope LSH on the sphere (first introduced by [31, 17]), which corresponds to a Voronoi partition on the sphere induced by a vertices of a randomly rotated cross-polytope. As their main result, they show that when near vs far corresponds to ℓ_2 distance $\sqrt{2}/c$ vs $\sqrt{2}$ (the latter case correspondings to orthogonal vectors), cross polytope LSH achieves the optimal limiting exponent of $1/(2c^2-1)$, corresponding to the base case of the recursive clustering approaches above. Furthermore, they show a fine grained lower bound on the LSH exponent (when the far case again corresponds to orthogonal vectors) of any hash function which partitions the sphere into at most T parts[1], which allows them to

[1] Under the mild technical assumption that each piece covers at most 1/2 the sphere.

conclude that any spherical LSH function that substantially improves upon cross polytope LSH needs to have query time *sublinear* in the number of parts. It is tempting here to seek an analogy with lattice based LSH, in that the complexity of CVP computations on a d-dimensional lattice L, after appropriate preprocessing, can be bounded by $\tilde{O}(d^{O(1)}|\mathcal{V}_L|)$ [13] where $|\mathcal{V}_L|$ denotes the number of facets of the Voronoi cell of L. Thus, one may wonder if $|\mathcal{V}_L|$ can be associated with the number of "parts" in an analogous manner. For a generic d-dimensional lattice, we note that $|\mathcal{V}_L| = 2(2^d-1)$, and thus the corresponding question would be to find an LSH-good lattice for which CVP takes $\tilde{O}(2^{(1-\epsilon)d})$ for some positive $\epsilon > 0$. As another interesting comparison, the d-dimensional cross polytope induces a partion with $2d$ parts whose gap to optimality (in terms of the spherical LSH exponent) is $O(\log\log d/\log d)$, whereas a random d-dimensional lattice has a Voronoi cell is $2(2^d-1)$ facets with a gap to optimality (for ℓ_2 LSH) of $O(1/\sqrt{d})$.

1.3 Conclusions and Open Problems

To summarize, for a fixed approximation factor $c > 1$, we show that random space partitions induced by *shifts of a single lattice* can achieve the optimal *data oblivious* LSH exponent for the ℓ_2 metric. While this shows that we can hope for "well-structured" space partitions for ℓ_2, the lattices we use to show existence are *random*, and are in many ways devoid of easy to exploit structure (at least algorithmically). Thus, a natural open question is whether one can find a more structured family of lattices achieving the same limiting LSH exponent for which CVP queries can be executed faster. In terms of improving the present result, another natural question would be to make our proof constructive and to show that for a fixed dimension k, there exists a single k-dimensional lattice which achieves the optimal LSH exponent for every $c \geq 1$.

Organization. In Section 2, we setup notations and define formally the notion of lattices and approximate nearest neighbor search problem. We describe our lattice based hash function family in Section 3 and analyze its performance. The helper theorems needed to show the main result are proved in subsequent sections.

2 Preliminaries

We denote the set $\{1, 2, \ldots, n\}$ by $[n]$. We work over the Euclidean space. For $x \in \mathbb{R}^d$, let $||x|| = \sqrt{\sum_i x_i^2}$ denote the ℓ_2 norm of x. Let V_B denote the volume of a k-dimensional unit-radius ball. Let $\tau = \sqrt{k} \cdot V_B^{\frac{1}{k}}$. By standard geometry facts, $\tau = \sqrt{2\pi e}\left(1 + O\left(\frac{1}{k}\right)\right)$. For $x \in \mathbb{R}^k$, let B_x denote the open ball centered at x of radius $\|x\|$ and let V_x denote its volume. Note that $V_x = V_B\|x\|^k$.

Lattices. A lattice $L \subset \mathbb{R}^d$ is the set of all linear combinations with integer coefficients of a set of linearly independent vectors $\{b_1, b_2, \ldots, b_r\}$, i.e., $L = \{\sum_i \alpha_i b_i \mid \alpha_i \in \mathbb{Z}\ \forall i \in [r]\}$. The lattice may be represented by the $d \times r$ basis matrix B, whose columns are the vectors b_i. If the *rank* r is exactly equal to d, then the lattice is said to have *full rank*. It is common to assume that the lattice has full rank, and we do so in what follows, since otherwise one may just work over the real span of B.

The *quotient group* $\mathbb{R}^d/L$ of L is the set of cosets $c + L = \{c + v \mid v \in L\}$, where $c \in \mathbb{R}^d$, with the group operation $(c_1 + L) + (c_2 + L) = (c_1 + c_2) + L$. The *determinant* of L, denoted $\det(L)$, is defined as $\det(L) = \sqrt{B^T B}$. A lattice has multiple bases: if B is a basis then

BU is also a basis, for any unimodular matrix U (i.e., a matrix U with integer entries with $\det(U) = 1$.) The *Voronoi cell* of a lattice is the set of all points closer to the origin than to any other lattice point. Formally, $\mathcal{V}_L := \{x \in \mathbb{R}^d \mid ||x|| \leq ||x-v||, \forall v \in L\}$. Define the *shifted* Voronoi cell centered at v, denoted $\mathcal{V}_L(v)$, to be the set of points $v + \mathcal{V}_L = \{v + u \mid u \in \mathcal{V}_L\}$. It is a standard fact that the set of cells $\{v + \mathcal{V}_L\}_{v \in L}$ cover the entire space $\mathbb{R}^d$. Moreover, for every $x \in \mathbb{R}^d$, there exists a $v \in L$ such that $x - v \in \mathcal{V}_L$. In fact, the (half-open) Voronoi cell contains exactly one representative from each coset $c + L$, for $c \in \mathbb{R}^d$. One of the fundamental computational problem on lattices is the *Closest Vector Problem* (CVP) defined as follows: given a target vector $t \in \mathbb{R}^d$, find a closest vector from the lattice L to t. We will denote a solution to CVP with input t by $CV_L(t)$. We will use recent algorithms running in time $O(2^d)$ as a blackbox [1]. We will need the following property of the Voronoi cell.

▶ **Fact 1.** *$v \in CV_L(t)$ if and only if $t - v \in \mathcal{V}_L$.*

Approximate Near Neighbor and LSH. In the c-approximate near neighbor(c-ANN) problem, given a collection $\mathcal{P}$ of n points in $\mathbb{R}^d$, and parameters $r, \delta > 0$, the goal is to construct a data structure with the following property: on input a query point $q \in \mathbb{R}^d$, with probability $1 - \delta$, if there exists $p \in \mathcal{P}$ with $||q - p|| \leq r$, it outputs some point $p' \in \mathcal{P}$, with $||q - p'|| \leq c \cdot r$. By a simple scaling of the coordinates, one may assume that $r = 1$. Also, δ is assumed to be a constant, and the success probability can be amplified by building several instances of the data structure.

A family $\mathcal{H}$ is a *locality-sensitive hashing* scheme with parameters $(1, c, p_1, p_2)$ if it satisfies the following properties: for any $p, q \in \mathbb{R}^d$

- if $||p - q|| \leq 1$ then $\Pr_{\mathcal{H}}[h(q) = h(p)] \geq p_1$,
- if $||p - q|| \geq c$ then $\Pr_{\mathcal{H}}[h(q) = h(p)] \leq p_2$.

The initial work of [20] shows that an LSH scheme implies a data structure for c-ANN.

▶ **Theorem 2.** *[20] Given a LSH family $\mathcal{H}$ with parameters $(1, c, p_1, p_2)$, where each function in $\mathcal{H}$ can be evaluated in time τ, let $\rho = \frac{\log(1/p_1)}{\log(1/p_2)}$. Then there exists a data structure for c-ANN with $O((d + \tau)n^{\rho} \log_{1/p_2} n)$ query time, using $O(dn + n^{1+\rho})$ amount of space.*

Multidimensional Gaussian. A d-dimensional Gaussian distribution with mean 0 and covariance matrix $\sigma^2 I_d \in \mathbb{R}^{d \times d}$ has density function

$$p(x) = \frac{1}{(2\pi)^{d/2}\sigma^d} \exp(-\frac{||x||^2}{2\sigma^2}),$$

and is denoted by $N(0, \sigma^2 I_d)$.

3 Our Lattice-based Hash Family and Proof Strategy

LSH family for lattice L with $\det(L) = 1$. A hash function $h = h_{M,t}$ indexed by a projection matrix $M \in \mathbb{R}^{k \times d}$ from $\mathbb{R}^d$ to $\mathbb{R}^k$, and a vector $t \in \mathbb{R}^k$ is constructed as follows:

1. pick the entries $M_{i,j}$ according to a Gaussian distribution with mean 0 and variance $1/k$.
2. pick t uniformly from the Voronoi cell $\mathcal{V}_L$ of L (centered at 0). Sampling t can be achieved by sampling from $\mathbb{R}^k/L$, namely by sampling from the fundamental parallelepiped with respect to any basis.

Given a point $a \in \mathbb{R}^d$, we define $h(a)$ to be a closest vector in L to its projection Ma translated by t. Formally,

$$h(a) = CV_L(Ma + t).$$

We first show that for $a, b \in \mathbb{R}^d$ the quantity $\Pr_{M,t}[h(a) = h(b)]$ only depends on the distance $||a - b||$, and not on the points a, b themselves.

▶ **Proposition 3.** *Let $a, b \in \mathbb{R}^d$ be arbitrary and let $\Delta = ||a - b||$. Then*

$$\Pr_{M,t}[h(a) = h(b)] = \Pr_{x \leftarrow \mathcal{V}_L, y \leftarrow N(0, \Delta^2 I_k/k)}[x + y \in \mathcal{V}_L].$$

Let p_Δ denote the probability of collision of two inputs which are exactly distance Δ apart. i.e., $p_\Delta := \Pr_{M,t}[h(a) = h(b)]$, where $||a - b|| = \Delta$. An easy argument shows that p_Δ is non-increasing as a function of Δ.

▶ **Corollary 4.** *p_Δ is non-increasing as a function of Δ.*

The performance of our LSH family is measured by the LSH constant defined by

$$\rho_L := \min_{\Delta > 0} \frac{\ln 1/p_\Delta}{\ln 1/p_{c\Delta}}.$$

Our result shows the existence of a lattice L with optimal performance guarantee.

▶ **Theorem 5.** *For every k large enough and $c > 1$, there exists a k-dimensional lattice L with $\det(L) = 1$ achieving*

$$\rho_L \leq \frac{1}{c^2} + O\left(\frac{1}{\sqrt{k}}\right).$$

Theorem 5 follows from our main technical result, which bounds the expected collision probabilities p_Δ and $p_{c\Delta}$ for $\Delta = k^{1/4}$.

▶ **Theorem 6.** *For every k large enough and $c > 1$, there exist absolute constants K_1, K_2, K_3 such that for $\Delta = k^{1/4}$,*

$$\mathbf{E}_L\left[p_\Delta\right] \geq K_1\, e^{-\frac{\tau^2}{8}\sqrt{k}} \qquad \text{and,}$$
$$\mathbf{E}_L\left[p_{c\Delta}\right] \leq K_2\, e^{-\frac{\tau^2}{8}c^2\sqrt{k}\left(1 - \frac{K_3 c^2}{\sqrt{k}}\right)},$$

where the expectation is over k-dimensional lattices L with $\det(L) = 1$.

We can now prove Theorem 5 using Corollary 4 and Theorem 6.

Proof of Theorem 5. For any $\Delta > 1$, define $\tilde{\rho} := \frac{\ln 1/\mathbf{E}_L[p_\Delta]}{\ln 1/\mathbf{E}_L[p_{c\Delta}]}$. From Corollary 4, we know that p_Δ is non-increasing. Hence, $\tilde{\rho} \leq 1$ for any $c > 1$. So, we can use Jensen's inequality to get that

$$\begin{aligned}\mathbf{E}_L\left[p_\Delta^{1/\tilde{\rho}}\right] &\geq \mathbf{E}_L\left[p_\Delta\right]^{1/\tilde{\rho}} && \text{(Jensen's inequality)}\\ &= \mathbf{E}_L\left[p_{c\Delta}\right] && \text{(by the definition of } \tilde{\rho}).\end{aligned}$$

By the probabilistic method, it then follows that there exists a k-dimensional lattice L with $\det(L) = 1$, such that the collision probabilities satisfiy $\frac{\ln 1/p_\Delta}{\ln 1/p_{c\Delta}} = \tilde{\rho}$ and hence, $\rho_L \leq \tilde{\rho}$.

We now show that $\tilde{\rho} \leq \frac{1}{c^2} + O\left(\frac{1}{\sqrt{k}}\right)$. From Theorem 6 we know that for any $c > 1$, and $\Delta = k^{\frac{1}{4}}$, there exist constants K_1, K_2, K_3 such that

$$\mathbf{E}_L[p_\Delta] \geq K_1\, e^{-\frac{\tau^2}{8}\sqrt{k}} \qquad \text{and,}$$
$$\mathbf{E}_L[p_{c\Delta}] \leq K_2\, e^{-\frac{\tau^2}{8}c^2\sqrt{k}\left(1-\frac{K_3c^2}{\sqrt{k}}\right)}$$

Note that for $c > \frac{k^{\frac{1}{4}}}{\sqrt{K_3}}$, the upper bound on $\mathbf{E}_L[p_{c\Delta}]$ from Theorem 6 becomes trivial. First, we consider the case when $c \leq \frac{k^{\frac{1}{4}}}{2\sqrt{K_3}}$. For this value of c, we can use bounds obtained in Theorem 6 to show that $\tilde{\rho} \leq \frac{1}{c^2} + O\left(\frac{1}{\sqrt{k}}\right)$ as follows:

$$\begin{aligned}\frac{\ln 1/\mathbf{E}_L(p_\Delta)}{\ln 1/\mathbf{E}_L(p_{c\Delta})} &\leq \frac{\frac{\tau^2}{8}\sqrt{k} - \ln K_1}{\frac{\tau^2}{8}c^2\sqrt{k}\left(1-\frac{K_3c^2}{\sqrt{k}}\right) - \ln K_2} \\ &\leq \frac{1}{c^2}\left(1 + K_4\frac{c^2}{\sqrt{k}}\right) \qquad \text{for some constant } K_4\ .\end{aligned}$$

Now, for $c > \frac{k^{\frac{1}{4}}}{2\sqrt{K_3}}$, we need to show that there exists a k-dimensional lattice of determinant 1, such that $\rho_L \leq \frac{1}{c^2} + O\left(\frac{1}{\sqrt{k}}\right)$. From the monotonicity of p_Δ, we know that for any $c' < c$, $p_{c\Delta} \leq p_{c'\Delta}$. Therefore, consider $c' = k^{\frac{1}{4}}/2\sqrt{K_3} < c$. From Theorem 6, and the analysis above, we know that there exists a lattice of determinant 1 such that

$$\begin{aligned}\rho_L &\leq \frac{1}{c'^2}\left(1 + K_4\frac{c'^2}{\sqrt{k}}\right) \qquad \text{for some constant } K_4 \\ &= \frac{2K_3}{\sqrt{k}} + \frac{K_4}{\sqrt{k}} \\ &= \frac{1}{c^2} + O\left(\frac{1}{\sqrt{k}}\right).\end{aligned}$$

◀

Proving Theorem 6 poses substantial technical hurdles. We will break the proof into smaller components, which we describe after introducing some helpful notation.

For any $\Delta \geq 1$, define

$$I(\Delta^2) := \int_{x\in\mathbb{R}^k : V_x \leq \frac{k}{8}} \mathbf{E}_{y\leftarrow N(0,\Delta^2 I_k/k)}\left[e^{-V_x - V_{x+y}}\right]\, dx.$$

In the next lemma, we show tight bounds on $\mathbf{E}_L[p_\Delta]$ in terms of $I(\Delta^2)$.

▶ **Lemma 7.** *For every k large enough and any $\Delta \geq 1$,*

$$I(\Delta^2) - e^{-k/8} \;\leq\; \mathbf{E}_L[p_\Delta] \;\leq\; 4I(4^{-\frac{2}{k}}\Delta^2) + 3e^{-k/8}.$$

where the expectation is over k-dimensional lattices L with $\det(L) = 1$.

We now show tight bounds for $I(\Delta^2)$ for $\Delta^2 = \beta\sqrt{k}$,where $1 \leq \beta \leq O(\sqrt{k})$ in Lemma 8, which is the most technically delicate part of the analysis, as it involves precise balancing of parameters and taking care of minutious details.

▶ **Lemma 8.** *There exist absolute constants $K \in [0,1], K_1, K_2, \bar{K}_1, \bar{K}_2$ such that for any $1 \leq \beta \leq K\sqrt{k}$,*

$$\bar{K}_1\, e^{-\alpha\beta\sqrt{k}\left(1+\frac{\bar{K}_2\beta}{\sqrt{k}}\right)} \;\leq\; I\left(\beta\sqrt{k}\right) \;\leq\; K_1\, e^{-\alpha\beta\sqrt{k}\left(1-\frac{K_2\beta}{\sqrt{k}}\right)}$$

We now show how Lemmas 7 and Lemma 8 imply Theorem 6.

Proof of Theorem 6. First we prove the lower bound on $\mathbf{E}_L[p_\Delta]$ for $\Delta = k^{\frac{1}{4}}$. From Lemma 7 and Lemma 8, we have

$$\begin{aligned}
\mathbf{E}_L[p_\Delta] &\geq I(\Delta^2) - e^{-k/8} \qquad \text{(from Lemma 7)} \\
&\geq \bar{K}_1\ e^{-\alpha\sqrt{k}\left(1+\frac{\bar{K}_2}{\sqrt{k}}\right)} - e^{-k/8} \qquad \text{(from Lemma 8 with } \beta = 1) \\
&\geq \bar{K}_3\ e^{-\alpha\sqrt{k}}.
\end{aligned}$$

Similarly, for the upper bound on $\mathbf{E}_L[p_{c\Delta}]$ for $\Delta = k^{\frac{1}{4}}$, we get

$$\begin{aligned}
\mathbf{E}_L[p_{c\Delta}] &\leq 4\ I(4^{-\frac{2}{k}}c^2\Delta^2) + 3e^{-k/8} \qquad \text{(from Lemma 7)} \\
&\leq K_1\ e^{-4^{-\frac{2}{k}}c^2\alpha\sqrt{k}\left(1-\frac{K_2c^2}{\sqrt{k}}\right)} + 3e^{-k/8} \qquad \text{(from Lemma 8 with } \beta = 4^{-\frac{2}{k}}c^2) \\
&\leq K_3\ e^{-c^2\alpha\sqrt{k}\left(1-\frac{K_2c^2}{\sqrt{k}}\right)} \qquad \text{(since } 4^{-\frac{2}{k}} \geq 1 - O(1/k)) \ .
\end{aligned}$$

Note that since Lemma 8 holds for $\beta < O(\sqrt{k})$, the upper bound on $\mathbf{E}_L[p_{c\Delta}]$ holds for $c^2 \leq K\sqrt{k}$ for some constant K. ◀

We conclude this section with the proof of Proposition 3 and of Corollary 4, while devoting the rest of the paper for the proof of Lemma 7. Due to space constraints, the proof of Lemma 8 will appear in the full version of the paper.

Proof of Proposition 3. Let M and t be as defined above. From the definition of the hash function, $h(a) = h(b)$ if $Ma + t$ and $Mb + t$ land in the same Voronoi cell of L about some lattice point. Let $||a - b|| = \Delta$. We have

$$\begin{aligned}
p_\Delta &= \Pr_{M,t}[h(a) = h(b)] \\
&= \Pr_{M,t}[CV_L(Ma + t) = CV_L(Mb + t)] \\
&= \Pr_{M,t}[Ma + t, Ma + M(b - a) + t \text{ lie in the same Voronoi cell }]. \qquad (7)
\end{aligned}$$

Let $Ma + t \in \mathcal{V}_L(\ell)$ for some $\ell \in L$. Define $x := Ma + t - \ell \in \mathcal{V}_L$. Note that because of the random shift t, x is a uniform random point in the Voronoi cell of L about 0.

Let $y := M(b - a) \in \mathbb{R}^k$. Since each entry M_{ij} of M is a Gaussian random variable with 0 mean and variance $1/k$, therefore, the i^{th} entry of y, given as $y_i = \sum_{j=1}^{k} M_{ij}(b_j - a_j)$ has mean 0 and variance $\frac{1}{k}\sum_j (b_j - a_j)^2 = \frac{\Delta^2}{k}$.

Plugging these observations in Equation 7, we get

$$\begin{aligned}
p_\Delta &= \Pr_{M,t}[Ma + t - \ell, Ma + M(b - a) + t - \ell \in \mathcal{V}_L] \\
&= \Pr_{x \leftarrow \mathcal{V}_L, y \leftarrow N(0, \Delta^2 I_k/k)}[x, x + y \in \mathcal{V}_L] \\
&= \Pr_{x \leftarrow \mathcal{V}_L, y \leftarrow N(0, \Delta^2 I_k/k)}[x + y \in \mathcal{V}_L].
\end{aligned}$$ ◀

Proof of Corollary 4. By Proposition 3, it suffices to show that the function

$$f(s) = \Pr_{x \leftarrow \mathcal{V}_L, y \leftarrow N(0, I_k/k)}[x + sy \in \mathcal{V}_L],$$

where x is uniform in $\mathcal{V}_L$ and y is standard Gaussian, is a non-increasing function of s on $\mathbb{R}_+$. Since $\mathcal{V}_L$ has volume 1 and $x + sy \in \mathcal{V}_L \Leftrightarrow x \in \mathcal{V}_L - sy$, we have that

$$\Pr_{x,y}[x + sy \in \mathcal{V}_L] = \Pr_y[\mathrm{vol}(\mathcal{V}_L \cap (\mathcal{V}_L - sy)] \ .$$

Define $g_y(s) := \mathrm{vol}(\mathcal{V}_L \cap (\mathcal{V}_L - sy))$. We claim that $g_y(s)$ is non-decreasing on $(-\infty, 0]$ and non-increasing on $[0, \infty)$. To see this, note that by symmetry of $\mathcal{V}$, g_y is symmetric, i.e. $g_y(s) = g_y(-s)$. Furthermore, for $\lambda \in [0, 1]$, $s_1, s_2 \in \mathbb{R}$,

$$\begin{aligned}
g_y(\lambda s_1 + (1-\lambda)s_2)^{1/n} &= \mathrm{vol}(\mathcal{V}_L \cap (\mathcal{V}_L - \lambda(s_1 + (1-\lambda)s_2)y))^{1/n} \\
&\geq \mathrm{vol}(\lambda(\mathcal{V}_L \cap (\mathcal{V}_L - s_1 y)) + (1-\lambda)(\mathcal{V}_L \cap (\mathcal{V}_L - s_2 y)))^{1/n} \\
&\qquad (\text{ by containment }) \\
&\geq \lambda \mathrm{vol}(\mathcal{V}_L \cap (\mathcal{V}_L - s_1 y))^{1/n} + (1-\lambda)\mathrm{vol}(\mathcal{V}_L \cap (\mathcal{V}_L - s_2 y))^{1/n} \\
&\qquad (\text{ by Brunn-Minkowski }) \\
&= \lambda g_y(s_1)^{1/n} + (1-\lambda) g_y(s_2)^{1/n} \ .
\end{aligned}$$

Therefore, $g_y(s)^{1/n}$ is a symmetric, non-negative and concave function of s. Any symmetric concave function on $\mathbb{R}$ must attain its maximum value at 0, and hence must be non-increasing away from 0.

Now consider $0 \leq s_1 \leq s_2$. Since g_y is non-increasing on $\mathbb{R}_+$, we get that

$$f(s_1) = \mathbf{E}_y[g_y(s_1)] \geq \mathbf{E}_y[g_y(s_2)] = f(s_2)$$

as needed. ◀

4 Proof of Lemma 7

In the previous section, we had seen that the expected collision probability between points which are Δ apart is defined as

$$\begin{aligned}
\mathbf{E}_L[p_\Delta] &= \mathbf{E}_L\left[\Pr_{x \leftarrow \mathcal{V}_L, y \leftarrow N(0, \Delta^2 I_k/k)}[x + y \in \mathcal{V}_L]\right] \\
&= \int_{x \in \mathbb{R}^k} \int_{y \in \mathbb{R}^k} \Pr_L(x, x + y \in \mathcal{V}_L) \cdot \frac{e^{-\frac{\|y\|^2}{2\sigma^2}}}{(2\pi\sigma^2)^{\frac{k}{2}}} \, dy \, dx \qquad \text{for } \sigma^2 = \Delta^2/k.
\end{aligned}$$

The goal of this section is to derive tight bounds for this expression through the proof of Lemma 7.

Recall that B_x denotes the open k-dimensional ball centered at $x \in \mathbb{R}^k$ of radius $\|x\|$ and B_{x+y} denotes the open k-dimensional ball centered at $x + y \in \mathbb{R}^k$ of radius $\|x + y\|$. Also, V_x and V_{x+y} denotes their volumes. Consider $B_{x,y} = B_x \cup B_{x+y}$, the union of B_x and B_{x+y} and let $V_{x,y}$ denote its volume. We will need the following theorem for the proof of Lemma 7.

▶ **Lemma 9.**

$$e^{-V_{x,y}} - e^{-k/4} \ \leq \ \Pr_L(x, x + y \in \mathcal{V}_L) \ \leq \ e^{-\frac{1}{2}V_{x,y}} + e^{-k/4}.$$

In order to prove Lemmas 7 and 9, we invoke the following results of Rogers [27] and Schmidt [28].

▶ **Theorem 10** (Corollary of [27], Theorem 1). *Let B be the k-dimensional ball of volume V centered at the origin. If $V \leq \frac{k}{8}$, then there exists a constant k_0 such that for $k > k_0$,*

$$\left| \int_{x \in \mathbb{R}^k} \Pr_L [x \in \mathcal{V}_L \setminus B] \; dx - e^{-V} \right| \leq c_1 k^3 \left(\frac{16}{27}\right)^{\frac{k}{4}}$$

where, the probability is taken over the space of all lattices of determinant 1.

▶ **Theorem 11** ([28], Theorem 4). *Let S be a Borel set of measure V such that $0 \notin S$ and for all $x \in S$, $-x \notin S$. If $V \leq k-1$, then for $k \geq 13$,*

$$\Pr_L [L \cap S = \emptyset] = e^{-V} (1 - R).$$

where, the probability is taken over the space of all lattices of determinant 1 *and* $|R| < 6\left(\frac{3}{4}\right)^{\frac{k}{2}} e^{4V} + V^{k-1} k^{-k+1} e^{V+k}$.

▶ **Fact 12.**

$$\frac{1}{2}(V_x + V_{x+y}) \leq V_{x,y} \leq V_x + V_{x+y}.$$

Proof. Let WLOG, $V_x \leq V_{x+y}$. Also, we know that $V_{x,y} = V_x + V_{x+y} - V(B_x \cap B_{x+y})$. We now show that $V(B_x \cap B_{x+y}) \leq \frac{1}{2}(V_x + V_{x+y})$. This fact follows easily from the observation that the intersection volume is at most the volume of the smaller ball. Therefore,

$$V(B_x \cap B_{x+y}) \leq V_x = \frac{1}{2}V_x + \frac{1}{2}V_x \leq \frac{1}{2}(V_x + V_{x+y}). \qquad \blacktriangleleft$$

We now prove Lemma 7 using Lemma 9.

Proof of Lemma 7 . For notational convenience, we will use σ^2 to denote Δ^2/k. From the definition of p_Δ and Proposition 3, we have

$$\begin{aligned}
\mathbf{E}_L[p_\Delta] &= \mathbf{E}_L \left[\Pr_{x \leftarrow \mathcal{V}_L, y \leftarrow N(0,\sigma^2 I_k)} [x + y \in \mathcal{V}_L] \right] \\
&= \int_{x \in \mathbb{R}^k} \int_{y \in \mathbb{R}^k} \Pr_L(x, x+y \in \mathcal{V}_L) \cdot \frac{e^{-\frac{\|y\|^2}{2\sigma^2}}}{(2\pi\sigma^2)^{\frac{k}{2}}} \; dy \; dx \\
&= \int_{x \in \mathbb{R}^k : V_x \leq \frac{k}{8}} \int_{y \in \mathbb{R}^k} \Pr_L(x, x+y \in \mathcal{V}_L) \cdot \frac{e^{-\frac{\|y\|^2}{2\sigma^2}}}{(2\pi\sigma^2)^{\frac{k}{2}}} \; dy \; dx \\
&\qquad + \int_{x \in \mathbb{R}^k : V_x > \frac{k}{8}} \int_{y \in \mathbb{R}^k} \Pr_L(x, x+y \in \mathcal{V}_L) \cdot \frac{e^{-\frac{\|y\|^2}{2\sigma^2}}}{(2\pi\sigma^2)^{\frac{k}{2}}} \; dy \; dx. \qquad (8)
\end{aligned}$$

We first note that if $V_x \geq \frac{k}{8}$, then the probability that $x \in \mathcal{V}_L$ is itself very small. This fact gives us tight bounds on $\mathbf{E}_L[p_\Delta]$ up to additive $e^{-\Omega(k)}$ term. We use Theorem 10 to

formalize this statement. Let B_0 be the 0 centered ball of volume $\frac{k}{8}$. We have,

$$\begin{aligned}
&\int_{x\in\mathbb{R}^k:V_x\geq\frac{k}{8}}\int_{y\in\mathbb{R}^k}\Pr_L(x,x+y\in\mathcal{V}_L)\cdot\frac{e^{-\frac{\|y\|^2}{2\sigma^2}}}{(2\pi\sigma^2)^{\frac{k}{2}}}\,dy\,dx\\
&\qquad\leq\int_{x\in\mathbb{R}^k:V_x\geq\frac{k}{8}}\int_{y\in\mathbb{R}^k}\Pr_L(x\in\mathcal{V}_L)\cdot\frac{e^{-\frac{\|y\|^2}{2\sigma^2}}}{(2\pi\sigma^2)^{\frac{k}{2}}}\,dy\,dx\\
&\qquad=\int_{x\in\mathbb{R}^k:V_x\geq\frac{k}{8}}\Pr_L(x\in\mathcal{V}_L)dx\\
&\qquad=\int_{x\in\mathbb{R}^k}\Pr_L(x\in\mathcal{V}_L\setminus B_0)\,dx\\
&\qquad=e^{-\frac{k}{8}}+e^{-\frac{k}{8}}.\qquad(\text{ from Theorem 10 })
\end{aligned}$$

Plugging this observation into the expression for $\mathbf{E}_L[p_\Delta]$ in Equation 8, we get that

$$\begin{aligned}
&\int_{x\in\mathbb{R}^k:V_x\leq\frac{k}{8}}\int_{y\in\mathbb{R}^k}\Pr_L(x,x+y\in\mathcal{V}_L)\cdot\frac{e^{-\frac{\|y\|^2}{2\sigma^2}}}{(2\pi\sigma^2)^{\frac{k}{2}}}\,dy\,dx\\
&\qquad\leq\mathbf{E}_L[p_\Delta]\\
&\qquad\leq\int_{x\in\mathbb{R}^k:V_x\leq\frac{k}{8}}\int_{y\in\mathbb{R}^k}\Pr_L(x,x+y\in\mathcal{V}_L)\cdot\frac{e^{-\frac{\|y\|^2}{2\sigma^2}}}{(2\pi\sigma^2)^{\frac{k}{2}}}\,dy\,dx+2e^{-k/8}.
\end{aligned}$$

Further, using the bounds on $\Pr_L(x,x+y\in\mathcal{V}_L)$ from Lemma 9, we get

$$\begin{aligned}
&\int_{x\in\mathbb{R}^k:V_x\leq\frac{k}{8}}\int_{y\in\mathbb{R}^k}\left(e^{-V_{x,y}}-e^{-k/4}\right)\cdot\frac{e^{-\frac{\|y\|^2}{2\sigma^2}}}{(2\pi\sigma^2)^{\frac{k}{2}}}\,dy\,dx\\
&\qquad\leq\mathbf{E}_L[p_\Delta]\\
&\leq\int_{x\in\mathbb{R}^k:V_x\leq\frac{k}{8}}\int_{y\in\mathbb{R}^k}\left(e^{-\frac{1}{2}V_{x,y}}+e^{-k/4}\right)\cdot\frac{e^{-\frac{\|y\|^2}{2\sigma^2}}}{(2\pi\sigma^2)^{\frac{k}{2}}}\,dy\,dx+2e^{-k/8}.
\end{aligned}$$

Since $V_{x,y}\leq V_x+V_{x+y}$, the lower bound in the theorem statement then follows trivially.

$$\begin{aligned}
\mathbf{E}_L[p_\Delta]&\geq\int_{x\in\mathbb{R}^k:V_x\leq\frac{k}{8}}\int_{y\in\mathbb{R}^k}\left(e^{-V_{x,y}}-e^{-k/4}\right)\cdot\frac{e^{-\frac{\|y\|^2}{2\sigma^2}}}{(2\pi\sigma^2)^{\frac{k}{2}}}\,dy\,dx\\
&=\int_{\substack{x\in\mathbb{R}^k\\V_x\leq\frac{k}{8}}}\int_{y\in\mathbb{R}^k}e^{-V_{x,y}}\frac{e^{-\frac{\|y\|^2}{2\sigma^2}}}{(2\pi\sigma^2)^{\frac{k}{2}}}\,dy\,dx-\int_{\substack{x\in\mathbb{R}^k\\V_x\leq\frac{k}{8}}}\int_{y\in\mathbb{R}^k}e^{-k/4}\frac{e^{-\frac{\|y\|^2}{2\sigma^2}}}{(2\pi\sigma^2)^{\frac{k}{2}}}\,dy\,dx\\
&\geq\int_{x\in\mathbb{R}^k:V_x\leq\frac{k}{8}}\mathbf{E}_{y\sim N(0,\sigma^2\ I_k)}\left[e^{-V_x-V_{x+y}}\right]\,dx-\frac{k}{8}e^{-k/4}\\
&\geq\int_{x\in\mathbb{R}^k:V_x\leq\frac{k}{8}}\mathbf{E}_{y\sim N(0,\sigma^2\ I_k)}\left[e^{-V_x-V_{x+y}}\right]\,dx-e^{-k/8}
\end{aligned}$$

For the upper bound, set $u=4^{-\frac{1}{k}}x$, and $v=4^{-\frac{1}{k}}y$. Since $\frac{1}{2}V_{x,y}\geq\frac{V_x+V_{x+y}}{4}=V_u+V_{u+v}$,

we have

$$\begin{aligned}
\mathbf{E}_L[p_\Delta] &\le \int_{x\in\mathbb{R}^k : V_x \le \frac{k}{8}} \int_{y\in\mathbb{R}^k} \left(e^{-\frac{1}{2}V_{x,y}} + e^{-k/4}\right) \cdot \frac{e^{-\frac{\|y\|^2}{2\sigma^2}}}{(2\pi\sigma^2)^{\frac{k}{2}}}\, dy\, dx + 2e^{-k/8} \\
&\le \int_{x\in\mathbb{R}^k} \int_{y\in\mathbb{R}^k} e^{-\frac{V_x + V_{x+y}}{4}} \frac{e^{-\frac{\|y\|^2}{2\sigma^2}}}{(2\pi\sigma^2)^{\frac{k}{2}}}\, dy\, dx + 3e^{-k/8} \\
&= \int_{u\in\mathbb{R}^k} \int_{v\in\mathbb{R}^k} e^{-V_u - V_{u+v}} \frac{e^{-\frac{\|v\|^2}{2\sigma^2}\frac{2}{4^k}}}{(2\pi\sigma^2)^{\frac{k}{2}}}\, 4dv\, 4du + 3e^{-k/8} \\
&= 4\int_{u\in\mathbb{R}^k} \int_{v\in\mathbb{R}^k} e^{-V_u - V_{u+v}} \frac{e^{-\frac{\|v\|^2}{2(4^{-\frac{1}{k}}\sigma)^2}}}{\left(2\pi(4^{-\frac{1}{k}}\sigma)^2\right)^{\frac{k}{2}}}\, dv\, du + 3e^{-k/8} \\
&= 4\int_{u\in\mathbb{R}^k} \mathbf{E}_v\left[e^{-V_u - V_{u+v}}\right] du + 3e^{-k/8} \quad \text{where, } v \sim N(0, 4^{-\frac{2}{k}}\sigma^2\, I_k).
\end{aligned}$$

◀

Now it remains to prove Lemma 9.

Proof of Lemma 9. Recall that $B_{x,y} = B_x \cup B_{x+y}$, the union of B_x and B_{x+y} and $V_{x,y}$ denotes its volume. We note that x and $x+y$ are in the voronoi cell of a lattice L if and only if $B_{x,y}$ does not contain any lattice points. Therefore,

$$\Pr_L[x, x+y \in \mathcal{V}_L] = \Pr_L[B_{x,y} \cap L = \emptyset]$$

As a first case, suppose $V_{x,y} < \frac{k}{32}$. Now consider the following partition of $B_{x,y}$. Let S be the set of points $a \in B_{x,y}$ such that $-a \in B_{x,y}$.

$$S = \{a \in B_{x,y} \mid -a \in B_{x,y}\}.$$

Partition S with respect to an arbitrary hyperplane as follows: Define $S_1 = \{a \in S \mid a^t y < 0\}$ and $S_2 = S \setminus S_1$ for an arbitrarily chosen $y \in \mathbb{R}^k$. Note that for every $a \in S_1$, $-a \in S_2$. Define $A = (B_{x,y} \setminus S) \cup S_1$. Note that $\{A, S_2\}$ is a partition of $B_{x,y}$, i.e., $B_{x,y} = A \cup S_2$, and $A \cap S_2 = \emptyset$.

Without loss of generality, assume that A is the larger partition of $B_{x,y}$, i.e $V_A \ge \frac{1}{2}V_{x,y}$. Also from the definition of A and S_2, we have that if $A \cap L = \emptyset$, then $S_2 \cap L = \emptyset$. We can now apply Theorem 11 for both A and S_2.

$$\begin{aligned}
\Pr_L[B_{x,y} \cap L = \emptyset] &= \Pr_L[(A \cap L = \emptyset), (S_2 \cap L = \emptyset)] \\
&= \Pr_L[A \cap L = \emptyset]\ \Pr_L[(S_2 \cap L = \emptyset) \mid (A \cap L = \emptyset)] \\
&= \Pr_L[A \cap L = \emptyset] \\
&= e^{-V_A}(1 - R_A) \quad \text{where, } |R_A| = 6\left(\frac{3}{4}\right)^{\frac{k}{2}} e^{4V_A} + V_A^{k-1}k^{-k+1}e^{V_A+k}.
\end{aligned}$$

Since $\frac{1}{2}V_{x,y} \le V_A \le V_{x,y} < \frac{k}{32}$, we have $|R_A| < e^{-k/4}$. Therefore,

$$e^{-V_{x,y}}\left(1 - e^{-k/4}\right) \le \Pr_L[B_{x,y} \cap L = \emptyset] \le e^{-\frac{1}{2}V_{x,y}}\left(1 + e^{-k/4}\right).$$

Next, suppose $V_{x,y} > \frac{k}{32}$. Then consider a body $B'_{x,y}$ contained in $B_{x,y}$ of volume $\frac{k}{32}$. Using a similar argument as above with $B_{x,y}$ replaced with $B'_{x,y}$, we conclude that

$$\Pr_L[B_{x,y} \cap L = \emptyset] \le \Pr_L\left[B'_{x,y} \cap L = \emptyset\right] \le e^{-k/4}.$$

◀

References

1 Divesh Aggarwal, Daniel Dadush, and Noah Stephens-Davidowitz. Solving the closest vector problem in 2^n time—the discrete Gaussian strikes again! In *FOCS*, pages 563–582, 2015.

2 Miklós Ajtai. Random lattices and a conjectured 0-1 law about their polynomial time computable properties. In *FOCS*, pages 733–742. IEEE, 2002.

3 Ofer Amrani and Yair Beery. Efficient bounded-distance decoding of the hexacode and associated decoders for the leech lattice and the golay code. *IEEE Transactions on Communications*, 44(5):534–537, 1996.

4 Alexandr Andoni and Piotr Indyk. Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions. In *FOCS*, pages 459–468. IEEE, 2006.

5 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.

6 Alexandr Andoni, Piotr Indyk, Huy L Nguyên, and Ilya Razenshteyn. Beyond locality-sensitive hashing. In *SODA*, pages 1018–1028. SIAM, 2014.

7 Alexandr Andoni, Thijs Laarhoven, Ilya Razenshteyn, and Erik Waingarten. Optimal hashing-based time-space trade-offs for approximate near neighbors. In *SODA*, 2017.

8 Alexandr Andoni and Ilya Razenshteyn. Optimal data-dependent hashing for approximate near neighbors. In *STOC*, 2015.

9 Alexandr Andoni and Ilya Razenshteyn. Tight lower bounds for data-dependent locality-sensitive hashing. *arXiv preprint arXiv:1507.04299*, 2015.

10 Anja Becker, Léo Ducas, Nicolas Gama, and Thijs Laarhoven. New directions in nearest neighbor searching with applications to lattice sieving. In *SODA*, pages 10–24, Philadelphia, PA, USA, 2016. Society for Industrial and Applied Mathematics. URL: `http://dl.acm.org/citation.cfm?id=2884435.2884437`.

11 L. A. Carraher, P. A. Wilsey, and F. S. Annexstein. A gpgpu algorithm for c-approximate r-nearest neighbor search in high dimensions. In *2013 IEEE International Symposium on Parallel Distributed Processing*, pages 2079–2088, May 2013.

12 Tobias Christiani. A framework for similarity search with space-time tradeoffs using locality-sensitive filtering. In *SODA*, pages 31–46, Philadelphia, PA, USA, 2017. Society for Industrial and Applied Mathematics. URL: `http://dl.acm.org/citation.cfm?id=3039686.3039689`.

13 Daniel Dadush and Nicolas Bonifas. Short paths on the Voronoi graph and closest vector problem with preprocessing. In *SODA*, pages 295–314, 2015.

14 Mayur Datar, Nicole Immorlica, Piotr Indyk, and Vahab S Mirrokni. Locality-sensitive hashing scheme based on p-stable distributions. In *Proceedings of the Twentieth Annual Symposium on Computational Geometry (SOCG)*, pages 253–262. ACM, 2004.

15 Léo Ducas and Wessel P. J. van Woerden. The closest vector problem in tensored root lattices of type a and in their duals. *Designs, Codes and Cryptography*, 2017.

16 Uri Erez, Simon Litsyn, and Ram Zamir. Lattices which are good for (almost) everything. *IEEE Transactions on Information Theory*, 51(10):3401–3416, 2005.

17 Kave Eshghi and Shyamsundar Rajaram. Locality sensitive hash functions based on concomitant rank order statistics. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 221–229. ACM, 2008.

18 Aristides Gionis, Piotr Indyk, and Rajeev Motwani. Similarity search in high dimensions via hashing. In *VLDB*, pages 518–529, 1999.

19 Sariel Har-Peled, Piotr Indyk, and Rajeev Motwani. Approximate nearest neighbor: Towards removing the curse of dimensionality. *Theory of computing*, 8(1):321–350, 2012.

20 Piotr Indyk and Rajeev Motwani. Approximate nearest neighbors: towards removing the curse of dimensionality. In *STOC*, pages 604–613. ACM, 1998.

21 Hervé Jégou, Laurent Amsaleg, Cordelia Schmid, and Patrick Gros. Query adaptative locality sensitive hashing. In *IEEE International Conference on Acoustics, Speech and Signal Processing, 2008.*, pages 825–828. IEEE, 2008.

22 Ravi Kannan. Minkowski's convex body theorem and integer programming. *Mathematics of operations research*, 12(3):415–440, 1987.

23 Thijs Laarhoven. Sieving for shortest vectors in lattices using angular locality-sensitive hashing. In Rosario Gennaro and Matthew Robshaw, editors, *Advances in Cryptology – CRYPTO*, pages 3–22, Berlin, Heidelberg, 2015. Springer Berlin Heidelberg.

24 R. McKilliam, A. Grant, and I. Clarkson. Finding a closest point in a lattice of voronoi's first kind. *SIAM J. on Discret. Math.*, 28(3):1405–1422, 2014.

25 Rajeev Motwani, Assaf Naor, and Rina Panigrahy. Lower bounds on locality sensitive hashing. *SIAM Journal on Discrete Mathematics (SIDMA)*, 21(4):930–935, 2007.

26 Ryan ODonnell, Yi Wu, and Yuan Zhou. Optimal lower bounds for locality-sensitive hashing (except when q is tiny). *ACM Transactions on Computation Theory (TOCT)*, 6(1):5, 2014. Preliminary version in ICS 2011.

27 CA Rogers. Lattice coverings of space: the minkowski–hlawka theorem. *Proceedings of the London Mathematical Society*, 3(3):447–465, 1958.

28 Wolfgang M Schmidt. The measure of the set of admissible lattices. *Proceedings of the American Mathematical Society*, 9(3):390–403, 1958.

29 Gregory Shakhnarovich, Trevor Darrell, and Piotr Indyk. *Nearest-Neighbor Methods in Learning and Vision: Theory and Practice (Neural Information Processing)*. The MIT press, 2006.

30 Carl Ludwig Siegel. A mean value theorem in geometry of numbers. *Annals of Mathematics*, pages 340–347, 1945.

31 Kengo Terasawa and Yuzuru Tanaka. Spherical lsh for approximate nearest neighbor search on unit hypersphere. *Algorithms and Data Structures*, pages 27–38, 2007.

32 G. Valiant. Finding correlations in subquadratic time, with applications to learning parities and juntas. In *FOCS*, 2012.

33 Frank Vallentin. *Sphere coverings, lattices, and tilings (in low dimensions)*. PhD thesis, Technical University of Munich, 2003.

Differential Privacy on Finite Computers*†

Victor Balcer[1] and Salil Vadhan[2]

1 **Center for Research on Computation & Society, School of Engineering & Applied Sciences, Harvard University, 33 Oxford Street, Cambridge, MA 02138, USA**
vbalcer@g.harvard.edu

2 **Center for Research on Computation & Society, School of Engineering & Applied Sciences, Harvard University, 33 Oxford Street, Cambridge, MA 02138, USA**
salil_vadhan@harvard.edu

Abstract

We consider the problem of designing and analyzing differentially private algorithms that can be implemented on *discrete* models of computation in *strict* polynomial time, motivated by known attacks on floating point implementations of real-arithmetic differentially private algorithms (Mironov, CCS 2012) and the potential for timing attacks on expected polynomial-time algorithms. We use a case study: the basic problem of approximating the histogram of a categorical dataset over a possibly large data universe $\mathcal{X}$. The classic Laplace Mechanism (Dwork, McSherry, Nissim, Smith, TCC 2006 and J. Privacy & Confidentiality 2017) does not satisfy our requirements, as it is based on real arithmetic, and natural discrete analogues, such as the Geometric Mechanism (Ghosh, Roughgarden, Sundarajan, STOC 2009 and SICOMP 2012), take time at least linear in $|\mathcal{X}|$, which can be exponential in the bit length of the input.

In this paper, we provide strict polynomial-time discrete algorithms for approximate histograms whose simultaneous accuracy (the maximum error over all bins) matches that of the Laplace Mechanism up to constant factors, while retaining the same (pure) differential privacy guarantee. One of our algorithms produces a sparse histogram as output. Its "per-bin accuracy" (the error on individual bins) is worse than that of the Laplace Mechanism by a factor of $\log |\mathcal{X}|$, but we prove a lower bound showing that this is necessary for any algorithm that produces a sparse histogram. A second algorithm avoids this lower bound, and matches the per-bin accuracy of the Laplace Mechanism, by producing a compact and efficiently computable representation of a dense histogram; it is based on an $(n+1)$-wise independent implementation of an appropriately clamped version of the Discrete Geometric Mechanism.

1998 ACM Subject Classification F.2.m Analysis of Algorithms and Problem Complexity, Miscellaneous

Keywords and phrases Algorithms, Differential Privacy, Discrete Computation, Histograms

Digital Object Identifier 10.4230/LIPIcs.ITCS.2018.43

* This work was supported by NSF grant CNS-1237235 and CNS-1565387, a Simons Investigator Award, and a grant from the Sloan Foundation.

† A full version of the paper is available at [1], http://arxiv.org/abs/1709.05396.

1 Introduction

Differential Privacy [11] is by now a well-established framework for privacy-protective statistical analysis of sensitive datasets. Much work on differential privacy involves an interplay between statistics and computer science. Statistics provides many of the (non-private) analyses that we wish to approximate with differentially private algorithms, as well as probabilistic tools that are useful in analyzing such algorithms, which are necessarily randomized. From computer science, differential privacy draws upon a tradition of adversarial modeling and strong security definitions, techniques for designing and analyzing randomized algorithms, and considerations of algorithmic resource constraints (such as time and memory).

Because of its connection to statistics, it is very natural that much of the literature on differential privacy considers the estimation of real-valued functions on real-valued data (e.g. the sample mean) and introduces noise from continuous probability distributions (e.g. the Laplace distribution) to obtain privacy. However, these choices are incompatible with standard computer science models for algorithms (like the Turing machine or RAM model) as well as implementation on physical computers (which use only finite approximations to real arithmetic, e.g. via floating point numbers). This discrepancy is not just a theoretical concern; Mironov [21] strikingly demonstrated that common floating-point implementations of the most basic differentially private algorithm (the Laplace Mechanism) are vulnerable to real attacks. Mironov shows how to prevent his attack with a simple modification to the implementation, but this solution is specific to a single differentially private mechanism and particular floating-point arithmetic standard. His solution increases the error by a constant factor and is most likely more efficient in practice than the algorithm we will use to replace the Laplace Mechanism. However, he provides no bounds on asymptotic running time. Gazeau, Miller and Palamidessi [13] provide more general conditions for which an implementation of real numbers and a mechanism that perturbs the correct answer with noise maintains differential privacy. However, they do not provide an explicit construction with bounds on accuracy and running time.

From a theoretical point of view, a more appealing approach to resolving these issues is to avoid real or floating-point arithmetic entirely and only consider differentially private computations that involve discrete inputs and outputs, and rational probabilities, as first done in [10]. Such algorithms are realizable in standard discrete models of computation. However, some such algorithms have running times that are only bounded in expectation (e.g. due to sampling from an exponential distribution supported on the natural numbers), and this raises a potential vulnerability to timing attacks. If an adversary can observe the running time of the algorithm, it learns something about the algorithm's coin tosses, which are assumed to be secret in the definition of differential privacy. (Even if the time cannot be directly observed, in practice an adversary can determine an upper bound on the running time, which again is information that is implicitly assumed to be secret in the privacy definition.)

Because of these considerations, we advocate the following principle:

> ***Differential Privacy for Finite Computers:***
> *We should describe how to implement differentially private algorithms on* ***discrete*** *models of computation with* ***strict*** *bounds on running time (ideally polynomial in the bit length of the input) and* ***analyze*** *the effects of those constraints on both privacy and accuracy.*

Note that a strict *bound* on running time does not in itself prevent timing attacks, but once we have such a bound, we can pad all executions to take the same amount of time.

Also, while standard discrete models of computation (e.g. randomized Turing machines) are defined in terms of countable rather than finite resources (e.g. the infinite tape), if we have a strict bound on running time, then once we fix an upper bound on input length, they can indeed be implemented on a truly finite computer (e.g. like a randomized Boolean circuit).

In many cases, the above goal can be achieved by appropriate discretizations and truncations applied to a standard, real-arithmetic differentially private algorithm. However, such modifications can have a nontrivial price in accuracy or privacy, and thus we also call for a rigorous analysis of these effects.

In this paper, we carry out a case study of achieving "differential privacy for finite computers" for one of the first tasks studied in differential privacy, namely approximating a histogram of a categorical dataset. Even this basic problem turns out to require some nontrivial effort, particularly to maintain strict polynomial time, optimal accuracy and pure differential privacy when the data universe is large.

We recall the definition of differential privacy.

▶ **Definition 1.1** ([11])**.** Let $\mathcal{M} : \mathcal{X}^n \to \mathcal{R}$ be a randomized algorithm. We say $\mathcal{M}$ is **(ε, δ)-differentially private** if for every pair of neighboring datasets D and D' (datasets differing on one row) and every subset $S \subseteq \mathcal{R}$

$$\Pr[\mathcal{M}(D) \in S] \leq e^{\varepsilon} \cdot \Pr[\mathcal{M}(D') \in S] + \delta$$

We say an (ε, δ)-differentially private algorithm satisfies **pure differential privacy** when $\delta = 0$ and say it satisfies **approximate differential privacy** when $\delta > 0$.

In this paper, we study the problem of estimating the *histogram* of a dataset $D \in \mathcal{X}^n$, which is the vector $c = c(D) \in \mathbb{N}^{\mathcal{X}}$, where c_x is the number of rows in D that have value x. Histograms can be approximated while satisfying differential privacy using the *Laplace Mechanism*, introduced in the original paper of Dwork, McSherry, Nissim and Smith [11]. Specifically, to obtain $(\varepsilon, 0)$-differential privacy, we can add independent noise distributed according to a Laplace distribution, specifically $\mathrm{Lap}(2/\varepsilon)$, to each component of c and output the resulting vector $\tilde{c}$. Here $\mathrm{Lap}(2/\varepsilon)$ is the *continuous*, real-valued random variable with probability density function $f(z)$ that is proportional to $\exp(-\varepsilon \cdot |z|/2)$. The Laplace Mechanism also achieves very high accuracy in two respects:

Per-Query Error: For each bin $x \in \mathcal{X}$, with high probability we have $|\tilde{c}_x - c_x| \leq O(1/\varepsilon)$.
Simultaneous Error: With high probability, we have $\max_x |\tilde{c}_x - c_x| \leq O(\log(|\mathcal{X}|)/\varepsilon)$.

Note that both of the bounds are independent of the number n of rows in the dataset, and so the fractional error vanishes linearly as n grows.

Simultaneous error is the more well-studied notion in the differential privacy literature, but we consider per-query error to be an equally natural concept: if we think of the approximate histogram $\tilde{c}$ as containing approximate answers to the $|\mathcal{X}|$ different counting queries corresponding to the bins of $\mathcal{X}$, then per-query error captures the error as experienced by an analyst who may be only interested in one or a few of the bins of $\tilde{c}$. The advantage of considering per-query error is that it can be significantly smaller than the simultaneous error, as is the case in the Laplace Mechanism when the data universe $\mathcal{X}$ is very large. It is known that both of the error bounds achieved by the Laplace Mechanism are optimal up to constant factors; no $(\varepsilon, 0)$-differentially private algorithm for histograms can achieve smaller per-query error or simultaneous error [17, 2].

Unfortunately, the Laplace Mechanism uses real arithmetic and thus cannot be implemented on a finite computer. To avoid real arithmetic, we could use the Geometric Mechanism

[14], which adds noise to each component of c according to the 2-sided geometric distribution, Geo($2/\varepsilon$), which is supported on the integers and has probability mass function $f(z) \propto \exp(-\varepsilon \cdot |z|/2)$. However, this mechanism uses integers of unbounded size and thus cannot be implemented on a finite computer. Indeed, while the algorithm can be implemented with a running time that is bounded in expectation (after reducing ε so that $e^{\varepsilon/2}$ and hence all the probabilities are rational numbers), truncating long executions or allowing an adversary to observe the actual running time can lead to a violation of differential privacy. Thus, as first described by Dwork, Kenthapadi, McSherry, Mironov and Naor [10], it is better to restrict the output of the mechanism to a binary representation of fixed length in order to avoid small tail probabilities. Similarly, we work with the Truncated Geometric Mechanism of Ghosh, Roughgarden and Sundararajan [14], where we clamp each noisy count $\tilde{c}_x$ to the interval $[0, n]$. We observe that the resulting probability distribution of $\tilde{c}_x$, supported on $\{0, 1, \ldots, n\}$, can be described explicitly in terms of c_x, ε and n, and it can be sampled in polynomial time using only integer arithmetic (after ensuring $e^{\varepsilon/2}$ is rational). Thus, we obtain:

▶ **Theorem 1.2** (Bounded Geometric Mechanism, informal statement of Thm. 5.1)**.** *For every finite $\mathcal{X}$, n and $\varepsilon \in (0, 1]$, there is an $(\varepsilon, 0)$-differentially private algorithm $\mathcal{M} : \mathcal{X}^n \to \{0, 1, \ldots, n\}^{\mathcal{X}}$ for histograms achieving:*

- *Per-query error $O(1/\varepsilon)$.*
- *Simultaneous error $O(\log(|\mathcal{X}|)/\varepsilon)$.*
- *Strict running time $\tilde{O}\left((|\mathcal{X}|/\varepsilon) \cdot \log^2 n\right) + O(n \log n \cdot \log |\mathcal{X}|)$*

We note that while we only consider our particular definition of per-query accuracy, namely that with high probability $|\tilde{c}_x - c_x| \leq O(1/\varepsilon)$, Ghosh et al. [14] proved that the output of the Bounded Geometric Mechanism can be used (with post-processing) to get optimal expected loss with respect to an extremely general class of loss functions and arbitrary priors. The same result applies to each individual noisy count $\tilde{c}_x$ output by our mechanism, since each bin is distributed according to the Bounded Geometric Mechanism (up to a modification of ε to ensure rational probabilities).

The Bounded Geometric Mechanism is not polynomial time for large data universes $\mathcal{X}$. Indeed, its running time (and output length) is linear in $|\mathcal{X}|$, rather than polynomial in the bit length of data elements, which is $\log |\mathcal{X}|$. To achieve truly polynomial time, we can similarly discretize and truncate a variant of the Stability-Based Histogram that was introduced by Korolova, Kenthapadi, Mishra and Ntoulas [19], and explicitly described by Bun, Nissim and Stemmer [4]. This mechanism only adds Lap($2/\varepsilon$) noise to the *nonzero* components of c_x and then retains only the noisy values $\tilde{c}_x$ that are larger than a threshold $t = \Theta(\log(1/\delta)/\varepsilon)$. Thus, the algorithm only outputs a partial histogram, i.e. counts $\tilde{c}_x$ for a subset of the bins x, with the rest of the counts being treated as zero. By replacing the use of the Laplace Mechanism with the (rational) Bounded Geometric Mechanism as above, we can implement this algorithm in strict polynomial time:

▶ **Theorem 1.3** (Stability-Based Histogram, formal statement omitted from this version)**.** *For every finite $\mathcal{X}$, n, $\varepsilon \in (0, 1]$ and $\delta \in (0, 1/n)$, there is an (ε, δ)-differentially private algorithm $\mathcal{M} : \mathcal{X}^n \to \{0, 1, \ldots, n\}^{\subseteq \mathcal{X}}$ for histograms achieving:*

- *Per-query error $O(1/\varepsilon)$ on bins with true count at least $O(\log(1/\delta)/\varepsilon)$.*
- *Simultaneous error $O(\log(1/\delta)/\varepsilon)$.*
- *Strict running time $\tilde{O}((n/\varepsilon) \cdot \log(1/\delta)) + O(n \log n \cdot \log |\mathcal{X}|)$.*

Notice that the simultaneous error bound of $O(\log(1/\delta)/\varepsilon)$ is better than what is achieved by the Laplace Mechanism when $\delta > 1/|\mathcal{X}|$, and is known to be optimal up to constant

factors in this range of parameters. The fact that this error bound is independent of the data universe size $|\mathcal{X}|$ makes it tempting to apply even for infinite data domains $\mathcal{X}$. However, we note that when $\mathcal{X}$ is infinite, it is impossible for the algorithm to have a strict bound on running time (as it needs time to read arbitrarily long data elements) and thus is vulnerable to timing attacks and is not implementable on a finite computer.

Note also that the per-query error bound only holds on bins with large enough true count (namely, those larger than our threshold t); we will discuss this point further below.

A disadvantage of the Stability-based Histogram is that it sacrifices pure differential privacy. It is natural to ask whether we can achieve polynomial running time while retaining pure differential privacy. A step in this direction was made by Cormode, Procopiuc, Srivastava and Tran [9]. They observe that for an appropriate threshold $t = \Theta(\log(|\mathcal{X}|)/\varepsilon)$, if we run the Bounded Geometric Mechanism and only retain the noisy counts $\tilde{c}_x$ that are larger than t, then the expected number of bins that remain is less than $n + 1$. Indeed, the expected number of bins we retain whose true count is zero ("empty bins") is less than 1. They describe a method to directly sample the distribution of the empty bins that are retained, without actually adding noise to all $|\mathcal{X}|$ bins. This yields an algorithm whose output length is polynomial in expectation. However, the output length is not strictly polynomial, as there is a nonzero probability of outputting all $|\mathcal{X}|$ bins. And it is not clear how to implement the algorithm even in expected polynomial time, because even after making the probabilities rational, they have denominators of bit length linear in $|\mathcal{X}|$.

To address these issues, we consider a slightly different algorithm. Instead of trying to retain all noisy counts $\tilde{c}_x$ that are larger than some fixed threshold t, we retain the n largest noisy counts (since there are at most n nonzero true counts). This results in a mechanism whose output length is always polynomial, rather than only in expectation. However, the probabilities still have denominators of bit length linear in $|\mathcal{X}|$. Thus, we show how to approximately sample from this distribution, to within an arbitrarily small statistical distance δ, at the price of a $\mathrm{poly}(\log(1/\delta))$ increase in running time. Naively, this would result only in $(\varepsilon, O(\delta))$-differential privacy. However, when δ is significantly smaller than $1/|\mathcal{R}|$, where $\mathcal{R}$ is the range of the mechanism, we can convert an (ε, δ)-differentially private mechanism to an $(\varepsilon, 0)$-differentially private mechanism by simply outputting a uniformly random element of $\mathcal{R}$ with small probability. (A similar idea for the case that $|\mathcal{R}| = 2$ has been used in [18, 5].) Since our range is of at most exponential size (indeed at most polynomial in bit length), the cost in our runtime for taking $\delta \ll 1/|\mathcal{R}|$ is at most polynomial. With these ideas we obtain:

▶ **Theorem 1.4** (Pure DP Histogram in Polynomial Time, informal statement of Thm. 6.6)**.** *For every finite $\mathcal{X}$, n and $\varepsilon \in (0, 1]$, there is an $(\varepsilon, 0)$-differentially private algorithm $\mathcal{M} : \mathcal{X}^n \to \{0, 1, \ldots, n\}^{\subseteq \mathcal{X}}$ for histograms achieving:*

- *Per-query error $O(1/\varepsilon)$ on bins with true count at least $O(\log(|\mathcal{X}|)/\varepsilon)$.*
- *Simultaneous error $O(\log(|\mathcal{X}|)/\varepsilon)$.*
- *Strict running time $\tilde{O}(n^2 \cdot \log^2 |\mathcal{X}| + (n/\varepsilon) \cdot \log^2 |\mathcal{X}|)$.*

It is an open problem as to whether or not we can improve the nearly quadratic dependence in running time on n to nearly linear while maintaining the sparsity, privacy and accuracy guarantees achieved in Theorem 1.4.

Both Theorems 1.3 and 1.4 only retain per-query error $O(1/\varepsilon)$ on bins with a large enough true count. In the full version of the paper [1], we also prove a lower bound showing that this limitation is inherent in any algorithm that outputs a sparse histogram (as both of these algorithms do).

▶ **Theorem 1.5** (Lower Bound on Per-Query Error for Sparse Histograms)**.** *Suppose that there is an (ε, δ)-differentially private algorithm $\mathcal{M} : \mathcal{X}^n \to \{0, 1, \ldots, n\}^{\mathcal{X}}$ for histograms that always outputs histograms with at most n' nonempty bins and has per-query error at most E on all bins. Then*

$$E \geq \Omega\left(\frac{\min\{\log|\mathcal{X}|, \log(1/\delta)\}}{\varepsilon}\right)$$

provided that $\varepsilon > 0$, $\varepsilon^2 > \delta > 0$ and $|\mathcal{X}| \geq (n')^2$.

This lower bound is similar in spirit to a lower bound of [2], which shows that no $(\varepsilon, 0)$-differentially private PAC learner for "point functions" (functions that are 1 on exactly one element of the domain) can produce sparse functions as hypotheses.

To bypass this lower bound, we can consider algorithms that produce succinct descriptions of dense histograms. That is, the algorithm can output a polynomial-length description of a function $\tilde{c} : \mathcal{X} \to [0, n]$ that can be evaluated in polynomial time, even though $\mathcal{X}$ may be of exponential size.

We show that this relaxation allows us to regain per-query error $O(1/\varepsilon)$.

▶ **Theorem 1.6** (Polynomial-Time DP Histograms with Optimal Per-Query Accuracy, informal statement of Thm. 7.2)**.** *For every finite $\mathcal{X}$, n and $\varepsilon \in (0, 1]$, there is an $(\varepsilon, 0)$-differentially private algorithm $\mathcal{M} : \mathcal{X}^n \to \mathcal{H}$ for histograms (where $\mathcal{H}$ is an appropriate class of succinct descriptions of histograms) achieving:*

- *Per-query error $O(1/\varepsilon)$.*
- *Simultaneous error $O(\log(|\mathcal{X}|)/\varepsilon)$.*
- *Strict running time $O(n) \cdot \tilde{O}((1/\varepsilon^2) \cdot (\log^2 n + \log^2 |\mathcal{X}|))$*
- *Evaluation time $O(n) \cdot \tilde{O}((1/\varepsilon) \cdot (\log n + \log |\mathcal{X}|))$.*

The algorithm is essentially an $(n + 1)$-wise independent instantiation of the Bounded Geometric Mechanism. Specifically, we release a function $h : \mathcal{X} \to \{0, 1\}^r$ selected from an $(n + 1)$-wise independent family of hash functions, and for each $x \in \mathcal{X}$, we view $h(x)$ as coin tosses specifying a sample from the Bounded Geometric Distribution. That is, we let $S : \{0, 1\}^r \to [0, n]$ be an efficient sampling algorithm for the Bounded Geometric Distribution, and then $\tilde{c}_x = S(h(x))$ is our noisy count for x. The hash function is chosen randomly from the family conditioned on values $\tilde{c}_x$ for the nonempty bins x, which we obtain by running the actual Bounded Geometric Mechanism on those bins. The $(n + 1)$-wise independence ensures that the behavior on any two neighboring datasets (which together involve at most $n + 1$ distinct elements of $\mathcal{X}$) are indistinguishable in the same way as in the ordinary Bounded Geometric Mechanism. The per-query accuracy comes from the fact that the marginal distributions of each of the noisy counts are the same as in the Bounded Geometric Mechanism. (Actually, we incur a small approximation error in matching the domain of the sampling procedure to the range of a family of hash functions.)

As far as we know, the only other use of limited independence in constructing differentially private algorithms is a use of pairwise independence by [2] in differentially private PAC learning algorithms for the class of point functions. Although that problem is related to the one we consider (releasing a histogram amounts to doing "query release" for the class of point functions, as discussed below), the design and analysis of our algorithm appears quite different. (In particular, our analysis seems to rely on $(n + 1)$-wise independence in an essential way.)

Another potential interest in our technique is as another method for bypassing limitations of *synthetic data* for *query release*. Here, we have a large family of predicates $\mathcal{Q} = \{q : \mathcal{X} \to \{0,1\}\}$, and are interested in differentially private algorithms that, given a dataset $D = (x_1, \ldots, x_n) \in \mathcal{X}^n$, output a "summary" $\mathcal{M}(D)$ that allows one to approximate the answers to all of the *counting queries* $q(D) = \sum_i q(x_i)$ associated with predicates $q \in \mathcal{Q}$. For example, if $\mathcal{Q}$ is the family of *point functions* consisting of all predicates that evaluate to 1 on exactly one point in the data universe $\mathcal{X}$, then this query release problem amounts to approximating the histogram of D. The fundamental result of Blum, Ligett, and Roth [3] and successors show that this is possible even for families $\mathcal{Q}$ and data universes $\mathcal{X}$ that are of size exponential in n. Moreover, the summaries produced by these algorithms has the form of a synthetic dataset — a dataset $\hat{D} \in \mathcal{X}^{\hat{n}}$ such that for every query $q \in \mathcal{Q}$, we have $q(\hat{D}) \approx q(D)$. Unfortunately, it was shown in [24] that even for very simple families $\mathcal{Q}$ of queries, such correlations between pairs of binary attributes, constructing such a differentially private synthetic dataset requires time exponential in the bit length $\log |\mathcal{X}|$ of data universe elements. Thus, it is important to find other ways of representing approximate answers to natural families $\mathcal{Q}$ of counting queries, which can bypass the inherent limitations of synthetic data, and progress along these lines was made in a variety of works [15, 7, 16, 23, 6, 12]. Our algorithm, and its use of $(n+1)$-wise independence, can be seen as yet another representation that bypasses a limitation of synthetic data (albeit a statistical rather than computational one). Indeed, a sparse histogram is simply a synthetic dataset that approximates answers to all point functions, and by Theorem 1.5, our algorithm achieves provably better per-query accuracy than is possible with synthetic datasets. This raises the question of whether similar ideas can also be useful in bypassing the computational limitations of synthetic data for more complex families of counting queries.

2 Preliminaries

Throughout this paper, let $\mathbb{N}$ be the set $\{0, 1, \ldots\}$, $\mathbb{N}_+$ be the set $\{1, 2, \ldots\}$ and $\mathbb{N}^{-1}$ be the set $\{1/n : n \in \mathbb{N}_+\}$. For $n \in \mathbb{N}_+$, let $[n]$ denote the set $\{0, \ldots, n\}$ and $[n]_+$ denote the set $\{1, \ldots, n\}$. (Notice that $|[n]| = n + 1$ while $|[n]_+| = n$.) Given a set A and finite set B, we define A^B to be the set of length $|B|$ vectors over A indexed by the elements of B.

2.1 Histograms

For $x \in \mathcal{X}$, the **point function** $c_x : \mathcal{X}^n \to \mathbb{N}$ is defined to count the number of occurrences of x in a given dataset, i.e. $c_x(D) = |\{i \in [n]_+ : D_i = x\}|$. In this paper we focus on algorithms for privately releasing approximations to the values of all point functions, also known as a **histogram**. A histogram is a collection of **bins**, one for each element x in the data universe, with the x^{th} bin consisting of its **label** x and a **count** $c_x \in \mathbb{N}$.

2.1.1 Representations

The input to our algorithms is always a dataset (i.e. an element $D \in \mathcal{X}^n$) and the outputs represent approximate histograms. We consider the following histogram representations as our algorithms' outputs:

- A vector in $\mathbb{N}^{\mathcal{X}}$. We use $\{\tilde{c}_x\}_{x \in \mathcal{X}}$ to denote a histogram where $\tilde{c}_x \in \mathbb{N}$ is the approximate count for the element x.
- A partial vector $h \in (\mathcal{X} \times \mathbb{N})^*$ such that each element $x \in \mathcal{X}$ appears at most once in h with each pair $(x, \tilde{c}_x) \in \mathcal{X} \times \mathbb{N}$ interpreted as element x having approximate count $\tilde{c}_x$.

Elements x not listed in the partial vector are assumed to have count $\tilde{c}_x = 0$. Implicitly, an algorithm can return a partial vector by releasing bins for a subset of $\mathcal{X}$.[1]

- A data structure, encoded as a string, which defines a function $h : \mathcal{X} \to \mathbb{N}$ where $h(x)$, denoted h_x, is the approximate count for $x \in \mathcal{X}$ and h_x is efficiently computable given this data structure (e.g. time polynomial in the length of the data structure). In Section 7, this data structure consists of the coefficients of a polynomial, along with some parameters.

Each representation is able to express any histogram over $\mathcal{X}$. The difference between them is the memory used and the efficiency of computing a count. For example, computing the approximate count for $x \in \mathcal{X}$, when using the data structure representation is bounded by the time it takes to compute the associated function. But when using partial vectors, one only needs to iterate through the vector to determine the approximate count.

We define the following class of histograms. Let $\mathcal{H}_{n,n'}(\mathcal{X}) \subseteq \mathbb{N}^{\mathcal{X}}$ be the set of all histograms over $\mathcal{X}$ with integer counts in $[0, n]$ (or $\mathbb{N}$ when $n = \infty$) and at most n' of them nonzero. By using partial vectors each element of $\mathcal{H}_{n,n'}(\mathcal{X})$ can be stored in $O(n' \cdot (\log n + \log |\mathcal{X}|))$ bits, which is shorter than the vector representation when $n' = o(|\mathcal{X}|/\log |\mathcal{X}|)$.

2.1.2 Accuracy

In order to preserve privacy, our algorithms return histograms with noise added to the counts. Therefore, it is crucial to understand their accuracy guarantees. So given a dataset $D \in \mathcal{X}^n$ we compare the **noisy count** $\tilde{c}_x = \mathcal{M}(D)_x$ of $x \in \mathcal{X}$ (the count released by algorithm $\mathcal{M}$) to its **true count**, $c_x(D)$. We focus on the following two metrics:

▶ **Definition 2.1.** A histogram algorithm $\mathcal{M} : \mathcal{X}^n \to \mathbb{N}^{\mathcal{X}}$ has **(a, β)-per-query accuracy** if

$$\forall D \in \mathcal{X}^n \quad \forall x \in \mathcal{X} \qquad \Pr[|\mathcal{M}(D)_x - c_x(D)| \leq a] \geq 1 - \beta$$

▶ **Definition 2.2.** A histogram algorithm $\mathcal{M} : \mathcal{X}^n \to \mathbb{N}^{\mathcal{X}}$ has **(a, β)-simultaneous accuracy** if

$$\forall D \in \mathcal{X}^n \qquad \Pr[\forall x \in \mathcal{X} \quad |\mathcal{M}(D)_x - c_x(D)| \leq a] \geq 1 - \beta$$

Respectively, these metrics capture the maximum error for any one bin and the maximum error simultaneously over all bins. However, we may not always be able to achieve as good per-query accuracy as we want. So we will also use the following relaxation which bounds the error only on bins with large enough true count.

▶ **Definition 2.3.** A histogram algorithm $\mathcal{M} : \mathcal{X}^n \to \mathbb{N}^{\mathcal{X}}$ has **(a, β)-per-query accuracy on counts larger than t** if

$$\forall D \in \mathcal{X}^n \quad \forall x \in \mathcal{X} \text{ s.t. } c_x(D) > t \qquad \Pr[|\mathcal{M}(D)_x - c_x(D)| \leq a] \geq 1 - \beta$$

2.2 Probability Terminology

▶ **Definition 2.4.** Let Z be an integer-valued random variable. The **probability mass function of Z**, denoted f_Z, is the function $f_Z(z) = \Pr[Z = z]$ for all $z \in \mathbb{Z}$. The **cumulative distribution function of Z**, denoted F_Z, is the function $F_Z(z) = \Pr[Z \leq z]$ for all $z \in \mathbb{Z}$. The **support** of Z, denoted $\text{supp}(Z)$, is the set of elements for which $f(z) \neq 0$.

[1] Note that the order in which bins are released can result in a breach of privacy (e.g. releasing the bins of elements in the dataset before the bins of elements not in the dataset). As a result, our algorithms always sort the released bins according to a predefined ordering based only on $\mathcal{X}$.

▶ **Definition 2.5.** Let Y and Z be random variables taking values in discrete range $\mathcal{R}$. The **total variation distance between Y and Z** is defined as

$$\Delta(Y,Z) = \max_{A \subseteq \mathcal{R}} \left| \Pr[Y \in A] - \Pr[Z \in A] \right| = \frac{1}{2} \cdot \sum_{a \in \mathcal{R}} \left| \Pr[Z = a] - \Pr[Y = a] \right|$$

2.2.1 Sampling

Because we are interested in the computational efficiency of our algorithms we need to consider the efficiency of sampling from various distributions.

A standard method for sampling a random variable is via **inverse transform sampling**. Let $\mathrm{Unif}(A)$ denote the uniform distribution over the set A.

▶ **Lemma 2.6.** *Let $U \sim \mathrm{Unif}((0,1])$. Then for any integer-valued random variable Z we have $F_Z^{-1}(U) \sim Z$ where $F_Z^{-1}(u)$ is defined as $\min\{z \in \mathrm{supp}(Z) : F_Z(z) \geq u\}$.*

If Z, the random variable we wish to sample, has finite support we can compute the inverse cumulative distribution by performing binary search on $\mathrm{supp}(Z)$ to find the minimum. This method removes the need to compute the inverse function of the cumulative distribution function. If in addition, the cumulative distribution function of Z can be represented by rational numbers, then we only need to sample from a discrete distribution instead of $(0,1]$.

▶ **Lemma 2.7.** *Let Z be an integer-valued random variable with finite support and has all probabilities of its cumulative distribution function expressible as rational numbers with denominator d. Then $F_Z^{-1}(U) \sim Z$ where $U \sim (1/d) \cdot \mathrm{Unif}([d]_+)$ and $F_Z^{-1}(u)$ is defined as $\min\{z \in \mathrm{supp}(Z) : F_Z(z) \geq u\}$.*

2.2.2 Order Statistics

▶ **Definition 2.8.** Let $Z_1, \ldots, Z_\ell$ be integer-valued random variables. The **i-th order statistic of $Z_1, \ldots, Z_\ell$** denoted $Z_{(i)}$ is the i-th smallest value among $Z_1, \ldots, Z_\ell$.

▶ **Lemma 2.9.** *Let $Z_1, \ldots, Z_\ell$ be i.i.d. integer-valued random variables with cumulative distribution function F. Then $F_{Z_{(\ell)}}(z) = (F(z))^\ell$ and*

$$F_{Z_{(i)} | Z_{(i+1)} = v_{i+1}, \ldots Z_{(\ell)} = v_\ell}(z) = F_{Z_{(i)} | Z_{(i+1)} = v_{i+1}}(z) = \begin{cases} 1 & \text{if } z > v_{i+1} \\ \left(F(z)/F(v_{i+1})\right)^i & \text{otherwise} \end{cases}$$

for all $1 \leq i < \ell$ and $v_{i+1} \leq v_{i+2} \leq \ldots \leq v_\ell$ all in the support of Z_1.

From this lemma, we can iteratively sample random variables distributed identically to $Z_{(\ell)}, Z_{(\ell-1)}, \ldots, Z_{(i)}$ without having to sample all ℓ of the original random variables.

2.3 Model of Computation

We analyze the running time of our algorithms with respect to the **w-bit word RAM model** taking w logarithmic in our input length, namely $w = O(\log n + \log\log|\mathcal{X}|)$. In this model, memory accesses and basic operations (arithmetic, comparisons and logical) on w-bit words are constant time. In addition, we assume the data universe $\mathcal{X} = [m]$ for some $m \in \mathbb{N}$. Some parameters to our algorithms are rational. We represent rationals by pairs of integers.

Some of our algorithms will use numbers that span many words. And one of our algorithms operates on finite fields $\mathbb{F}_d$ where $d = 2^{2 \cdot 3^\ell}$ for some $\ell \in \mathbb{N}$. We will use the fact

that $\mathbb{F}_d \simeq \mathbb{F}_2[x]/(x^{2\cdot 3^\ell} + x^{3^\ell} + 1)$ [20] to provide an explicit representation of $\mathbb{F}_d$. For ease of notation, we make the following assumptions on running time: (i) multiplying two x-bit numbers is $\tilde{O}(x)$ [25], (ii) multiplying two elements of $\mathbb{F}_d$ is $\tilde{O}(\log d)$ [22], (iii) multiplying two degree q polynomials over $\mathbb{F}_d$ is $\tilde{O}(q \cdot \log d)$ [22], (iv) evaluating a degree q polynomial over $\mathbb{F}_d$ is $O(q) \cdot \tilde{O}(\log d)$ [25] and (v) polynomial interpolation of q distinct points over $\mathbb{F}_d$ is $\tilde{O}(q \cdot \log^2 d)$ [25].

Our algorithms require randomness so we assume that they have access to an oracle that when given a number $d \in \mathbb{N}_+$ returns a uniformly random integer between 1 and d inclusive.

Finally, for representing histograms as partial vectors, we will assume internally to the algorithms that they are stored as red-black trees. This will allow us to insert and search for elements in $O(\log n \cdot \log |\mathcal{X}|)$ time [8]. However, when releasing a partial vector we return a list of bins using an in-order traversal of tree as the tree's structure could violate privacy.

3 A General Framework for Implementing Differential Privacy

In this section, we outline a basic framework for implementing a pure differentially private algorithm $\mathcal{M}$ on a finite computer with only a small loss in privacy and possibly a small loss in accuracy. It can be broken down into the following steps:

1. Start by discretizing the input and output of $\mathcal{M}$ so that they can only take on a finite number of values (e.g. rounding a real-valued number to the nearest integer in some finite set). Depending on how utility is measured, the loss in accuracy by discretizing may be acceptable.
2. Then find an algorithm $\mathcal{M}'$ that runs on a finite computer and approximates the output distribution of the discretized version of $\mathcal{M}$ to within "small" statistical distance. Notice that $\mathcal{M}'$ is only guaranteed to satisfy approximate differential privacy and may not satisfy pure differentially privacy. (This step may requires a non-trivial amount of work. For one example, see Theorem 6.5.)
3. Finally, provided that the statistical distance of the previous step is small enough, by mixing $\mathcal{M}'$ with uniformly random output (from the discretized and finite output space), the resulting algorithm satisfies pure differentially privacy.

We will use this framework several times in designing our algorithms. Here we start by formalizing Step 3. That is, for algorithms whose output distribution is close in total variation distance to that of a pure differentially private algorithm, we construct an algorithm satisfying pure differentially privacy by mixing it with random output inspired by similar techniques in [18, 5].

Algorithm 1 $\mathcal{M}^*_{\mathcal{M}',\mathcal{D},\gamma}(D)$ for $D \in \mathcal{X}^n$ where $\mathcal{R}$ is discrete and finite, an algorithm $\mathcal{M}' : \mathcal{X}^n \to \mathcal{R}$, a distribution $\mathcal{D}$ over $\mathcal{R}$ and $\gamma \in \mathbb{N}^{-1}$

1. With probability $1 - \gamma$ release $\mathcal{M}'(D)$.
2. Otherwise release an element sampled from the distribution $\mathcal{D}$.

▶ **Lemma 3.1.** *Suppose that there is an $(\varepsilon, 0)$-differentially private algorithm $\mathcal{M} : \mathcal{X}^n \to \mathcal{R}$ such that $\Delta(\mathcal{M}(D), \mathcal{M}'(D)) \leq \delta$ for all input datasets $D \in \mathcal{X}^n$ with parameter $\delta \in [0, 1)$. Then the algorithm $\mathcal{M}^*_{\mathcal{M}',\mathcal{D},\gamma} : \mathcal{X}^n \to \mathcal{R}$ has the following properties:*

- *$(\varepsilon, 0)$-differential privacy whenever*

$$\delta \leq \frac{e^\varepsilon - 1}{e^\varepsilon + 1} \cdot \frac{\gamma}{1 - \gamma} \cdot \min_{r \in \mathcal{R}} \left\{ \Pr_{Z \sim \mathcal{D}}[Z = r] \right\} \tag{1}$$

- *Running time* $O(\log(1/\gamma)) + \text{Time}(\mathcal{M}') + \text{Time}(\mathcal{D})$ *where* $\text{Time}(\mathcal{D})$ *is the time to sample from the distribution* $\mathcal{D}$.

By taking γ and δ small enough and satisfying (1), the algorithm $\mathcal{M}^*_{\mathcal{M}',\mathcal{D},\gamma}$ satisfies pure differential privacy and has nearly the same utility as $\mathcal{M}$ (due to having a statistical distance at most $\gamma + \delta$ from $\mathcal{M}$) while allowing for a possibly more efficient implementation since we only need to approximately sample from the output distribution of $\mathcal{M}$.

To maximize the minimum in (1), one can take $\mathcal{D} \sim \text{Unif}(\mathcal{R})$. However, it may the case that sampling this distribution exactly is inefficient and we are willing to trade needing a smaller δ to maintain pure differentially privacy for a faster sampling algorithm.

4 Counting Queries

Before discussing algorithms for privately releasing histograms, we show how to privately answer a single counting query using only integers of bounded length. While there exist known algorithms for this problem [10, 21], our algorithms have additional properties that will be used to construct histogram algorithms in later sections. In general, counting queries have as input the dataset $D \in \mathcal{X}^n$ and the bin x to query. However, we will take the true count, $c_x(D)$, as the input to our counting query algorithms. When constructing histogram algorithms in later sections, this will allow us to improve the running time as we will only need to iterate through the dataset once to determine all true counts prior to answering any counting query. In addition, we would like to keep track of the randomness used by our algorithms so we write that as an explicit second input. As a result, we have the following definitions:

▶ **Definition 4.1.** Let $n, d \in \mathbb{N}_+$. We say an algorithm $\mathcal{M} : [n] \times [d]_+ \to [n]$ is **(ε, δ)-differentially private for counting queries** if the algorithm $\mathcal{M} : \{0,1\}^n \to [n]$ defined as $\mathcal{M}(D) = \mathcal{M}(\sum_{i=1}^n D_i, U)$ where $U \sim \text{Unif}([d]_+)$ is (ε, δ)-differentially private.

▶ **Definition 4.2.** Let $n, d \in \mathbb{N}_+$. We say $\mathcal{M} : [n] \times [d]_+ \to [n]$ has **(α, β)-accuracy** if for all $c \in [n]$, $\Pr[|\mathcal{M}(c, U) - c| \leq a] \geq 1 - \beta$ where $U \sim \text{Unif}([d]_+)$.

▶ **Definition 4.3.** Let $n, d \in \mathbb{N}_+$ and $\mathcal{M} : [n] \times [d]_+ \to [n]$ be deterministic. Let the **scaled cumulative distribution function of $\mathcal{M}$ at 0** denoted $F_{\mathcal{M}}$ be the function $F_{\mathcal{M}} : [n] \to [d]_+$ defined as $F_{\mathcal{M}}(z) = d \cdot F_{\mathcal{M}(0,U)}(z)$ where $U \sim \text{Unif}([d]_+)$ for all $z \in [n]$.

4.1 The Geometric Mechanism

As shown by Dwork, McSherry, Nissim and Smith [11], we can privately release a counting query by adding appropriately scaled Laplace noise to the count. Because our algorithm's outputs are counts, we do not need to use continuous noise and instead use a discrete analogue, as in [10, 14].

We say an integer-valued random variable Z follows a **two-sided geometric distribution with scale parameter s centered at $c \in \mathbb{Z}$** (denoted $Z \sim c + \text{Geo}(s)$) if its probability mass function $f_Z(z)$ is proportional to $e^{-|z-c|/s}$. It can be verified that f_Z and its cumulative distribution function F_Z are

$$f_Z(z) = \left(\frac{e^{1/s} - 1}{e^{1/s} + 1}\right) \cdot e^{-|z-c|/s} \qquad F_Z(z) = \begin{cases} \frac{e^{1/s}}{e^{1/s}+1} \cdot e^{-(c-z)/s} & \text{if } z \leq c \\ 1 - \frac{1}{e^{1/s}+1} \cdot e^{-(z-c)/s} & \text{otherwise} \end{cases}$$

for all $z \in \mathbb{Z}$. When c is not specified, it is assumed to be 0.

Now, we state the counting query algorithm using discrete noise, formally studied in [14]. We will not keep track of the randomness used by this algorithm, but to match our syntax for counting query algorithms we use a dummy parameter.

Algorithm 2 $\texttt{GeometricMechanism}_{n,\varepsilon}(c, 1)$ for $c \in [n]$ where $n \in \mathbb{N}_+$ and $\varepsilon > 0$

1. Return $\tilde{c}$ set to $c + \text{Geo}(2/\varepsilon)$ clamped to the interval $[0, n]$. i.e.

$$\tilde{c} = \begin{cases} 0 & \text{if } Z \leq 0 \\ n & \text{if } Z \geq n \\ Z & \text{otherwise} \end{cases} \qquad \text{where } Z = c + \text{Geo}(2/\varepsilon).$$

▶ **Theorem 4.4.** *Let $n \in \mathbb{N}_+$ and $\varepsilon > 0$. Then $\texttt{GeometricMechanism}_{n,\varepsilon} : [n] \times [1]_+ \to [n]$ has the following properties:*

- $\texttt{GeometricMechanism}_{n,\varepsilon}$ *is $(\varepsilon/2, 0)$-differentially private for counting queries [14].*
- $\texttt{GeometricMechanism}_{n,\varepsilon}$ *has (a, β)-accuracy for $\beta \in (0, 1)$ and $a = \lceil (2/\varepsilon) \cdot \ln(1/\beta) \rceil$.*

As presented above, this algorithm needs to store integers of unbounded size since $\text{Geo}(2/\varepsilon)$ is unbounded in magnitude. As noted in [14], by restricting the generated noise to a fixed range we can avoid this problem. However, even when the generated noise is restricted to a fixed range, generating this noise via inverse transform sampling may require infinite precision. By appropriately choosing ε, the probabilities of this noise's cumulative distribution function can be represented with finite precision, and therefore generating this noise via inverse transform sampling only requires finite precision.

▶ **Theorem 4.5.** *Let $n \in \mathbb{N}_+$, $\varepsilon \in \mathbb{N}^{-1}$ and $\tilde{\varepsilon} = 2 \cdot \ln\left(1 + 2^{-\lceil \log(2/\varepsilon) \rceil}\right) \in (4/9 \cdot \varepsilon, \varepsilon]$. Then there is a deterministic algorithm $\texttt{GeoSample}_{n,\varepsilon} : [n] \times [d]_+ \to [n]$ where $\log d = O(n \cdot \log(1/\varepsilon))$ with the following properties:*

- $\texttt{GeoSample}_{n,\varepsilon}(c, U) \sim \texttt{GeometricMechanism}_{n,\tilde{\varepsilon}}(c, 1)$ *where $U \sim \text{Unif}([d]_+)$ for all $c \in [n]$. Thus, $\texttt{GeoSample}_{n,\varepsilon}$ is $(\tilde{\varepsilon}/2, 0)$-differentially private for counting queries and has (a, β)-accuracy for $\beta \in (0, 1)$ and $a = \lceil (2/\tilde{\varepsilon}) \cdot \ln(1/\beta) \rceil$.*
- $\texttt{GeoSample}_{n,\varepsilon}$ *has running time $\tilde{O}(n \cdot \log(1/\varepsilon))$.*
- *For all $z \in [n]$, $F_{\texttt{GeoSample}_{n,\varepsilon}}(z)$ can be computed in time $\tilde{O}(n \cdot \log(1/\varepsilon))$.*

4.2 Approximating Geometric Noise to Release Counting Queries Faster

Notice that $\texttt{GeoSample}_{n,\varepsilon}$ has running time at least linear in n. This is due to evaluating a (scaled) cumulative distribution function operating on integers with bit length $\Omega(n)$. We can improve the running time by approximately sampling from a two-sided geometric distribution. Small tail probabilities are dropped to reduce the number of required bits to represent probabilities to logarithmic in n. And then to recover pure differential privacy, following Lemma 3.1, we mix with uniformly random output.

▶ **Theorem 4.6.** *Let $n \in \mathbb{N}+$, $\varepsilon, \gamma \in \mathbb{N}^{-1}$ and $\tilde{\varepsilon} = 2 \cdot \ln(1 + 2^{-\lceil \log(2/\varepsilon) \rceil}) \in (4/9 \cdot \varepsilon, \varepsilon]$. Then there is a deterministic algorithm $\texttt{FastSample}_{n,\varepsilon,\gamma} : [n] \times [d]_+ \to [n]$ where $\log d = \tilde{O}(1/\varepsilon) \cdot \log(n/\gamma)$ with the following properties:*

- $\texttt{FastSample}_{n,\varepsilon,\gamma}$ *is $(\varepsilon/2, 0)$-differentially private for counting queries.*
- *For every $\beta > \gamma$, $\texttt{FastSample}_{n,\varepsilon,\gamma}$ has (a, β)-accuracy for $a = \lceil (2/\tilde{\varepsilon}) \cdot \ln(1/(\beta - \gamma)) \rceil$.*

- $\texttt{FastSample}_{n,\varepsilon,\gamma}$ *has running time* $\tilde{O}((1/\varepsilon)\cdot\log^2 n + (1/\varepsilon)\cdot\log n\cdot\log(1/\gamma))$.
- *For all* $z\in[n]$, $F_{\texttt{FastSample}_{n,\varepsilon,\gamma}}(z)$ *can be computed in time* $\tilde{O}((1/\varepsilon)\cdot\log(n/\gamma))$.

5 Generalization of the Laplace Mechanism

As shown by Dwork, McSherry, Nissim and Smith [11], we can privately release a histogram by adding independent and appropriately scaled Laplace noise to each bin. Below we state a generalization guaranteeing privacy provided the counting query algorithm used is private and the released counts are independent.

Algorithm 3 $\texttt{BasicHistogram}_{\mathcal{M},A}(D)$ for $D\in\mathcal{X}^n$, $\mathcal{M}:[n]\times[d]_+\to[n]$ and $A\subseteq\mathcal{X}$

1. Compute $c_x(D)$ for all $x\in A$.
2. For each $x\in A$, do the following:
   a. Sample u_x uniformly at random from $[d]_+$.
   b. Let $\tilde{c}_x=\mathcal{M}(c_x(D),u_x)$.
   c. Release $(x,\tilde{c}_x)$.

The output of this algorithm is a collection of bins $(x,\tilde{c}_x)$ representing a partial vector.

▶ **Theorem 5.1.** *Let* $A\subseteq\mathcal{X}$ *and* $\mathcal{M}:[n]\times[d]_+\to[n]$ *be* $(\varepsilon/2,0)$*-differentially private for counting queries and have* (a,β)*-accuracy. Then* $\texttt{BasicHistogram}_{\mathcal{M},A}:\mathcal{X}^n\to\mathbb{N}^A$ *has the following properties:*

- $\texttt{BasicHistogram}_{\mathcal{M},A}$ *is* $(\varepsilon,0)$*-differentially private.*
- *For all* $D\in\mathcal{X}^n$, *we have*

$$\forall x\in A\quad \Pr[|(\texttt{BasicHistogram}_{\mathcal{M},A}(D))_x - c_x(D)|\le a]\ge 1-\beta$$

  In particular, $\texttt{BasicHistogram}_{\mathcal{M},\mathcal{X}}(D)$ *has* (a,β)*-per-query accuracy.*
- *For all* $D\in\mathcal{X}^n$, *we have*

$$\Pr[\forall x\in A\quad |(\texttt{BasicHistogram}_{\mathcal{M},A}(D))_x - c_x(D)|\le a]\ge 1-|A|\cdot\beta$$

  In particular, $\texttt{BasicHistogram}_{\mathcal{M},\mathcal{X}}$ *has* $(a,|\mathcal{X}|\cdot\beta)$*-simultaneous accuracy.*
- *Running time* $O(n\log n\cdot\log|\mathcal{X}| + |A|\cdot(\log n\cdot\log|\mathcal{X}| + \log d + \mathrm{Time}(\mathcal{M})))$.

It is important to note that the privacy guarantee only holds when A is fixed and does not depend on the dataset D. The choice of parameterizing by A will be convenient in defining more complex histogram algorithms later.

$\mathcal{M}$	Running Time	(a,β)-Per-Query	(a,β)-Simul.
`GeometricMechanism`	n/a	$\left\lceil\frac{2}{\varepsilon}\ln\frac{1}{\beta}\right\rceil$	$\left\lceil\frac{2}{\varepsilon}\ln\frac{\lvert\mathcal{X}\rvert}{\beta}\right\rceil$
`GeoSample`	$\tilde{O}(\lvert\mathcal{X}\rvert\cdot n\cdot\log(1/\varepsilon))$	$\left\lceil\frac{9}{2\varepsilon}\ln\frac{1}{\beta}\right\rceil$	$\left\lceil\frac{9}{2\varepsilon}\ln\frac{\lvert\mathcal{X}\rvert}{\beta}\right\rceil$
`FastSample`	$\tilde{O}\left((\lvert\mathcal{X}\rvert/\varepsilon)\cdot\log^2 n\right)+\tilde{O}(n)\cdot\log\lvert\mathcal{X}\rvert$	$\left\lceil\frac{9}{2\varepsilon}\ln\frac{2}{\beta}\right\rceil$	$\left\lceil\frac{9}{2\varepsilon}\ln\frac{2\lvert\mathcal{X}\rvert}{\beta}\right\rceil$

Figure 1 The running time and errors of $\texttt{BasicHistogram}_{\mathcal{M},\mathcal{X}}$ for the counting query algorithms of Section 4. Values shown are for a $(\varepsilon,0)$-differentially private release. For `FastSample`, we take $\gamma=\beta/(2|\mathcal{X}|)$ and assume $\beta\ge 1/n^{O(1)}$.

By taking $\mathcal{M} = \texttt{GeometricMechanism}$, $\texttt{BasicHistogram}_{\mathcal{M},\mathcal{X}}$ is identically distributed to the Truncated Geometric Mechanism of Ghosh, Roughgarden and Sundararajan [14] which achieves per-query and simultaneous accuracy with error up to constant factors matching known lower bounds for releasing a private histogram [17, 2].

6 Improving the Running Time

In this section, we present an algorithm whose running time depends only poly-logarithmically on the universe size while maintaining pure differential privacy based on the observation that most counts are 0 when $n \ll |\mathcal{X}|$; this is the same observation made by Cormode, Procopiuc, Srivastava and Tran [9] to release private histograms that are sparse in expectation.

6.1 Sparse Histograms

We start by reducing the output length of $\texttt{BasicHistogram}_{\mathcal{M},\mathcal{X}}$ to release only the bins with the heaviest (or largest) counts (interpreted as a partial vector).

Algorithm 4 $\texttt{KeepHeavy}_{\mathcal{M}}(D)$ for $D \in \mathcal{X}^n$ where $\mathcal{M} : [n] \times [d]_+ \to [n]$

1. Let $\{(x, \tilde{c}_x)\}_{x\in\mathcal{X}} = \texttt{BasicHistogram}_{\mathcal{M},\mathcal{X}}(D)$.
2. Let $x_1, \ldots, x_{n+1}$ be the elements of $\mathcal{X}$ with the largest counts in sorted order, i.e.

$$\tilde{c}_{x_1} \geq \tilde{c}_{x_2} \geq \ldots \geq \tilde{c}_{x_{n+1}} \geq \max_{x\in\mathcal{X}\setminus\{x_1,\ldots,x_{n+1}\}} \tilde{c}_x$$

3. Release $h = \{(x, \tilde{c}_x) : x \in \mathcal{X} \text{ and } \tilde{c}_x > \tilde{c}_{x_{n+1}}\} \in \mathcal{H}_{n,n}(\mathcal{X})$.

Observe that the output length has been improved to $O(n \cdot (\log|\mathcal{X}| + \log n))$ bits compared to the $O(|\mathcal{X}| \cdot (\log|\mathcal{X}| + \log n))$ bits needed to represent the outputs of $\texttt{BasicHistogram}_{\mathcal{M},\mathcal{X}}$.

▶ **Theorem 6.1.** *Let $\mathcal{M} : [n] \times [d]_+ \to [n]$ be $(\varepsilon/2, 0)$-differentially private for counting queries such that $\texttt{BasicHistogram}_{\mathcal{M},\mathcal{X}}$ has (a_1, β)-per-query accuracy and (a_2, β)-simultaneous accuracy with $a_1 \leq a_2$. Then $\texttt{KeepHeavy}_{\mathcal{M}} : \mathcal{X}^n \to \mathcal{H}_{n,n}(\mathcal{X})$ has the following properties:*

- *$(\varepsilon, 0)$-differential privacy.*
- *$(a_1, 2\beta)$-per-query accuracy on counts larger than $2a_2$.*
- *$(2a_2, \beta)$-simultaneous accuracy.*

Unlike $\texttt{BasicHistogram}_{\mathcal{M},\mathcal{X}}$, by taking $\mathcal{M} = \texttt{GeoSample}$, the algorithm $\texttt{KeepHeavy}_{\mathcal{M}}$ achieves $(O(\log(1/\beta)/\varepsilon), \beta)$-per-query accuracy only on counts larger than $O(\log(|\mathcal{X}|/\beta)/\varepsilon)$. This loss is necessary for any algorithm that outputs a sparse histogram by Theorem 1.5.

However, as described KeepHeavy still requires adding noise to the count of every bin. The following algorithm $\texttt{KH}' : \mathcal{X}^n \to \mathcal{H}_{n,n}(\mathcal{X})$ simulates KeepHeavy by generating a candidate set of heavy bins from which only the heaviest are released. This candidate set is constructed from all bins with nonzero true count and a sample representing the bins with a true count of 0 that have the heaviest noisy counts.

Algorithm 5 $\texttt{KH}'_{\mathcal{M}}(D)$ for $D \in \mathcal{X}^n$ where $\mathcal{M} : [n] \times [d]_+ \to [n]$ and $|\mathcal{X}| \geq 2n + 1$[2]

1. Let $A = \{x \in \mathcal{X} : c_x(D) > 0\}$ and $m = |\mathcal{X} \setminus A|$.
2. Let $\{(x, \tilde{c}_x)\}_{x \in A} = \texttt{BasicHistogram}_{\mathcal{M},A}(D)$.
3. Pick a uniformly random sequence $(q_0, \ldots, q_n)$ of distinct elements from $\mathcal{X} \setminus A$.
4. Sample $(\tilde{c}_{q_0}, \ldots, \tilde{c}_{q_n})$ from the joint distribution of the order statistics $(Z_{(m)}, \ldots, Z_{(m-n)})$ where $Z_1, \ldots, Z_m$ are i.i.d. $\mathcal{M}(0, U)$ random variables with $U \sim \text{Unif}([d]_+)$.
5. Sort the elements of $A \cup \{q_0, \ldots, q_n\}$ as $x_1, \ldots, x_{|A|+n+1}$ such that $\tilde{c}_{x_1} \geq \ldots \geq \tilde{c}_{x_{|A|+n+1}}$.
6. Release $h = \{(x, \tilde{c}_x) : x \in \{x_1, \ldots, x_n\} \text{ and } \tilde{c}_x > \tilde{c}_{x_{n+1}}\} \in \mathcal{H}_{n,n}(\mathcal{X})$.[3]

▶ **Proposition 6.2.** $\texttt{KH}'_{\mathcal{M}}(D)$ *is identically distributed to* $\texttt{KeepHeavy}_{\mathcal{M}}(D)$.

In order to sample from the order statistics used by $\texttt{KH}'_{\mathcal{M}}$ we construct an algorithm using inverse transform sampling similar to the counting query algorithms of Section 4.

▶ **Proposition 6.3.** *Let $n, d \in \mathbb{N}_+$ and $F : [n] \to [d]_+$ such that F is non-decreasing and $F(n) = d$. Let $m \in \mathbb{N}_+$ such that $m \geq n + 1$. Let $Z_1, \ldots, Z_m$ be i.i.d. random variables over $[n]$ with cumulative distribution function $F(z)/d$ for all $z \in [n]$. Then the following algorithm $\texttt{OrdSample}_F(m)$ is identically distributed to the top $n+1$ order statistics $(Z_{(m)}, \ldots, Z_{(m-n)})$.*

Also, $\texttt{OrdSample}_F(m)$ has running time $O(n \log n) \cdot \big(\tilde{O}(m \cdot \log d) + \text{Time}(F)\big)$.

Algorithm 6 $\texttt{OrdSample}_F(m)$ for $m \in \mathbb{N}_+$ such that $m \geq n + 1$ where $F : [n] \to [d]_+$

1. Let $v_{-1} = n$.
2. For $i \in [n]$, do the following:
   a. Sample u_i uniformly at random from $[F(v)^{m-i}]_+$.
   b. Using binary search find the smallest $z \in [v_{i-1}]$ such that $F(z)^{m-i} \geq u_i$. Call it v_i.
3. Return $(v_0, \ldots v_n)$.

Now from $\texttt{KH}'$ we replace sampling from the joint distribution of the order statistics with the explicit sampling algorithm $\texttt{OrdSample}$ to get the following algorithm.

[2] $|\mathcal{X}| \geq 2n + 1$ ensures that $|\mathcal{X} \setminus A| \geq n + 1$. One can use $\texttt{BasicHistogram}_{\mathcal{X}}(D, \mathcal{M})$ when $|\mathcal{X}| \leq 2n$.

[3] If instead $\mathcal{M}$ had real-valued range this last step is equivalent to releasing the n heaviest bins. However, in the discrete case, where ties can occur, from the set $A \cup \{x_1, \ldots, x_n\}$ we cannot determine all bins with a count tied for the n-th heaviest as there may be many other noisy counts tied with $\tilde{c}_{x_n}$. As a result, we only output the bins with a strictly heavier count than $\tilde{c}_{x_{n+1}}$.

Algorithm 7 $\texttt{KH}''_{\mathcal{M},\beta_1}(D)$ for $D \in \mathcal{X}^n$, $\mathcal{M} : [n] \times [d]_+ \to [n]$, $\beta_1 \in \mathbb{N}^{-1}$ and $|\mathcal{X}| \geq 4n$

1. Construct a sequence Q of $\lceil (4n+2)/\beta_1 \rceil$ elements sampled uniformly at random from $\mathcal{X}$. If Q has less than $2n+1$ distinct elements release an empty histogram and stop.[4]
2. Let $A = \{x \in \mathcal{X} : c_x(D) > 0\}$ and $m = |\mathcal{X} \setminus A|$.
3. Let $\{(x, \tilde{c}_x)\}_{x \in A} = \texttt{BasicHistogram}_{\mathcal{M},A}(D)$.
4. Let $(q_0, \ldots, q_n)$ be the first $n+1$ distinct elements of Q not in A.
5. Let $(\tilde{c}_{q_0}, \ldots, \tilde{c}_{q_n}) = \texttt{OrdSample}_{F_{\mathcal{M}}}(m)$.
6. Sort the elements of $A \cup \{q_0, \ldots, q_n\}$ as $x_1, \ldots, x_{|A|+n+1}$ such that $\tilde{c}_{x_1} \geq \ldots \geq \tilde{c}_{x_{|A|+n+1}}$.
7. Release $h = \{(x, \tilde{c}_x) : x \in \{x_1, \ldots, x_n\} \text{ and } \tilde{c}_x > \tilde{c}_{x_{n+1}}\} \in \mathcal{H}_{n,n}(\mathcal{X})$.

▶ **Theorem 6.4.** *Let deterministic $\mathcal{M} : [n] \times [d]_+ \to [n]$ be $(\varepsilon/2, 0)$-differentially private for counting queries such that $\texttt{BasicHistogram}_{\mathcal{X}}(D, \mathcal{M})$ has (a_1, β_2)-per-query accuracy and (a_2, β_2)-simultaneous accuracy with $a_1 \leq a_2$. Then $\texttt{KH}''_{\mathcal{M},\beta_1} : \mathcal{X}^n \to \mathcal{H}_{n,n}(\mathcal{X})$ has the following properties:*

- *$(\varepsilon, 0)$-differential privacy.*
- *$(a_1, \beta_1 + 2\beta_2)$-per-query accuracy on counts larger than $2a_2$.*
- *$(2a_2, \beta_1 + \beta_2)$-simultaneous accuracy.*

This algorithm only has an output of length $O(n \cdot (\log |\mathcal{X}| + \log n))$. However, its running time depends polynomially on $|\mathcal{X}|$ since sampling the m^{th} order statistic, $\tilde{c}_{q_0}$, using $\texttt{OrdSample}$ takes time at least linear in $m \geq |\mathcal{X}| - n$. Indeed, this is necessary since the distribution of the order statistic $Z_{(m)}$ has probabilities that are exponentially small in m.

6.2 An Efficient Approximation

To remedy the inefficiency of $\texttt{KH}''$ we consider an efficient algorithm that approximates the output distribution of $\texttt{KH}''$.

▶ **Theorem 6.5.** *Let $\beta_1, \delta \in \mathbb{N}^{-1}$ and $\mathcal{M} : [n] \times [d]_+ \to [n]$ be $(\varepsilon/2, 0)$-differentially private for counting queries. Then there exists an algorithm $\texttt{SparseHistogram}_{\mathcal{M},\beta_1,\delta} : \mathcal{X}^n \to \mathcal{H}_{n,n}(\mathcal{X})$ with the following properties:*

- *$\Delta\big(\texttt{KH}''_{\mathcal{M},\beta_1}(D), \texttt{SparseHistogram}_{\mathcal{M},\beta_1,\delta}(D)\big) \leq \delta$ for all $D \in \mathcal{X}^n$.*
- *$\texttt{SparseHistogram}_{\mathcal{M},\beta_1,\delta}$ is $(\varepsilon, (e^{\varepsilon}+1)\cdot\delta)$-differentially private.*
- *Moreover, for $\beta_1 \geq 1/\log^{O(1)} n$ the running time of $\texttt{SparseHistogram}_{\mathcal{M},\beta_1,\delta}$ is*

$$\tilde{O}(n) \cdot \big(\log |\mathcal{X}| \cdot \tilde{O}(\log d + \log(1/\delta) + \log |\mathcal{X}|)\big) + O(n \log n) \cdot (\text{Time}(\mathcal{M}) + \text{Time}(F_{\mathcal{M}}))$$

Note that this algorithm only achieves $(\varepsilon, O(\delta))$-differential privacy. By reducing δ, the algorithm better approximates $\texttt{KH}''$, at the cost of increasing running time (polynomial in the bit length of δ). Notice that $\texttt{KH}''$ passes an argument to $\texttt{OrdSample}$ that results in $\texttt{OrdSample}$ exponentiating an integer, which represents the numerator of a fraction $a/b = F(z)/F(v)$, to a power $i \geq |\mathcal{X}| - 2n$. To improve the efficiency of $\texttt{OrdSample}$, we want to ensure that

[4] Along with step 4, this process allows us to generate $n+1$ distinct elements of $\mathcal{X} \setminus A$ with running time that has a nearly linear dependence on n (whenever $\beta_1 \geq 1/\log^{O(1)} n$) at the cost of an additive increase in failure probability. However, if we are willing to accept a nearly quadratic dependence on n, we can always sample the distinct elements instead of just with high probability.

the numbers it uses do not exceed some maximum s with bit length polynomial in n and $\log|\mathcal{X}|$. We achieve this by approximating $s \cdot (a/b)^i$ using repeated squaring and truncating each intermediate result to keep its bit length manageable. This process results in sampling the order statistics to within a statistical distance of δ.

Now, we convert $\texttt{SparseHistogram}_{\mathcal{M},\beta_1,\delta}$ to a pure differentially private algorithm by mixing it with random output following Lemma 3.1.

Algorithm 8 $\texttt{PureSparseHistogram}_{\mathcal{M},\varepsilon,\beta_1,\beta_2}(D)$ for $D \in \mathcal{X}^n$ where $\mathcal{M} : [n] \times [d]_+ \to [n]$, $\varepsilon, \beta_1, \beta_2 \in \mathbb{N}^{-1}$ and $|\mathcal{X}| \geq 4n$

1. With probability $1 - \beta_2$ release $\texttt{SparseHistogram}_{\mathcal{M},\beta_1,\delta}(D)$ with
$$\delta = \frac{\varepsilon}{3} \cdot \beta_2 \cdot \left(\frac{1}{6 \cdot |\mathcal{X}|}\right)^n$$
2. Otherwise
   a. Draw $(x_1, \ldots x_n)$ uniformly at random from $\mathcal{X}$.
   b. Let Q be the set of distinct elements from $(x_1, \ldots, x_n)$.
   c. For each $q \in Q$, sample $\tilde{c}_q$ uniformly at random from $[n]$.
   d. Release $h = \{(q, \tilde{c}_q) : q \in Q \text{ and } \tilde{c}_q > 0\} \in \mathcal{H}_{n,n}(\mathcal{X})$.

▶ **Theorem 6.6.** *Let $\varepsilon, \beta_1, \beta_2 \in \mathbb{N}^{-1}$ and deterministic $\mathcal{M} : [n] \times [d]_+ \to [n]$ be $(\varepsilon/2, 0)$-differentially private for counting queries such that $\texttt{BasicHistogram}_{\mathcal{M},\mathcal{X}}$ has (a_1, β_3)-per-query accuracy and (a_2, β_3)-simultaneous accuracy with $a_1 \leq a_2$.*

Then $\texttt{PureSparseHistogram}_{\mathcal{M},\varepsilon,\beta_1,\beta_2} : \mathcal{X}^n \to \mathcal{H}_{n,n}(\mathcal{X})$ has the following properties:

- *$(\varepsilon, 0)$-differential privacy.*
- *$(a_1, \beta_1 + 2\beta_2 + 2\beta_3)$-per-query accuracy on counts larger than $2a_2$.*
- *$(2a_2, \beta_1 + 2\beta_2 + \beta_3)$-simultaneous accuracy.*
- *For $\beta_1 \geq 1/\log^{O(1)} n$ and $\beta_2 \geq 1/O(2^n)$, the running time is*

$$\tilde{O}(n^2 \cdot \log^2|\mathcal{X}| + n \cdot \log(d/\varepsilon) \cdot \log|\mathcal{X}|) + O(n \log n) \cdot (\text{Time}(\mathcal{M}) + \text{Time}(F_{\mathcal{M}}))$$

$\mathcal{M}$	Running Time	(a,β)-Per-Query on $c_x(D) > t$	(a,β)-Simultaneous
`GeoSample`	$\tilde{O}\left(n^2 \cdot \log^2|\mathcal{X}|\right)$	$\left\lceil \frac{9}{2\varepsilon} \ln \frac{4}{\beta} \right\rceil$	$2 \cdot \left\lceil \frac{9}{2\varepsilon} \ln \frac{4\lvert\mathcal{X}\rvert}{\beta} \right\rceil$	$2 \cdot \left\lceil \frac{9}{2\varepsilon} \ln \frac{2\lvert\mathcal{X}\rvert}{\beta} \right\rceil$
`FastSample`	$\tilde{O}\left(n^2 \cdot \log^2|\mathcal{X}|\right)$	$\left\lceil \frac{9}{2\varepsilon} \ln \frac{8}{\beta} \right\rceil$	$2 \cdot \left\lceil \frac{9}{2\varepsilon} \ln \frac{8\lvert\mathcal{X}\rvert}{\beta} \right\rceil$	$2 \cdot \left\lceil \frac{9}{2\varepsilon} \ln \frac{4\lvert\mathcal{X}\rvert}{\beta} \right\rceil$

Figure 2 The running time and errors of $\texttt{SparseHistogram}_{\mathcal{M},\varepsilon,\beta/6,\beta/6}(D)$ for the counting query algorithms of Section 4. For per-query accuracy, the first value is the error a and the second value is the threshold t. Values shown are for a $(\varepsilon, 0)$-differentially private release. We assume $\varepsilon \geq 1/O(n)$ and $\beta \geq 1/\log^{O(1)} n$. For `FastSample`, we take $\gamma = \beta/(4|\mathcal{X}|)$.

7 Better Per-Query Accuracy via Compact, Non-Sparse Representations

In this section, we present a histogram algorithm whose running time is poly-logarithmic in $|\mathcal{X}|$, but, unlike Algorithm 8, is able to achieve $(O(\log(1/\beta)/\varepsilon), \beta)$-per query accuracy. It

will output a histogram from a properly chosen family of succinctly representable histograms. This family necessarily contains histograms that have many nonzero counts to avoid the lower bound of Theorem 1.5.

7.1 The Family of Histograms

We start by defining this family of histograms.

▶ **Lemma 7.1.** *Let $\mathcal{M}_0 : [d_0]_+ \to [n]$, $d_0 = 2^{2\cdot 3^\ell}$ for some $\ell \in \mathbb{N}$, $d_0 \geq |\mathcal{X}|$ and $U \sim \mathrm{Unif}([d_0]_+)$. There exists a multiset of histograms $\mathcal{G}_{\mathcal{M}_0}(\mathcal{X})$ satisfying:*

- *Let $g \sim \mathrm{Unif}(\mathcal{G}_{\mathcal{M}_0}(\mathcal{X}))$. For all $x \in \mathcal{X}$, the marginal distribution g_x is distributed according to $\mathcal{M}_0(U)$.*
- *Let $g \sim \mathrm{Unif}(\mathcal{G}_{\mathcal{M}_0}(\mathcal{X}))$. For all $B \subseteq \mathcal{X}$ such that $|B| \leq n+1$ and for all $c \in [n]^B$*

$$\Pr[\forall x \in B \;\; g_x = c_x] = \prod_{x \in B} \Pr[g_x = c_x]$$

- *For all $g \in \mathcal{G}_{\mathcal{M}_0}(\mathcal{X})$, the histogram g can be represented by a string of length $O(n \cdot \log d_0)$ and given this representation for all $x \in \mathcal{X}$ the count g_x can be evaluated in time $O(n) \cdot \tilde{O}(\log d_0) + \mathrm{Time}(\mathcal{M}_0)$.*
- *For all $A \subseteq \mathcal{X}$ such that $|A| \leq n$ and $c \in [n]^A$ sampling a histogram h uniformly at random from $\{g \in \mathcal{G}_{\mathcal{M}_0}(\mathcal{X}) : \forall x \in A \;\; g_x = c_x\}$ can be done in time $O(n) \cdot \mathrm{Time}(\mathcal{S}) + \tilde{O}(n \cdot \log^2 d_0)$ where $\mathrm{Time}(\mathcal{S})$ is the maximum time over $v \in [n]$ to sample from the distribution $\mathcal{S}_v \sim \mathrm{Unif}(\{u_0 \in [d_0]_+ : \mathcal{M}_0(u_0) = v\})$.*

Proof. (Construction) Let $\mathcal{G}'_{\mathcal{M}_0}(\mathcal{X})$ be the set of all degree at most n polynomials over the finite field $\mathbb{F}_{d_0}$. Now, $\mathcal{G}'_{\mathcal{M}_0}(\mathcal{X})$ is a $(n+1)$-wise independent hash family mapping $\mathbb{F}_{d_0}$ to $\mathbb{F}_{d_0}$. And given any function $p_g \in \mathcal{G}'_{\mathcal{M}_0}(\mathcal{X})$ we construct a histogram $g \in \mathcal{G}_{\mathcal{M}_0}(\mathcal{X})$ by using $p_g(x)$ as the randomness for $\mathcal{M}_0$. More specifically, let $T : \mathbb{F}_{d_0} \to [d_0]_+$ be a bijection and for all $x \in \mathcal{X}$, define $g_x = \mathcal{M}_0(T(p_g(x)))$. ◀

7.2 The Algorithm

We can think of taking $d_0 = d$ and $\mathcal{M}_0(u) = \mathcal{M}(0, u)$ for all $u \in [d]_+$, but for technical reasons (e.g. requiring $d_0 \geq |\mathcal{X}|$), we will allow $\mathcal{M}_0(u)$ to approximate $\mathcal{M}(0, u)$. In this way, a histogram picked uniformly at random from the family $\mathcal{G}_{\mathcal{M}_0}(\mathcal{X})$ will have the desired marginal distributions for all empty bins.

Thus, for our algorithm to have the correct marginal distributions over all bins we first compute the noisy counts for the nonzero bins and then randomly pick a histogram from our family that is consistent with these computed counts.

Algorithm 9 $\texttt{CompactHistogram}_{\mathcal{M},\mathcal{M}_0}(D)$ for $D \in \mathcal{X}^n$ where $\mathcal{M} : [n] \times [d]_+ \to [n]$ and $\mathcal{M}_0 : [d_0]_+ \to [n]$ such that $d_0 = 2^{2\cdot 3^\ell}$ for some $\ell \in \mathbb{N}$ and $d_0 \geq |\mathcal{X}|$

1. Let $A = \{x \in \mathcal{X} : c_x(D) > 0\}$.
2. Let $\{(x, \tilde{c}_x)\}_{x \in A} = \texttt{BasicHistogram}_{\mathcal{M},A}(D)$.
3. Release h drawn uniformly at random from $\{g \in \mathcal{G}_{\mathcal{M}_0}(\mathcal{X}) : \forall x \in A \;\; g_x = \tilde{c}_x\}$.

▶ **Theorem 7.2.** *Let deterministic* $\mathcal{M} : [n] \times [d]_+ \to [n]$ *be* $(\varepsilon_1/2, 0)$*-differentially private for counting queries and have* (a, β)*-accuracy. Let deterministic* $\mathcal{M}_0 : [d_0]_+ \to [n]$ *such that* $d_0 = 2^{2 \cdot 3^\ell}$ *for some* $\ell \in \mathbb{N}$ *and* $d_0 \geq |\mathcal{X}|$*. Assume* $\Pr[\mathcal{M}_0(U_0) \leq a] \geq 1 - \beta$ *and for all* $c \in [n]$

$$e^{-\varepsilon_2} \cdot \Pr[\mathcal{M}_0(U_0) = c] \leq \Pr[\mathcal{M}(0, U) = c] \leq e^{\varepsilon_3} \cdot \Pr[\mathcal{M}_0(U_0) = c]$$

where $U \sim \mathrm{Unif}([d]_+)$ *and* $U_0 \sim \mathrm{Unif}([d_0]_+)$*. Then* $\texttt{CompactHistogram}_{\mathcal{M}, \mathcal{M}_0}(D)$ *has the following properties:*

- $(\varepsilon_1 + \varepsilon_2 + \varepsilon_3, 0)$*-differential privacy.*
- (a, β)*-per-query accuracy.*
- $(a, |\mathcal{X}| \cdot \beta)$*-simultaneous accuracy.*
- *Running time* $\tilde{O}(n \cdot \log d + n \cdot \mathrm{Time}(\mathcal{M}) + n \cdot \mathrm{Time}(\mathcal{S}) + n \cdot \log^2 d_0)$ *where* $\mathrm{Time}(\mathcal{S})$ *is the maximum time over* $v \in [n]$ *to sample from the distribution* $\mathcal{S}_v \sim \mathrm{Unif}(\{u_0 \in [d_0]_+ : \mathcal{M}_0(u_0) = v\})$*.*
- *Given its output* h*, the count* h_x *can be evaluated in time* $O(n) \cdot \tilde{O}(\log d_0) + \mathrm{Time}(\mathcal{M}_0)$ *for all* $x \in \mathcal{X}$*.*

As discussed above, a natural choice for $\mathcal{M}_0$ is to take $d_0 = d$ and $\mathcal{M}_0(u) = \mathcal{M}(0, u)$ for all $u \in [d]_+$. Although d does not does satisfy the required constraints for the counting algorithms of Section 4, as we show in the full version of the paper [1], we can construct $\mathcal{M}_0$ for our counting query algorithms at only a constant loss in privacy.

▶ **Corollary 7.3.** *Let* $\varepsilon \in \mathbb{N}^{-1}$*,* $\beta \in \mathbb{N}^{-1}$ *such that* $\beta \geq 1/n^{O(1)}$ *and* $\mathcal{M} = \texttt{FastSample}_{n, \varepsilon', \gamma}$ *where* $\varepsilon' = 1/\lceil 10/(9\varepsilon) \rceil$ *and* $\gamma = \beta/(2|\mathcal{X}|)$*. Then there exists* $\mathcal{M}_0 : [d_0]_+ \to [n]$ *where* $\log d_0 = \tilde{O}(1/\varepsilon) \cdot (\log n + \log |\mathcal{X}|)$ *such that* $\texttt{CompactHistogram}_{\mathcal{M}, \mathcal{M}_0}$ *has the following properties:*

- $(\varepsilon, 0)$*-differential privacy.*
- (a, β)*-per-query accuracy for* $a = \lceil (5/\varepsilon) \cdot \ln(2/\beta) \rceil$*.*
- (a, β)*-simultaneous accuracy for* $a = \lceil (5/\varepsilon) \cdot \ln(2|\mathcal{X}|/\beta) \rceil$*.*
- *Running time* $O(n) \cdot \tilde{O}((1/\varepsilon^2) \cdot (\log^2 n + \log^2 |\mathcal{X}|))$*.*
- *Given its output, a count can be computed in time* $O(n) \cdot \tilde{O}((1/\varepsilon) \cdot (\log n + \log |\mathcal{X}|))$*.*

Acknowledgements. We thank the Harvard Privacy Tools differential privacy research group, particularly Mark Bun and Kobbi Nissim, for informative discussions and feedback. And we thank Ashwin Machanavajjhala, Frank McSherry, Uri Stemmer, and the anonymous TPDP and ITCS reviewers for their helpful comments.

References

1 Victor Balcer and Salil P. Vadhan. Differential privacy on finite computers. *CoRR*, abs/1709.05396, 2017. arXiv:1709.05396.

2 Amos Beimel, Hai Brenner, Shiva Prasad Kasiviswanathan, and Kobbi Nissim. Bounds on the sample complexity for private learning and private data release. *Machine learning*, 94(3):401–437, 2014.

3 Avrim Blum, Katrina Ligett, and Aaron Roth. A learning theory approach to noninteractive database privacy. *J. ACM*, 60(2):12:1–12:25, 2013. doi:10.1145/2450142.2450148.

4 Mark Bun, Kobbi Nissim, and Uri Stemmer. Simultaneous private learning of multiple concepts. In Madhu Sudan, editor, *Proceedings of the 2016 ACM Conference on Innovations in Theoretical Computer Science, Cambridge, MA, USA, January 14-16, 2016*, pages 369–380. ACM, 2016. doi:10.1145/2840728.2840747.

5 Bryan Cai, Constantinos Daskalakis, and Gautam Kamath. Priv'it: Private and sample efficient identity testing. *CoRR*, abs/1703.10127, 2017. arXiv:1703.10127.

6 Karthekeyan Chandrasekaran, Justin Thaler, Jonathan Ullman, and Andrew Wan. Faster private release of marginals on small databases. In Moni Naor, editor, *Innovations in Theoretical Computer Science, ITCS'14, Princeton, NJ, USA, January 12-14, 2014*, pages 387–402. ACM, 2014. doi:10.1145/2554797.2554833.

7 Mahdi Cheraghchi, Adam Klivans, Pravesh Kothari, and Homin K. Lee. Submodular functions are noise stable. In *Proceedings of the Twenty-third Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '12, pages 1586–1592, Philadelphia, PA, USA, 2012. Society for Industrial and Applied Mathematics.

8 Thomas H Cormen. *Introduction to algorithms*. MIT press, 2009.

9 Graham Cormode, Cecilia M. Procopiuc, Divesh Srivastava, and Thanh T. L. Tran. Differentially private summaries for sparse data. In Alin Deutsch, editor, *15th International Conference on Database Theory, ICDT '12, Berlin, Germany, March 26-29, 2012*, pages 299–311. ACM, 2012. doi:10.1145/2274576.2274608.

10 Cynthia Dwork, Krishnaram Kenthapadi, Frank McSherry, Ilya Mironov, and Moni Naor. Our data, ourselves: Privacy via distributed noise generation. In *Eurocrypt*, volume 4004, pages 486–503. Springer, 2006.

11 Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *TCC*, volume 3876, pages 265–284. Springer, 2006.

12 Cynthia Dwork, Aleksandar Nikolov, and Kunal Talwar. Efficient algorithms for privately releasing marginals via convex relaxations. *Discrete & Computational Geometry*, 53(3):650–673, 2015. doi:10.1007/s00454-015-9678-x.

13 Ivan Gazeau, Dale Miller, and Catuscia Palamidessi. Preserving differential privacy under finite-precision semantics. In Luca Bortolussi and Herbert Wiklicky, editors, *Proceedings 11th International Workshop on Quantitative Aspects of Programming Languages and Systems, QAPL 2013, Rome, Italy, March 23-24, 2013.*, volume 117 of *EPTCS*, pages 1–18, 2013. doi:10.4204/EPTCS.117.1.

14 Arpita Ghosh, Tim Roughgarden, and Mukund Sundararajan. Universally utility-maximizing privacy mechanisms. *SIAM Journal on Computing*, 41(6):1673–1693, 2012.

15 Anupam Gupta, Aaron Roth, and Jonathan Ullman. Iterative constructions and private data release. *Theory of Cryptography*, pages 339–356, 2012.

16 Moritz Hardt, Guy N. Rothblum, and Rocco A. Servedio. Private data release via learning thresholds. In *Proceedings of the Twenty-third Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '12, pages 168–187, Philadelphia, PA, USA, 2012. Society for Industrial and Applied Mathematics.

17 Moritz Hardt and Kunal Talwar. On the geometry of differential privacy. 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 705–714. ACM, 2010. doi:10.1145/1806689.1806786.

18 Shiva Prasad Kasiviswanathan, Homin K Lee, Kobbi Nissim, Sofya Raskhodnikova, and Adam Smith. What can we learn privately? *SIAM Journal on Computing*, 40(3):793–826, 2011.

19 Aleksandra Korolova, Krishnaram Kenthapadi, Nina Mishra, and Alexandros Ntoulas. Releasing search queries and clicks privately. In Juan Quemada, Gonzalo León, Yoëlle S. Maarek, and Wolfgang Nejdl, editors, *Proceedings of the 18th International Conference on World Wide Web, WWW 2009, Madrid, Spain, April 20-24, 2009*, pages 171–180. ACM, 2009. doi:10.1145/1526709.1526733.

20 Jacobus Hendricus van Lint. *Introduction to coding theory*. Springer, 1982.

21 Ilya Mironov. On significance of the least significant bits for differential privacy. In Ting Yu, George Danezis, and Virgil D. Gligor, editors, *the ACM Conference on Computer and Communications Security, CCS'12, Raleigh, NC, USA, October 16-18, 2012*, pages 650–661. ACM, 2012. doi:10.1145/2382196.2382264.

22 Arnold Schönhage. Schnelle multiplikation von polynomen über körpern der charakteristik 2. *Acta Informatica*, 7(4):395–398, 1977.

23 Justin Thaler, Jonathan Ullman, and Salil Vadhan. Faster algorithms for privately releasing marginals. In *International Colloquium on Automata, Languages, and Programming*, pages 810–821. Springer, 2012.

24 Jonathan Ullman and Salil P. Vadhan. Pcps and the hardness of generating private synthetic data. In *TCC*, volume 6597, pages 400–416. Springer, 2011.

25 Joachim Von Zur Gathen and Jürgen Gerhard. *Modern computer algebra*. Cambridge university press, 2013.

Finite Sample Differentially Private Confidence Intervals*†

Vishesh Karwa[1] and Salil Vadhan[2]

1 **Department of Statistics, The Ohio State University, Columbus, Ohio, USA**
karwa.8@osu.edu
2 **Center for Research on Computation & Society and School of Engineering & Applied Sciences, Harvard University, Cambridge, Massachussets, USA**
salil_vadhan@harvard.edu

Abstract

We study the problem of estimating finite sample confidence intervals of the mean of a normal population under the constraint of differential privacy. We consider both the known and unknown variance cases and construct differentially private algorithms to estimate confidence intervals. Crucially, our algorithms guarantee a finite sample coverage, as opposed to an asymptotic coverage. Unlike most previous differentially private algorithms, we do not require the domain of the samples to be bounded. We also prove lower bounds on the expected size of any differentially private confidence set showing that our the parameters are optimal up to polylogarithmic factors.

1998 ACM Subject Classification G.3 Probability and Statistics

Keywords and phrases Differential Privacy, Confidence Intervals, Lower bounds, Finite Sample

Digital Object Identifier 10.4230/LIPIcs.ITCS.2018.44

1 Overview

Differential privacy [7] is a strong and by now widely accepted definition of privacy for statistical analysis of datasets with sensitive information about individuals. While there is now a rich and flourishing body of research on differential privacy, extending well beyond theoretical computer science, the following three basic goals for research in the area have not been studied in combination with each other:

Differentially private statistical inference: The vast majority of work in differential privacy has studied how well one can approximate statistical properties of the dataset itself, i.e. empirical quantities, rather than inferring statistics of an underlying *population* from which a dataset is drawn. Since the latter is the ultimate goal of most data analysis, it should also be a more prominent object of study in the differential privacy literature.

Conservative statistical inference: An important purpose of statistical inference is to limit the chance that data analysts draw incorrect conclusions because their dataset may not accurately reflect the underlying population, for example due to the sample size being too small. For this reason, classical statistical inference also offers measures of statistical

* A full version of the paper is available at [17], https://arxiv.org/abs/1711.03908.

† This research was supported by NSF grant CNS-1237235, a Simons Investigator Award to Salil Vadhan, a grant from the Sloan Foundation, and Cooperative Agreement CB16ADR0160001 with the Census Bureau.

significance such as p-values and confidence intervals. Constructing such measures for differentially private algorithms is more complex, as one must also take into account the additional noise that is introduced for the purpose of privacy protection. For this reason, we advocate that differentially private inference procedures should be *conservative*, and err on the side of underestimating statistical significance, even at small sample sizes and for all settings of other parameters.

Rigorous analysis of the inherent price of privacy: As has been done extensively in the differential privacy literature for empirical statistics, we should also investigate the fundamental "privacy–utility tradeoffs" for (conservative) differentially private statistical inference. This involves both designing and analyzing differentially private statistical inference procedures, as well as proving negative results about the performance that can be achieved, using the best non-private procedures as a benchmark.

In this paper, we pursue all of these goals, using as a case study the problem of constructing a confidence interval for the mean of normal data. The latter is one of the most basic problems in statistical inference, yet already turns out to be nontrivial to fully understand under the constraints of differential privacy. We expect that most of our modeling and methods will find analogues for other inferential problems (e.g. hypothesis testing, Bayesian credible intervals, non-normal data, and estimating statistics other than the mean).

2 Confidence Intervals for a Normal Mean

We begin by recalling the problem of constructing a $(1-\alpha)$-level confidence interval for a normal mean without privacy. Let $X_1, \ldots, X_n$ be an independent and identically distributed (*iid*) random sample from a normal distribution with an unknown mean μ and variance σ^2. The goal is to design an estimator I that given $X_1, \ldots, X_n$, outputs an interval $I(X_1, \ldots, X_n) \subseteq \mathbb{R}$ such that

$$\mathbb{P}\left(I(X_1, \ldots, X_n) \ni \mu\right) \geq 1 - \alpha,$$

for all μ and σ. Here $1-\alpha$ is called the *coverage probability*. Given a desired coverage probability, the goal is minimize the *expected length* of the interval, namely $\mathbb{E}[|I(X_1, \ldots, X_n)|]$.

Known Variance. In the case that variance σ^2 is known (so only μ is unknown), the classic confidence interval for a normal mean is:

$$I(X_1, \ldots, X_n) = \bar{X} \pm \frac{\sigma}{\sqrt{n}} \cdot z_{1-\alpha/2},$$

where $\bar{X}$ is the sample mean and z_a represents the a^{th} quantile of a standard normal distribution.[1] It is known that this interval has the smallest expected size among all $1-\alpha$ level confidence sets for a normal mean, see for example, [20]. In this case, the length of the confidence interval is fixed and equal to

$$|I(X_1, \ldots, X_n)| = (2\sigma z_{1-\alpha/2})/\sqrt{n} = \Theta\left(\sigma\sqrt{\log(1/\alpha)/n}\right).$$

[1] The proof that this is in fact a $(1-\alpha)$-confidence interval follows by observing that $\sqrt{n} \cdot (\bar{X} - \mu)$ has a standard normal distribution, and $[-z_{1-\alpha/2}, z_{1-\alpha/2}]$ covers a $1-\alpha$ fraction of the mass of this distribution.

Unknown Variance. In the case that the variance σ^2 is unknown, the variance must be estimated from the data itself, and the classic confidence interval is:

$$I(X_1, \ldots, X_n) = \bar{X} \pm \frac{s}{\sqrt{n}} \cdot t_{n-1,1-\alpha/2},$$

where s^2 is the *sample variance* defined by

$$s^2 = \frac{1}{n-1} \sum_{i=1}^{n} (X_i - \bar{X})^2,$$

and $t_{n-1,a}$ is the a^{th} quantile of a t-distribution with $n-1$ degrees of freedom (see the appendix for definitions).[2] Now the length of the interval is a random variable with expectation

$$\mathbb{E}[|I(X_1, \ldots, X_n)|] = \frac{2\sigma}{\sqrt{n}} \cdot k_n \cdot t_{n-1,1-\alpha/2} = \Theta\left(\sigma\sqrt{\log(1/\alpha)/n}\right),$$

for an appropriate constant $k_n = 1 - O(1/n)$. (See [20].)

Relation to Hypothesis Tests. In general, including both cases above, a confidence interval for a population parameter also gives rise to hypothesis tests, which is often how the confidence intervals are used in applied statistics. For example, if our null hypothesis is that the mean μ is nonnegative, then we could reject the null hypothesis if the interval $I(X_1, \ldots, X_n)$ does not intersect the positive real line. The significance level of this hypothesis test is thus at least $1 - \alpha$. Minimizing the length of the confidence interval corresponds to being able to reject the alternate hypotheses that are closer to the null hypothesis; that is, when the confidence interval is of length at most β and μ is distance greater than β from the null hypothesis, then the test will reject with probability at least $1 - \alpha$.

3 Statistical Inference with Differential Privacy

The Laplace mechanism [7], one of the most basic differentially private algorithm, is used for estimating a function $f(\underline{x})$ of the dataset $\underline{x}$, rather than the population from which $\underline{x}$ is drawn, and much of the differential privacy literature is about estimating such empirical statistics. There are several important exceptions, the earliest being the work on differentially private PAC learning ([2, 18]), but still many basic statistical inference questions have not been addressed.

However, a natural approach for inference was already suggested in early works on differential privacy. In many cases, we know that population statistics are well-approximated by empirical statistics, and thus we can try to estimate these empirical statistics with differential privacy. For example, the population mean μ for a normal population is well-approximated by the sample mean $\overline{X}$, which we can estimate using the Laplace mechanism:

$$M(X_1, \ldots, X_n) = \overline{X} + Z, \text{where } Z \sim \text{Lap}(2B/\epsilon n).$$

On the positive side, observe that the noise being introduced for privacy vanishes linearly in $1/n$, whereas $\overline{X}$ converges to the population mean at a rate of $1/\sqrt{n}$, so asymptotically we obtain privacy "for free" compared to the (optimal) non-private estimator $\overline{X}$.

[2] Again the proof follows by observing that $s \cdot (\bar{X} - \mu)$ follows a t distribution, with no dependence on the unknown parameters.

However, this rough analysis hides some important issues. First, it is misleading to look only at the dependence on n. The other parameters, such as σ, ϵ, and B can be quite significant and should not be treated as constants. Indeed $\sigma/\sqrt{n} \gg B/\epsilon n$ only when $n \gg (B/\epsilon\sigma)^2$, which means that the asymptotics only kick in at a very large value of n. Thus it is important to determine whether the dependence on these parameters is necessary or can be improved. Second, the parameter B is supposed to be a (worst-case) bound on the range of the data, which is incompatible with a modeling the population as following a normal distribution (which is supported on the entire real line). Thus, there have been several works seeking the best asymptotic approximations we can obtain for population statistics under differential privacy, such as [6, 22, 27, 26, 12, 4, 5, 1].

4 Conservative Statistical Inference with DP

The works discussed in the previous section focus on providing point estimates for population quantities, but as mentioned earlier, it is also important to be able to provide measures of statistical significance, to prevent analysts from drawing incorrect conclusions from the results. These measures of statistical significance need to take into account the uncertainty coming both from the sampling of the data and from the noise introduced for privacy. Ignoring the noise introduced for privacy can result in wildly incorrect results at finite sample sizes, as demonstrated empirically many times (e.g. [8, 14, 15]) and this can have severe consequences. For example, [9] found that naive use of differential privacy in calculating warfarin dosage would lead to unsafe levels of medication, but of course one should never use any sort of statistics for life-or-death decisions without some analysis of statistical significance.

Since calculating the exact statistical significance of differentially private computations seems difficult in general, we advocate *conservative* estimates of significance. That is, we require $\mathbb{P}(I(X_1, \ldots, X_n) \ni \mu) \geq 1-\alpha$, for *all* values of n, values of the population parameters, and values of the privacy parameter.

For sample sizes that are too small or privacy parameters that are too aggressive, we may achieve this property by allowing the algorithm to sometimes produce an extremely large confidence interval, but that is preferable to producing a small interval that does not actually contain the true parameter which may violate the desired coverage property. Note that what constitutes a sample size that is "too small" can depend on the unknown parameters of the population (e.g. the unknown variance σ^2) and their interplay with other parameters (such as the privacy parameter ϵ).

Returning to our example of estimating a normal mean with known variance under differential privacy, if we use the Laplace Mechanism to approximate the empirical mean (as discussed above), we can obtain a conservative confidence interval for the population mean by increasing the length of classical, non-private confidence interval to account for the likely magnitude of the Laplace noise. More precisely, starting with the differentially private mechanism

$$M(X_1, \ldots, X_n) = \overline{X} + Z, \text{where } Z \sim \text{Lap}(2B/\epsilon n),$$

the following is a $(1 - O(\alpha))$-level confidence interval for the population mean μ:

$$I(X_1, \ldots, X_n) = M(X_1, \ldots, X_n) \pm \left(\frac{\sigma}{\sqrt{n}} \cdot z_{1-\alpha/2} + \frac{B}{\epsilon n} \cdot \log(1/\alpha) \right).$$

The point is that with probability $1 - O(\alpha)$, the Laplace noise Z has magnitude at most $(B/\epsilon n) \cdot \log(1/\alpha)$, so increasing the interval by this amount will preserve coverage (up to an $O(\alpha)$ change in the probability). Again, the privacy guarantees of the Laplace mechanism

relies on the data points being guaranteed to lie in $[-B, B]$; otherwise, points need to be clamped to lie in the range, which can bias the empirical mean and compromise the coverage guarantee. Thus, to be safe, a user may choose a very large value of B, but then this makes for a much larger (and less useful) interval, as the length of the interval grows linearly with B. Thus, a natural research question (which we investigate) is whether such a choice and corresponding cost is necessary.

Conservative hypothesis testing with differential privacy, where we require that the significance level is at least $1 - \alpha$, was advocated by [11]. Methods aimed at calculating the combined uncertainty due to sampling and privacy (for various differentially private algorithms) were given in [24, 28, 14, 13, 16, 15, 11, 23, 25, 19], but generally the utility of these methods (e.g. the expected length of a confidence interval or power of a hypothesis test) is only evaluated empirically or the conservativeness only holds in a particular asymptotic regime. Rigorous, finite-sample analyses of conservative inference were given in [21] for confidence intervals on the coefficients from ordinary least-squares regression (which can be seen as a generalization of the problem we study to multivariate Gaussians) and in [3] for hypothesis testing of discrete distributions. However, neither paper provides matching lower bounds, and in particular, the algorithms of [21] only apply for bounded data (similar to the basic Laplace mechanism). In our work, we provide a comprehensive theoretical analysis of conservative differentially private confidence intervals for a normal mean, with both algorithms and lower bounds, without any bounded data assumption.

5 Our Results

As discussed above, in this paper we develop conservative differentially private estimators of confidence intervals for the mean μ of a normal distribution with known and unknown variance σ^2. Our algorithms are designed to be differentially private for all input datasets and they provide $(1 - \alpha)$-level coverage whenever the data is generated from a normal distribution. Unlike the Laplace mechanism described above and many other differentially private algorithms, we do not make any assumptions on the boundedness of the data. Our pure DP (i.e. $(\epsilon, 0)$-DP) algorithms assume that the mean μ and variance σ^2 lie in a bounded (but possibly very large) interval, and we show (using lower bounds) that such an assumption is necessary. Our approximate (i.e. (ϵ, δ)) differentially private algorithms do not make any such assumptions, i.e. both the data and the parameters (μ, σ^2) can remain unbounded. We also show that the differentially private estimators that we construct have nearly optimal expected length, up to logarithmic factors. This is done by proving lower bounds on the length of differentially private confidence intervals. A key aspect of the confidence intervals that we construct is their conservativeness — the coverage guarantee holds in finite samples, as opposed to only holding asymptotically. We also show that as $n \to \infty$, the length of our differentially private confidence intervals is at most $1 + o(1)$ factor larger than length of their non-private counterparts.

Let $X_1, \ldots, X_n$ be an independent and identically distributed (*iid*) random sample from a normal distribution with an unknown mean μ and variance σ^2, where $\mu \in (-R, R)$ and $\sigma \in (\sigma_{\min}, \sigma_{\max})$. Our goal is to construct (ϵ, δ)-differentially private $(1 - \alpha)$-level confidence sets for μ in both the known and the unknown variance case, i.e. we seek a set $I = I(X_1, \ldots, X_n)$ such that

1. $I(X_1, \ldots, X_n)$ is a $(1 - \alpha)$-level confidence interval, and
2. $I(x_1, \ldots, x_n)$ is (ϵ, δ)-differentially private.
3. $\mathop{\mathbb{E}}_{X_1,\ldots,X_n,I}[I(X_1, \ldots, X_n)]$ is as small as possible.

Known Variance: For the known variance case, we construct differentially private algorithms that output a fixed width $(1-\alpha)$-level confidence interval for any n. Moreover, when n is large enough, the algorithm outputs a confidence interval of length β which is non-trivial in the sense that $\beta \ll R$. Specifically, β is a maximum of two terms: The first term is $\mathcal{O}\left(\sigma\sqrt{\log(1/\alpha)/n}\right)$ which is the same as the length of the non-private confidence interval discussed in Section 2 up to constant factors. The second term is $\mathcal{O}\left(\sigma/(\epsilon n)\right)$ up to polylogarithmic factors − it goes to 0 at the rate of $\tilde{\mathcal{O}}\left(1/n\right)$ which is faster than the rate at which the first term goes to 0. Thus for large n the increase in the length of the confidence interval due to privacy is mild. Note that, unlike the basic approach based on the Laplace mechanism discussed in Section 4, the length of the confidence interval has no dependence on the range of the data, or even the range $(-R, R)$ of the mean μ.

The sample complexity required for obtaining a non-trivial confidence interval is the minimum of two terms: $\mathcal{O}\left((1/\epsilon)\log(R/\alpha\sigma)\right)$ and $\mathcal{O}\left((1/\epsilon)\log(1/\alpha\delta)\right)$. The dependence of sample complexity on R/σ is only logarithmic. Thus one can choose a very large value of R. Moreover, when $\delta > 0$, we can set $R = \infty$ and hence there is no dependence of the sample complexity on R.

Unknown Variance: As in the known variance case, we construct (ϵ, δ) differentially private algorithms that output an $(1-\alpha)$ confidence interval of μ for all n. If n is large enough, the length of the confidence interval is a maximum of two terms, where the first term is same as the length of the non-private confidence interval and the second term goes to 0 at a faster rate.

As before, the dependence of sample complexity on $R/\sigma_{\min}$ and $\sigma_{\max}/\sigma_{\min}$ is logarithmic, as opposed to linear. Hence we can set these parameters to a large number. Moreover, when $\delta > 0$, we can set R and $\sigma_{\max}$ to be ∞ and $\sigma_{\min}$ to be 0. Thus when $\delta > 0$, there are no assumptions on the boundedness of the parameters.

Lower Bounds: We also prove lower bounds on the length of any $(1-\alpha)$-level (ϵ, δ)-differentially private confidence set of expected size β. Our lower bounds show that one must pay $\Omega\left(\sigma/(\epsilon n) \cdot \log(1/\alpha)\right)$ in the length of the confidence interval when R is very large. Our algorithms come quite close to this lower bound with an extra factor of $\text{polylog}(n/\alpha)$. We also show that the sample complexity required by our algorithms is necessary to obtain a confidence interval that saves more than a factor of 2 over the trivial interval $(-R, R)$.

6 Directions for Future Work

The most immediate direction for future work is to close the (small) gaps between our upper and lower bounds. We came to the problem of constructing confidence intervals for a normal mean as part of an effort to bring differential privacy to practice in the sharing of social science research data through the design of the software tool PSI [10], as confidence intervals are a workhorse of data analysis in the social sciences. However, our algorithms are not optimized for practical performance, but rather for asymptotic analysis of the confidence interval length. Initial experiments indicate that alternative approaches (not just tuning of parameters) may be needed to reasonably sized confidence intervals (e.g. length at most twice that of the non-private length) handle modest sample sizes (e.g. in the 1000's). Thus designing practical differentially private algorithms for confidence intervals remains an important open problem, whose solution could have wide applicability.

As mentioned earlier, we expect that much of the modelling and techniques we develop should also be applicable more widely. In particular, it would be natural to study the estimation of other population statistics, and families of distributions, such as other continuous random variables, Bernoulli random variables, and multivariate families. In particular, a natural generalization of the problem we consider is to construct confidence intervals for the parameters of a (possibly degenerate) multivariate Gaussian, which is closely related to the problem of ordinary least-squares regression (cf. [21]).

Finally, while we have advocated for conservative inference at finite sample size, to avoid spurious conclusions coming from the introduction of privacy, many practical, non-private inference methods rely on asymptotics also for measuring statistical significance. In particular, the standard confidence interval for a normal mean with unknown variance and its corresponding hypothesis test (see Section 2) is often applied on non-normal data, and heuristically justified using the Central Limit Theorem. (This is heuristic since the rate of convergence depends on the data distribution, which is unknown.) Is there a criterion to indicate what asymptotics are "safe"? In particular, can we formalize the idea of only using the "same" asymptotics that are used without privacy? [19] analyze their hypothesis tests using asymptotics that constrain the setting of the privacy parameter in terms of the sample size n (e.g. $\epsilon \geq \Omega(1/\sqrt{n})$), but it's not clear that this relationship is safe to assume in general.

Acknowledgments. We are grateful to the Harvard Privacy Tools differential privacy research group for illuminating and motivating discussions, particularly James Honaker, Gary King, Kobbi Nissim and Uri Stemmer. We thank members of the Center for Disclosure Avoidance Research at the US Census Bureau, in particular Philip Leclerc, for carefully reading our paper and giving helpful feedback.

References

1 Rina Foygel Barber and John C Duchi. Privacy and statistical risk: Formalisms and minimax bounds. *arXiv preprint arXiv:1412.4451*, 2014.

2 Avrim Blum, Cynthia Dwork, Frank McSherry, and Kobbi Nissim. Practical privacy: the sulq framework. In *Proceedings of the twenty-fourth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 128–138. ACM, 2005.

3 Bryan Cai, Constantinos Daskalakis, and Gautam Kamath. Priv'it: Private and sample efficient identity testing. *arXiv preprint arXiv:1703.10127*, 2017.

4 John Duchi, Martin J Wainwright, and Michael I Jordan. Local privacy and minimax bounds: Sharp rates for probability estimation. In *Advances in Neural Information Processing Systems*, pages 1529–1537, 2013.

5 John C Duchi, Michael I Jordan, and Martin J Wainwright. Local privacy and statistical minimax rates. In *Foundations of Computer Science (FOCS), 2013 IEEE 54th Annual Symposium on*, pages 429–438. IEEE, 2013.

6 Cynthia Dwork and Jing Lei. Differential privacy and robust statistics. In *Proceedings of the forty-first annual ACM symposium on Theory of computing*, pages 371–380. ACM, 2009.

7 Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *TCC*, pages 265–284. Springer, 2006.

8 Stephen E. Fienberg, Alessandro Rinaldo, and Xiaolin Yang. Differential privacy and the risk-utility tradeoff for multi-dimensional contingency tables. In *Proceedings of the 2010 international conference on Privacy in statistical databases*, PSD'10, pages 187–199, Berlin,

Heidelberg, 2010. Springer-Verlag. URL: http://dl.acm.org/citation.cfm?id=1888848.1888869.

9 Matthew Fredrikson, Eric Lantz, Somesh Jha, Simon Lin, David Page, and Thomas Ristenpart. Privacy in pharmacogenetics: An end-to-end case study of personalized warfarin dosing. In *USENIX Security Symposium*, pages 17–32, 2014.

10 Marco Gaboardi, James Honaker, Gary King, Kobbi Nissim, Jonathan Ullman, and Salil Vadhan. Psi ({\Psi}): a private data sharing interface. *arXiv preprint arXiv:1609.04340*, 2016.

11 Marco Gaboardi, Hyun-Woo Lim, Ryan M. Rogers, and Salil P. Vadhan. Differentially private chi-squared hypothesis testing: Goodness of fit and independence testing. In Maria-Florina Balcan and Kilian Q. Weinberger, editors, *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*, volume 48 of *JMLR Workshop and Conference Proceedings*, pages 2111–2120. JMLR.org, 2016. URL: http://jmlr.org/proceedings/papers/v48/rogers16.html.

12 Rob Hall, Alessandro Rinaldo, and Larry Wasserman. Differential privacy for functions and functional data. *Journal of Machine Learning Research*, 14(Feb):703–727, 2013.

13 Vishesh Karwa, Dan Kifer, and Aleksandra B Slavković. Private posterior distributions from variational approximations. *arXiv preprint arXiv:1511.07896*, 2015.

14 Vishesh Karwa and Aleksandra Slavković. Differentially private graphical degree sequences and synthetic graphs. In *Privacy in Statistical Databases*, pages 273–285. Springer, 2012.

15 Vishesh Karwa and Aleksandra Slavković. Inference using noisy degrees: Differentially private β-model and synthetic graphs. *The Annals of Statistics*, 44(1):87–112, 2016.

16 Vishesh Karwa, Aleksandra B Slavković, and Pavel Krivitsky. Differentially private exponential random graphs. In *International Conference on Privacy in Statistical Databases*, pages 143–155. Springer, 2014.

17 Vishesh Karwa and Salil Vadhan. Finite sample differentially private confidence intervals. *arXiv preprint arXiv:1711.03908*, 2017.

18 Shiva Prasad Kasiviswanathan, Homin K Lee, Kobbi Nissim, Sofya Raskhodnikova, and Adam Smith. What can we learn privately? *SIAM Journal on Computing*, 40(3):793–826, 2011.

19 Daniel Kifer and Ryan Rogers. A new class of private chi-square tests. *arXiv preprint arXiv:1610.07662*, 2016.

20 Erich L Lehmann and Joseph P Romano. *Testing statistical hypotheses*. Springer Science & Business Media, 2006.

21 Or Sheffet. Differentially private ordinary least squares. In *International Conference on Machine Learning*, pages 3105–3114, 2017.

22 Adam Smith. Privacy-preserving statistical estimation with optimal convergence rates. In *Proceedings of the forty-third annual ACM symposium on Theory of computing*, pages 813–822. ACM, 2011.

23 Eftychia Solea. Differentially private hypothesis testing for normal random variables. Master's thesis, Pennsylvania State University, 2014. URL: https://etda.libraries.psu.edu/catalog/21486.

24 Duy Vu and Aleksandra Slavkovic. Differential privacy for clinical trial data: Preliminary evaluations. In *Data Mining Workshops, 2009. ICDMW'09. IEEE International Conference on*, pages 138–143. IEEE, 2009.

25 Yue Wang, Jaewoo Lee, and Daniel Kifer. Differentially private hypothesis testing, revisited. *arXiv preprint arXiv:1511.03376*, 2015.

26 Larry Wasserman. Minimaxity, statistical thinking and differential privacy. *Journal of Privacy and Confidentiality*, 4(1):3, 2012.

27 Larry Wasserman and Shuheng Zhou. A statistical framework for differential privacy. *J. Amer. Statist. Assoc.*, 105(489):375–389, 2010. doi:10.1198/jasa.2009.tm08651.

28 Oliver Williams and Frank McSherry. Probabilistic inference and differential privacy. In *Advances in Neural Information Processing Systems*, pages 2451–2459, 2010.

Resilience: A Criterion for Learning in the Presence of Arbitrary Outliers*

Jacob Steinhardt[†1], **Moses Charikar**[‡2], **and Gregory Valiant**[§3]

1 **Stanford University, Stanford, USA**
`jsteinha@stanford.edu`
2 **Stanford University, Stanford, USA**
`moses@stanford.edu`
3 **Stanford University, Stanford, USA**
`valiant@stanford.edu`

Abstract

We introduce a criterion, *resilience*, which allows properties of a dataset (such as its mean or best low rank approximation) to be robustly computed, even in the presence of a large fraction of arbitrary additional data. Resilience is a weaker condition than most other properties considered so far in the literature, and yet enables robust estimation in a broader variety of settings. We provide new information-theoretic results on robust distribution learning, robust estimation of stochastic block models, and robust mean estimation under bounded kth moments. We also provide new algorithmic results on robust distribution learning, as well as robust mean estimation in ℓ_p-norms. Among our proof techniques is a method for pruning a high-dimensional distribution with bounded 1st moments to a stable "core" with bounded 2nd moments, which may be of independent interest.

1998 ACM Subject Classification G.3 Probability and Statistics, Robust Estimation

Keywords and phrases robust learning, outliers, stochastic block models, p-norm estimation

Digital Object Identifier 10.4230/LIPIcs.ITCS.2018.45

1 Introduction

What are the fundamental properties that allow one to robustly learn from a dataset, even if some fraction of that dataset consists of arbitrarily corrupted data? While much work has been done in the setting of noisy data, or for restricted families of outliers, it is only recently that provable algorithms for learning in the presence of a large fraction of arbitrary (and potentially adversarial) data have been formulated in high-dimensional settings [14, 25, 5, 16, 24, 3]. In this work, we formulate a conceptually simple criterion that a dataset can satisfy–*resilience*–which guarantees that properties such as the mean of that dataset can be estimated even if a large fraction of additional arbitrary data is inserted.

To illustrate our setting, consider the following game between Alice (the adversary) and Bob. First, a set $S \subseteq \mathbb{R}^d$ of $(1-\epsilon)n$ points is given to Alice. Alice then adds ϵn additional points to S to create a new set $\tilde{S}$, and passes $\tilde{S}$ to Bob. Bob wishes to output a parameter

* A full version of the paper is available at `https://arxiv.org/abs/1703.04940`
† JS was supported by a Fannie & John Hertz Foundation Fellowship, an NSF Graduate Research Fellowship, and a Future of Life Instute grant.
‡ MC was supported by NSF grants CCF-1617577, CCF-1302518 and a Simons Investigator Award.
§ GV was supported by NSF CAREER Award CCF-1351108 and a Sloan Research Fellowship.

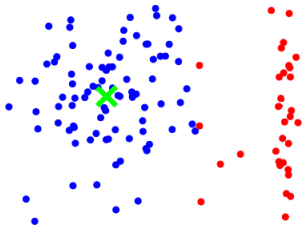


Figure 1 Illustration of the robust mean estimation setting. First, a set of points (blue) is given to Alice, who adds an ϵ fraction of adversarially chosen points (red). Bob's goal is to output the mean of the original set (indicated in green).

$\hat{\mu}$ that is as close as possible to the mean μ of the points in the original set S, with error measured according to some norm $\|\hat{\mu} - \mu\|$. The question is: how well can Bob do, assuming that Alice is an adversary with knowledge of Bob's algorithm?

The above game models mean estimation in the presence of arbitrary outliers; one can easily consider other problems as well (e.g. regression) but we focus on mean estimation here.

With no asumptions on S, Bob will clearly incur arbitrarily large error in the worst case—Alice can add points arbitrarily far away from the true mean μ, and Bob has no way of telling whether those points actually belong to S or were added by Alice. A first pass assumption is to suppose that S has diameter at most ρ; then by discarding points that are very far away from most other points, Bob can obtain error $\mathcal{O}(\epsilon\rho)$. However, in most high-dimensional settings, the diameter ρ grows polynomially with the dimension d (e.g. the d-dimensional hypercube has ℓ_2-diameter $\Theta(\sqrt{d})$). Subtler criteria are therefore needed to obtain dimension-independent bounds in most settings of interest.

Recently, [5] showed that Bob can incur ℓ_2-error $\mathcal{O}(\epsilon\sqrt{\log(1/\epsilon)})$ when the points in S are drawn from a d-dimensional Gaussian, while [16] concurrently showed that Bob can incur ℓ_2-error $\mathcal{O}(\sqrt{\epsilon \log(d)})$ if the points in S are drawn from a distribution with bounded 4th moments. Since then, a considerable amount of additional work has studied high-dimensional estimation in the presence of adversaries, which we discuss in detail below. However, in general, both Bob's strategy and its analysis tend to be quite complex, and specialized to particular distributional assumptions. This raises the question—is it possible to formulate a general and simple-to-understand criterion for the set S under which Bob has a (possibly inefficient) strategy for incurring small error?

In this paper, we provide such a criterion; we identify an assumption–*resilience*–on the set S, under which Bob has a straightforward exponential-time algorithm for estimating μ accurately. This yields new information-theoretic bounds for a number of robust learning problems, including robust learning of stochastic block models, of discrete distributions, and of distributions with bounded kth moments. We also identify additional assumptions under which Bob has an efficient (polynomial-time) strategy for estimating μ, which yields an efficient algorithm for robust learning of discrete distributions, as well as for robust mean estimation in ℓ_p-norms.

The resilience condition is essentially that the mean of every large subset of S must be close to the mean of all of S. More formally, for a norm $\|\cdot\|$, our criterion is as follows:

▶ **Definition 1** (Resilience)**.** A set of points $\{x_i\}_{i\in S}$ lying in $\mathbb{R}^d$ is (σ, ϵ)-*resilient* in a norm $\|\cdot\|$ around a point μ if, for all subsets $T \subseteq S$ of size at least $(1-\epsilon)|S|$, $\left\|\frac{1}{|T|}\sum_{i\in T}(x_i - \mu)\right\| \leq \sigma$.

More generally, a distribution p is said to be (σ, ϵ)-resilient if $\|\mathbb{E}[x - \mu \mid E]\| \leq \sigma$ for every event E of probability at least $1 - \epsilon$.

In the definition above, μ need not equal the mean of S; this distinction is useful in statistical settings where the sample mean of a finite set of points differs slightly from the true mean. However, resilience implies that μ differs from the mean of S by at most σ.

Importantly, Definition 1 is satisfied with high probability by a finite sample in many settings. For instance, samples from a distribution with kth moments bounded by σ will be $(\mathcal{O}(\sigma\epsilon^{1-1/k}), \epsilon)$-resilient in the ℓ_2-norm with high probability. Resilience also holds with high probability under many other natural distributional assumptions, discussed in more detail in Sections 1.1 and 6.

Assuming that the original set S is (σ, ϵ)-resilient, Bob's strategy is actually quite simple—find *any* large (σ, ϵ)-resilient subset S' of the corrupted set $\tilde{S}$, and output the mean of S'. By pigeonhole, S' and S have large intersection, and hence by resilience must have similar means. We establish this formally in Section 2.

Pleasingly, resilience reduces the question of whether Bob can win the game to a purely algorithmic question—that of finding any large resilient set. Rather than wondering whether it is even information-theoretically possible to estimate μ, we can instead focus on efficiently finding resilient subsets of $\tilde{S}$.

We provide one such algorithm in Section 4, assuming that the norm $\|\cdot\|$ is *strongly convex* and that we can approximately solve a certain generalized eigenvalue problem in the dual norm. When specialized to the ℓ_1-norm, our general algorithm yields an efficient procedure for robust learning of discrete distributions.

In the remainder of this section, we will outline our main results, starting with information-theoretic results and then moving on to algorithmic results. In Section 1.1, we show that resilience is indeed information-theoretically sufficient for robust mean estimation. In Section 1.2, we then provide finite-sample bounds showing that resilience holds with high probability for i.i.d. samples from a distribution.

In Section 1.3, we turn our attention to algorithmic bounds. We identity a property–bounded variance in the dual norm–under which efficient algorithms exist. We then show that, as long as the norm is strongly convex, every resilient set has a large subset with bounded variance, thus enabling efficient algorithms. This connection between resilience and bounded variance is the most technically non-trivial component of our results, and may be of independent interest.

Both our information-theoretic and algorithmic bounds yield new results in concrete settings, which we discuss in the corresponding subsections. In Section 1.4, we also discuss an extension of resilience to low-rank matrix approximation, which enables us to derive new bounds in that setting as well. In Section 1.5 we outline the rest of the paper and point to technical highlights, and in Section 1.6 we discuss related work.

1.1 Information-Theoretic Sufficiency

First, we show that resilience is indeed information-theoretically sufficient for robust recovery of the mean μ. Let $\sigma_*(\epsilon)$ denote the smallest σ such that S is (σ, ϵ)-resilient.

▶ **Proposition 2.** *Suppose that $\tilde{S} = \{x_1, \ldots, x_n\}$ contains a set S of size $(1-\epsilon)n$ that is resilient around μ (where S and μ are both unknown). Then if $\epsilon < \frac{1}{2}$, it is possible to recover a $\hat{\mu}$ such that $\|\hat{\mu} - \mu\| \leq 2\sigma_*(\frac{\epsilon}{1-\epsilon})$.*

More generally, if $|S| \geq \alpha n$ (even if $\alpha < \frac{1}{2}$), it is possible to output a (random) $\hat{\mu}$ such that $\|\hat{\mu} - \mu\| \leq \frac{16}{\alpha}\sigma_*(\frac{\alpha}{4})$ with probability at least $\frac{\alpha}{2}$.

The first part says that robustness to an ϵ fraction of outliers depends on resilience to a $\frac{\epsilon}{1-\epsilon}$ fraction of deletions. Thus, Bob has a good strategy as long as $\sigma_*(\frac{\epsilon}{1-\epsilon})$ is small.

The second part, which is more surprising, says that Bob has a good strategy *even if the majority of $\tilde{S}$ is controlled by Alice.* Here one cannot hope for recovery in the usual sense, because if $\alpha = \frac{1}{2}$ (i.e., Alice controls half the points) then Alice can make $\tilde{S}$ the disjoint union of two identical copies of S (one of which is shifted by a large amount) and Bob has no way of determining which of the two copies is the true S. Nevertheless, in this situation Bob can still identify S (and hence μ) with probability $\frac{1}{2}$; more generally, the second part of Proposition 2 says that if $|S| = \alpha|\tilde{S}|$ then Bob can identify μ with probability at least $\frac{\alpha}{2}$.

The fact that estimation is possible even when $\alpha < \frac{1}{2}$ was first established by [24] in a crowdsourcing setting, and later by [3] in a number of settings including mean estimation. Apart from being interesting due to its unexpectedness, estimation in this regime has immediate implications for robust estimation of mixtures of distributions (by considering each mixture component in turn as the "good" set S) or of planted substructures in random graphs. We refer the reader to [3] for a full elaboration of this point.

The proof of Proposition 2, given in detail in Section 2, is a pigeonhole argument. For the $\epsilon < \frac{1}{2}$ case, we simply search for any large resilient set S' and output its mean; then S and S' must have large overlap, and by resilience their means must both be close to the mean of their intersection, and hence to each other.

For the general case where $|S| = \alpha|\tilde{S}|$ (possibly with $\alpha < \frac{1}{2}$), a similar pigeonhole argument applies but we now need to consider a covering of $\tilde{S}$ by $\frac{2}{\alpha}$ approximately disjoint sets $S'_1, \ldots, S'_{2/\alpha}$. We can show that the true set S must overlap at least one of these sets by a decent amount, and so outputting the mean of one of these sets at random gives a good approximation to the mean of S with probability $\frac{\alpha}{2}$.

1.2 Finite-Sample Concentration

While Proposition 2 provides a deterministic condition under which robust mean estimation is possible, we would also like a way of checking that resilience holds with high probability given samples $x_1, \ldots, x_n$ from a distribution p. First, we provide an alternate characterization of resilience which says that a distribution is resilient if it has *thin tails* in every direction:

▶ **Lemma 3.** *Given a norm $\|\cdot\|$, define the dual norm $\|v\|_* = \sup_{\|x\| \le 1} \langle v, x \rangle$. For a fixed vector v, let $\tau_\epsilon(v)$ denote the ϵ-quantile of $\langle x - \mu, v \rangle$: $\mathbb{P}_{x \sim p}[\langle x - \mu, v \rangle \ge \tau_\epsilon(v)] = \epsilon$. Then, p is (σ, ϵ)-resilient around its mean μ if and only if*

$$\mathbb{E}_p[\langle x - \mu, v \rangle \mid \langle x - \mu, v \rangle \ge \tau_\epsilon(v)] \le \frac{1 - \epsilon}{\epsilon} \sigma \text{ whenever } \|v\|_* \le 1. \tag{1}$$

In other words, if we project onto any unit vector v in the dual norm, the ϵ-tail of $x - \mu$ must have mean at most $\frac{1-\epsilon}{\epsilon}\sigma$. Thus, for instance, a distribution with variance at most σ_0^2 along every unit vector would have $\sigma = \mathcal{O}(\sigma_0\sqrt{\epsilon})$. Note that Lemma 3 requires μ to be the mean, rather than an arbitrary vector as before.

We next provide a meta-result establishing that resilience of a population distribution p very generically transfers to a finite set of samples from that distribution. The number of samples necessary depends on two quantities B and $\log M$ that will be defined in detail later; for now we note that they are ways of measuring the effective dimension of the space.

▶ **Proposition 4.** *Suppose that a distribution p is (σ, ϵ)-resilient around its mean μ with $\epsilon < \frac{1}{2}$. Let B be such that $\mathbb{P}[\|x - \mu\| \ge B] \le \epsilon/2$. Also let M be the covering number of the unit ball in the dual norm $\|\cdot\|_*$.*

Then, given n samples $x_1, \ldots, x_n \sim p$, with probability $1 - \delta - \exp(-\epsilon n/6)$ there is a subset T of $(1-\epsilon)n$ of the x_i that is (σ', ϵ)-resilient with $\sigma' = \mathcal{O}\Big(\sigma \cdot \Big(1 + \sqrt{\frac{\log(M/\delta)}{\epsilon^2 n} + \frac{(B/\sigma)\log(M/\delta)}{n}}\Big)\Big)$.

Note that Proposition 4 only guarantees resilience on a $(1-\epsilon)n$-element subset of the x_i, rather than all of $x_1, \ldots, x_n$. From the perspective of robust estimation, this is sufficient, as we can simply regard the remaining ϵn points as part of the "bad" points controlled by Alice. This weaker requirement seems to be actually necessary to achieve Proposition 4, and was also exploited in [3] to yield improved bounds for a graph partitioning problem. There has been a great deal of recent interest in showing how to "prune" samples to achieve faster rates in random matrix settings citeguedon2014community,le2015concentration,rebrova2015coverings,rebrova2016norms, and we think the general investigation of such pruning results is likely to be fruitful.

We remark that the sample complexity in Proposition 4 is suboptimal in many cases, requiring roughly $d^{1.5}$ samples when d samples would suffice. At the end of the next subsection we discuss a tighter but more specialized bound based on spectral graph sparsification.

Applications. Propositions 2 and 4 together give us a powerful tool for deriving information-theoretic robust recovery results: one needs simply establish resilience for the population distribution p, then use Proposition 4 to obtain finite sample bounds and Proposition 2 to obtain robust recovery guarantees. We do this in three illustrative settings: ℓ_2 mean estimation, learning discrete distributions, and stochastic block models. We outline the results below; formal statements and proofs are deferred to the full version of the paper.

Mean estimation in ℓ_2-norm. Suppose that a distribution on $\mathbb{R}^d$ has bounded kth moments: $\mathbb{E}_{x\sim p}[|\langle x-\mu, v\rangle|^k]^{1/k} \le \sigma\|v\|_2$ for all v for some $k \ge 2$. Then p is $(\mathcal{O}(\sigma\epsilon^{1-1/k}), \epsilon)$-resilient in the ℓ_2-norm. Propositions 4 and 2 then imply that, given $n \ge \frac{d^{1.5}}{\epsilon} + \frac{d}{\epsilon^2}$ samples from p, and an ϵ-fraction of corruptions, it is possible to recover the mean to ℓ_2-error $\mathcal{O}(\sigma\epsilon^{1-1/k})$. Moreover, if only an α-fraction of points are good, the mean can be recovered to error $\mathcal{O}(\sigma\alpha^{-1/k})$ with probability $\Omega(\alpha)$.

The $d^{1.5}/\epsilon$ term in the sample complexity is likely loose, and we believe the true dependence on d is at most $d\log(d)$. This looseness comes from Proposition 4, which uses a naïve covering argument and could potentially be improved with more sophisticated tools. Nevertheless, it is interesting that resilience holds long before the empirical kth moments concentrate, which would require $d^{k/2}$ samples.

Distribution learning. Suppose that we are given k-tuples of independent samples from a discrete distribution: $p = \pi^k$, where π is a distribution on $\{1, \ldots, m\}$. By taking the empirical average of the k samples from π, we can treat a sample from p as an element in the m-dimensional simplex Δ_m. This distribution turns out to be resilient in the ℓ_1-norm with $\sigma = \mathcal{O}(\epsilon\sqrt{\log(1/\epsilon)/k})$, which allows us to estimate p in the ℓ_1-norm (i.e., total variation norm) and recover $\hat{\pi}$ such that $\|\hat{\pi} - \pi\|_{TV} = \mathcal{O}(\epsilon\sqrt{\log(1/\epsilon)/k})$. This reveals a pleasing "error correction" property: if we are given k samples at a time, either all or none of which are good, then our error is $\sqrt{k}$ times smaller than if we only observe the samples individually.

Stochastic block models. Finally, we consider the *semi-random stochastic block model* studied in [3]. For a graph on n vertices, this model posits a subset S of αn "good" vertices, which are connected to each other with probability $\frac{a}{n}$ and to the other ("bad") vertices with probability $\frac{b}{n}$ (where $b < a$); the connections among the bad vertices can be arbitrary. The goal is to recover the set S.

We think of each row of the adjacency matrix as a vector in $\{0,1\}^n$, and show that for the good vertices these vectors are resilient in a truncated ℓ_1-norm $\|x\|$, defined as the sum of the

αn largest coordinates of x (in absolute value). In this case, we have $\sigma = \mathcal{O}(\alpha\sqrt{a\log(2/\alpha)})$ (this requires a separate argument from Proposition 4 to get tight bounds). Applying Proposition 2, we find that we are able to recover (with probability $\frac{\alpha}{2}$) a set $\hat{S}$ with

$$\frac{1}{\alpha n}|S \triangle \hat{S}| = \mathcal{O}\left(\frac{a\log(2/\alpha)}{(a-b)^2\alpha^2}\right). \tag{2}$$

In particular, we get non-trivial guarantees as long as $\frac{(a-b)^2}{a} \gg \frac{\log(2/\alpha)}{\alpha^2}$. [3] derive a weaker (but computationally efficient) bound when $\frac{(a-b)^2}{a} \gg \frac{\log(2/\alpha)}{\alpha^3}$, and remark on the similarity to the famous *Kesten-Stigum threshold* $\frac{(a-b)^2}{a} \gg \frac{1}{\alpha^2}$, which is the conjectured threshold for computationally efficient recovery in the classical stochastic block model (see [4] for the conjecture, and [20, 18] for a proof in the two-block case). Our information-theoretic upper bound matches the Kesten-Stigum threshold up to a $\log(2/\alpha)$ factor. We conjecture that this upper bound is tight; some evidence for this is given in [23], which provides a nearly matching information-theoretic lower bound when $a = 1$, $b = \frac{1}{2}$.

1.3 Strong Convexity, Second Moments, and Efficient Algorithms

Most existing algorithmic results on robust mean estimation rely on analyzing the empirical covariance of the data in some way (see, e.g., [16, 5, 1]). In this section we establish connections between bounded covariance and resilience, and show that in a very general sense, bounded covariance is indeed sufficient to enable robust mean estimation.

Given a norm $\|\cdot\|$, we say that a set of points $x_1, \ldots, x_n$ has *variance bounded by* σ_0^2 in that norm if $\frac{1}{n}\sum_{i=1}^n \langle x_i - \mu, v\rangle^2 \le \sigma_0^2\|v\|_*^2$ (recall $\|\cdot\|_*$ denotes the dual norm). Since this implies a tail bound along every direction, it is easy to see (c.f. Lemma 3) that a set with variance bounded by σ_0^2 is $(\mathcal{O}(\sigma_0\sqrt{\epsilon}), \epsilon)$-resilient around its mean for all $\epsilon < \frac{1}{2}$. Therefore, bounded variance implies resilience.

An important result is that the converse is also true, *provided the norm is strongly convex*. We say that a norm $\|\cdot\|$ is γ-strongly convex if $\|x+y\|^2 + \|x-y\|^2 \ge 2(\|x\|^2 + \gamma\|y\|^2)$ for all $x, y \in \mathbb{R}^d$.[1] As an example, the ℓ_p-norm is $(p-1)$-strongly convex for $p \in (1, 2]$. For strongly convex norms, we show that any resilient set has a "core" with bounded variance:

▶ **Theorem 5.** *If S is $(\sigma, \frac{1}{2})$-resilient in a γ-strongly convex norm $\|\cdot\|$, then S contains a set S_0 of size at least $\frac{1}{2}|S|$ with bounded variance: $\frac{1}{|S_0|}\sum_{i\in S_0}\langle x_i - \mu, v\rangle^2 \le \frac{288\sigma^2}{\gamma}\|v\|_*^2$ for all v.*

Using Lemma 3, we can show that $(\sigma, \frac{1}{2})$-resilience is equivalent to having bounded 1st moments in every direction; Theorem 5 can thus be interpreted as saying that any set with bounded 1st moments can be pruned to have bounded 2nd moments.

We found this result quite striking—the fact that Theorem 5 can hold with no dimension-dependent factors is far from obvious. In fact, if we replace 2nd moments with 3rd moments or take a non-strongly-convex norm then the analog of Theorem 5 is false: we incur polynomial factors in the dimension even if S is the standard basis of $\mathbb{R}^d$ (see the full paper for details). The proof of Theorem 5 involves minimax duality and Khintchine's inequality. We can also strengthen Theorem 5 to yield S_0 of size $(1-\epsilon)|S|$. The proofs of both results are given in Section 3 and may be of independent interest.

[1] In the language of Banach space theory, this is also referred to as having bounded co-type.

Algorithmic results. Given points with bounded variance, we establish algorithmic results assuming that one can solve the "generalized eigenvalue" problem $\max_{\|v\|_* \le 1} v^\top A v$ up to some multiplicative accuracy κ. Specifically, we make the following assumption:

▶ **Assumption 6** (κ-Approximability). *There is a convex set $\mathcal{P}$ of PSD matrices such that*

$$\sup_{\|v\|_* \le 1} v^\top A v \le \sup_{M \in \mathcal{P}} \langle A, M \rangle \le \kappa \sup_{\|v\|_* \le 1} v^\top A v \tag{3}$$

for every PSD matrix A. Moreover, it is possible to optimize linear functions over $\mathcal{P}$ in polynomial time.

A result of [21] implies that this is true with $\kappa = \mathcal{O}(1)$ if $\|\cdot\|_*$ is any "quadratically convex" norm, which includes the ℓ_q-norms for $q \in [2, \infty]$. Also, while we do not use it in this paper, one can sometimes exploit weaker versions of Assumption 6 that only require $\sup_{M \in \mathcal{P}} \langle A, M \rangle$ to be small for certain matrices A; see for instance [17], which obtains an algorithm for robust sparse mean estimation even though Assumption 6 (as well as strong convexity) fails to hold in that setting.

Our main algorithmic result is the following:

▶ **Theorem 7.** *Suppose that $x_1, \dots, x_n$ contains a subset S of size $(1-\epsilon)n$ whose variance around its mean μ is bounded by σ_0^2 in the norm $\|\cdot\|$. Also suppose that Assumption 6 holds for the dual norm $\|\cdot\|_*$. Then, if $\epsilon \le \frac{1}{4}$, there is a polynomial-time algorithm whose output satisfies $\|\hat{\mu} - \mu\| = \mathcal{O}\big(\sigma_0 \sqrt{\kappa \epsilon}\big)$.*

If, in addition, $\|\cdot\|$ is γ-strongly convex, then even if S only has size αn there is a polynomial-time algorithm such that $\|\hat{\mu} - \mu\| = \mathcal{O}\big(\frac{\sqrt{\kappa}\sigma_0}{\sqrt{\gamma}\alpha}\big)$ with probability $\Omega(\alpha)$.

This is essentially a more restrictive, but computationally efficient version of Proposition 2. We note that for the ℓ_2-norm, the algorithm can be implemented as an SVD (singular value decomposition) combined with a filtering step; for more general norms, the SVD is replaced with a semidefinite program.

In the small-ϵ regime, Theorem 7 is in line with existing results which typically achieve errors of $\mathcal{O}(\sqrt{\epsilon})$ in specific norms. While several papers achieve stronger rates of $\mathcal{O}(\epsilon^{3/4})$ citelai2016agnostic or $\tilde{\mathcal{O}}(\epsilon)$ [5, 1], these stronger results rely crucially on specific distributional assumptions such as Gaussianity. At the time of writing of this paper, no results obtained rates better than $\mathcal{O}(\sqrt{\epsilon})$ for any general class of distributions (even under strong assumptions such as sub-Gaussianity). After initial publication of this paper, [15] surpassed $\sqrt{\epsilon}$ and obtained rates of $\epsilon^{1-\gamma}$ for any $\gamma > 0$, for distributions satisfying the Poincaré isoperimetric inequality.

In the small-α regime, Theorem 7 generalizes the mean estimation results of [3] to norms beyond the ℓ_2-norm. That paper achieves a better rate of $1/\sqrt{\alpha}$ (vs. the $1/\alpha$ rate given here). It is likely possible to achieve the $1/\sqrt{\alpha}$ rate here as well, but we leave this for future work.

Applications. Because Assumption 6 holds for ℓ_p-norms, we can perform robust estimation in ℓ_p-norms for any $p \in [1, 2]$, as long as the data have bounded variance in the dual ℓ_q-norm (where $\frac{1}{p} + \frac{1}{q} = 1$). This is the first efficient algorithm for performing robust mean estimation in any ℓ_p-norm with $p \neq 2$. The ℓ_1-norm in particular is often a more meaningful metric than the ℓ_2-norm in discrete settings, allowing us to improve on existing results.

Indeed, as in the previous section, suppose we are given k-tuples of samples from a discrete distribution π on $\{1, \dots, m\}$. Applying Theorem 7 with the ℓ_1-norm yields an

algorithm recovering a $\hat{\pi}$ with $\|\hat{\pi} - \pi\|_{TV} = \tilde{\mathcal{O}}(\sqrt{\epsilon/k})$.[2] In contrast, bounds using the ℓ_2-norm would only yield $\|\hat{\pi} - \pi\|_2 = \mathcal{O}(\sqrt{\epsilon\pi_{\max}/k})$, which is substantially weaker when the maximum probability $\pi_{\max}$ is large. Our result has a similar flavor to that of [5] on robustly estimating binary product distributions, for which directly applying ℓ_2 mean estimation was also insufficient. We discuss our bounds in more detail in the full version of the paper.

Finite-sample bound. To get the best sample complexity for the applications above, we provide an additional finite-sample bound focused on showing that a set of points has bounded variance. This is a simple but useful generalization of Proposition B.1 of [3]; it shows that in a very generic sense, given d samples from a distribution on $\mathbb{R}^d$ with bounded population variance, we can find a subset of samples with bounded variance with high probability. It involves pruning the samples in a non-trivial way based on ideas from graph sparsification [2]. The formal statement is given in Section 6.2.

1.4 Low-Rank Recovery

Finally, to illustrate that the idea behind resilience is quite general and not restricted to mean estimation, we also provide results on recovering a rank-k approximation to the data in the presence of arbitrary outliers. Given a set of points $[x_i]_{i\in S}$, let X_S be the matrix whose columns are the x_i. Our goal is to obtain a low-rank matrix P such that the operator norm $\|(I-P)X_S\|_2$ is not much larger than $\sigma_{k+1}(X_S)$, where σ_{k+1} denotes the $k+1$st singular value; we wish to do this even if S is corrupted to a set $\tilde{S}$ by adding arbitrary outliers.

As before, we start by formulating an appropriate resilience criterion:

▶ **Definition 8** (Rank-resilience)**.** A set of points $[x_i]_{i\in S}$ in $\mathbb{R}^d$ is *δ-rank-resilient* if for all subsets T of size at least $(1-\delta)|S|$, we have $\mathrm{col}(X_T) = \mathrm{col}(X_S)$ and $\|X_T^\dagger X_S\|_2 \le 2$, where $\dagger$ is the pseudoinverse and col denotes column space.

Rank-resilience says that the variation in X should be sufficiently spread out: there should not be a direction of variation that is concentrated in only a δ-fraction of the points. Under rank-resilience, we can perform efficient rank-k recovery even in the presence of a δ-fraction of arbitrary data:

▶ **Theorem 9.** *Let $\delta \le \frac{1}{3}$. If a set of n points contains a set S of size $(1-\delta)n$ that is δ-rank-resilient, then it is possible to efficiently recover a matrix P of rank at most $15k$ such that $\|(I-P)X_S\|_2 = \mathcal{O}(\sigma_{k+1}(X_S))$.*

The power of Theorem 9 comes from the fact that the error depends on σ_{k+1} rather than e.g. σ_2, which is what previous results yielded. This distinction is crucial in practice, since most data have a few (but more than one) large singular values followed by many small singular values. Note that in contrast to Theorem 7, Theorem 9 only holds when S is relatively large: at least $(1-\delta)n \ge \frac{2}{3}n$ in size.

1.5 Summary, Technical Highlights, and Roadmap

In summary, we have provided a deterministic condition on a set of points that enables robust mean estimation, and provided finite-sample bounds showing that this condition holds with

[2] The $\tilde{\mathcal{O}}$ notation suppresses log factors in m and ϵ; the dependence on m can likely be removed with a more careful analysis.

high probability in many concrete settings. This yields new results for distribution learning, stochastic block models, mean estimation under bounded moments, and mean estimation in ℓ_p norms. We also provided an extension of our condition that yields results for robust low-rank recovery.

Beyond the results themselves, the following technical aspects of our work may be particularly interesting: The proof of Proposition 2 (establishing that resilience is indeed sufficient for robust estimation), while simple, is a nice pigeonhole argument that we found to be conceptually illuminating.

In addition, the proof of Theorem 5, on pruning resilient sets to obtain sets with bounded variance, exploits strong convexity in a non-trivial way in conjunction with minimax duality; we think it reveals a fairly non-obvious geometric structure in resilient sets, and also shows how the ability to prune points can yield sets with meaningfully stronger properties.

Finally, in the proof of our algorithmic result (Theorem 7), we establish an interesting generalization of the inequality $\sum_{i,j} X_{ij}^2 \leq \text{rank}(X) \cdot \|X\|_2^2$, which holds not just for the ℓ_2-norm but for any strongly convex norm. This is given as Lemma 18.

Roadmap. The rest of the paper is organized as follows. In Section 2, we prove our information-theoretic recovery result for resilient sets (Proposition 2). In Section 3, we prove Theorem 5 establishing that all resilient sets in strongly convex norms contain large subsets with bounded variance; we also prove a more precise version of Theorem 5 in Section 3.1. In Section 4, we prove our algorithmic results, warming up with the ℓ_2-norm (Section 4.1) and then moving to general norms (Section 4.2). In Section 5, we prove our results on rank-k recovery. In Section 6, we present and prove the finite-sample bounds discussed in Section 1.2. Applications of our results are deferred to the full version of the paper.

1.6 Related Work

A number of authors have recently studied robust estimation and learning in high-dimensional settings: [16] study mean and covariance estimation, while [5] focus on estimating Gaussian and binary product distributions, as well as mixtures thereof; note that this implies mean/-covariance estimation of the corresponding distributions. [3] recently showed that robust estimation is possible even when the fraction α of "good" data is less than $\frac{1}{2}$. We refer to these papers for an overview of the broader robust estimation literature; since those papers, a number of additional results have also been published: [6] provide a case study of various robust estimation methods in a genomic setting, [1] study sparse mean estimation, and others have studied problems including regression, Bayes nets, planted clique, and several other settings [8, 9, 7, 10, 12, 19].

Special cases of the resilience criterion are implicit in some of these earlier works; for instance, ℓ_2-resilience appears in equation (9) in [5], and resilience in a sparsity-inducing norm appears in Theorem 4.5 of [17]. However, these conditions typically appear concurrently with other stronger conditions, and the general sufficiency of resilience for information-theoretic recovery appears to be unappreciated (for instance, [17], despite already having implicitly established resilience, proves its information-theoretic results via reduction to a tournament lemma from [5]).

Low rank estimation was studied by [16], but their bounds depend on the maximum eigenvalue $\|\Sigma\|_2$ of the covariance matrix, while our bound provides robust recovery guarantees in terms of lower singular values of Σ. (Some work, such as [5], shows how to estimate all of Σ in e.g. Frobenius norm, but appears to require the samples to be drawn from a Gaussian.)

2 Resilience and Robustness: Information-Theoretic Sufficiency

Recall the definition of resilience: S is (σ,ϵ)-resilient if $\|\frac{1}{|T|}\sum_{i\in T}(x_i-\mu)\|\le\sigma$ whenever $T\subseteq S$ and $|T|\ge(1-\epsilon)|S|$. Here we establish Proposition 2 showing that, if we ignore computational efficiency, resilience leads directly to an algorithm for robust mean estimation.

Proof (Proposition 2). We prove Proposition 2 via a constructive (albeit exponential-time) algorithm. To prove the first part, suppose that the true set S is $(\sigma,\frac{\epsilon}{1-\epsilon})$-resilient around μ, and let S' be any set of size $(1-\epsilon)n$ that is $(\sigma,\frac{\epsilon}{1-\epsilon})$-resilient (around some potentially different vector μ'). We claim that μ' is sufficiently close to μ.

Indeed, let $T=S\cap S'$, which by the pigeonhole principle has size at least $(1-2\epsilon)n=\frac{1-2\epsilon}{1-\epsilon}|S|=(1-\frac{\epsilon}{1-\epsilon})|S|$. Therefore, by the definition of resilience,

$$\left\|\tfrac{1}{|T|}\textstyle\sum_{i\in T}(x_i-\mu)\right\|\le\sigma. \tag{4}$$

But by the same argument, $\|\frac{1}{|T|}\sum_{i\in T}(x_i-\mu')\|\le\sigma$ as well. By the triangle inequality, $\|\mu-\mu'\|\le 2\sigma$, which completes the first part of the proposition.

For the second part, we need the following lemma relating ϵ-resilience to $(1-\epsilon)$-resilience:

▶ **Lemma 10.** *For any $0<\epsilon<1$, a distribution/set is (σ,ϵ)-resilient around its mean μ if and only if it is $(\frac{1-\epsilon}{\epsilon}\sigma,1-\epsilon)$-resilient. Moreover, even if μ is not the mean, the distribution/set is $(\frac{2-\epsilon}{\epsilon}\sigma,1-\epsilon)$-resilient. In other words, if $\|\frac{1}{|T|}\sum_{i\in T}(x_i-\mu)\|\le\sigma$ for all sets T of size at least $(1-\epsilon)n$, then $\|\frac{1}{|T'|}\sum_{i\in T'}(x_i-\mu)\|\le\frac{2-\epsilon}{\epsilon}\sigma$ for all sets T' of size at least ϵn.*

Given Lemma 10, the second part of Proposition 2 is similar to the first part, but requires us to consider multiple resilient sets S_i rather than a single S'. Suppose S is $(\sigma,\frac{\alpha}{4})$-resilient around μ–and thus also $(\frac{8}{\alpha}\sigma,1-\frac{\alpha}{4})$-resilient by Lemma 10–and let $S_1,\dots,S_m$ be a maximal collection of subsets of $[n]$ such that:

1. $|S_j|\ge\frac{\alpha}{2}n$ for all j.
2. S_j is $(\frac{8}{\alpha}\sigma,1-\frac{\alpha}{2})$-resilient around some point μ_j.
3. $S_j\cap S_{j'}=\emptyset$ for all $j\ne j'$.

Clearly $m\le\frac{2}{\alpha}$. We claim that at least one of the μ_j is close to μ. By maximality of the collection $\{S_j\}_{j=1}^m$, it must be that $S_0=S\backslash(S_1\cup\dots\cup S_m)$ cannot be added to the collection. First suppose that $|S_0|\ge\frac{\alpha}{2}n$. Then S_0 is $(\frac{8}{\alpha}\sigma,1-\frac{\alpha}{2})$-resilient (because any subset of $\frac{\alpha}{2}|S_0|$ points in S_0 is a subset of at least $\frac{\alpha}{4}|S|$ points in S). But this contradicts the maximality of $\{S_j\}_{j=1}^m$, so we must have $|S_0|<\frac{\alpha}{2}n$.

Now, this implies that $|S\cap(S_1\cup\dots\cup S_m)|\ge\frac{\alpha}{2}n$, so by pigeonhole we must have $|S\cap S_j|\ge\frac{\alpha}{2}|S_j|$ for some j. Letting $T=S\cap S_j$ as before, we find that $|T|\ge\frac{\alpha}{2}|S_j|\ge\frac{\alpha}{4}|S|$ and hence by resilience of S_j and S we have $\|\mu-\mu_j\|\le 2\cdot(\frac{8}{\alpha}\sigma)=\frac{16}{\alpha}\sigma$. If we output one of the μ_j at random, we are then within the desired distance of μ with probability $\frac{1}{m}\ge\frac{\alpha}{2}$. ◀

3 Powering up Resilience: Finding a Core with Bounded Variance

In this section we prove Theorem 5, which says that for strongly convex norms, every resilient set contains a core with bounded variance. Recall that this is important for enabling algorithmic applications that depend on a bounded variance condition.

First recall the definition of resilience (Definition 1): a set S is (σ,ϵ)-resilient if for every set $T\subseteq S$ of size $(1-\epsilon)|S|$, we have $\|\frac{1}{|T|}\sum_{i\in T}(x_i-\mu)\|\le\sigma$. For $\epsilon=\frac{1}{2}$, we observe that resilience in a norm is equivalent to having bounded first moments in the dual norm:

▶ **Lemma 11.** *Suppose that S is $(\sigma, \frac{1}{2})$-resilient in a norm $\|\cdot\|$, and let $\|\cdot\|_*$ be the dual norm. Then S has 1st moments bounded by 3σ: $\frac{1}{|S|}\sum_{i\in S}|\langle x_i - \mu, v\rangle| \le 3\sigma\|v\|_*$ for all $v \in \mathbb{R}^d$.*

Conversely, if S has 1st moments bounded by σ, it is $(2\sigma, \frac{1}{2})$-resilient.

The proof is routine and can be found in the full paper. Supposing a set has bounded 1st moments, we will show that it has a large core with bounded second moments. This next result is *not* routine:

▶ **Proposition 12.** *Let S be any set with 1st moments bounded by σ. Then if the norm $\|\cdot\|$ is γ-strongly convex, there exists a core S_0 of size at least $\frac{1}{2}|S|$ with variance bounded by $\frac{32\sigma^2}{\gamma}$. That is, $\frac{1}{|S_0|}\sum_{i\in S_0}|\langle x_i - \mu, v\rangle|^2 \le \frac{32\sigma^2}{\gamma}\|v\|_*^2$ for all $v \in \mathbb{R}^d$.*

The assumptions seem necessary: i.e., such a core does not exist when $\|\cdot\|$ is the ℓ_p-norm with $p > 2$ (which is a non-strongly-convex norm), or with bounded 3rd moments for $p = 2$. The proof of Proposition 12 uses minimax duality and Khintchine's inequality [13]. Note that Lemma 11 and Proposition 12 together imply Theorem 5.

Proof (Proposition 12). Without loss of generality take $\mu = 0$ and suppose that $S = [n]$. We can pose the problem of finding a resilient core as an integer program:

$$\min_{c\in\{0,1\}^n, \|c\|_1 \ge \frac{n}{2}} \max_{\|v\|_* \le 1} \frac{1}{n}\sum_{i=1}^n c_i |\langle x_i, v\rangle|^2. \tag{5}$$

Here the variable c_i indicates whether the point i lies in the core S_0. By taking a continuous relaxation and applying a standard duality argument, we obtain the following:

▶ **Lemma 13.** *Suppose that for all m and all vectors $v_1, \dots, v_m$ satisfying $\sum_{j=1}^m \|v_j\|_*^2 \le 1$, we have*

$$\frac{1}{n}\sum_{i=1}^n \sqrt{\sum_{j=1}^m |\langle x_i, v_j\rangle|^2} \le B. \tag{6}$$

Then the value of (5) is at most $8B^2$.

The proof is straightforward and deferred to the full paper. Now, to bound (6), let $s_1, \dots, s_m \in \{-1, +1\}$ be i.i.d. random sign variables. We have

$$\begin{aligned}
\frac{1}{n}\sum_{i=1}^n \sqrt{\sum_{j=1}^m |\langle x_i, v_j\rangle|^2} &\overset{(i)}{\le} \mathbb{E}_{s_{1:m}}\left[\frac{\sqrt{2}}{n}\sum_{i=1}^n \left|\sum_{j=1}^m s_j \langle x_i, v_j\rangle\right|\right] && (7)\\
&= \mathbb{E}_{s_{1:m}}\left[\frac{\sqrt{2}}{n}\sum_{i=1}^n \left|\left\langle x_i, \sum_{j=1}^m s_j v_j\right\rangle\right|\right] && (8)\\
&\overset{(ii)}{\le} \mathbb{E}_{s_{1:m}}\left[\sqrt{2}\sigma\left\|\sum_{j=1}^m s_j v_j\right\|_*\right] && (9)\\
&\le \sqrt{2}\sigma\mathbb{E}_{s_{1:m}}\left[\left\|\sum_{j=1}^m s_j v_j\right\|_*^2\right]^{\frac{1}{2}}. && (10)
\end{aligned}$$

Here (i) is Khintchine's inequality [11] and (ii) is the assumed first moment bound. It remains to bound (10). The key is the following inequality asserting that the dual norm $\|\cdot\|_*$ is strongly smooth whenever $\|\cdot\|$ is strongly convex (c.f. Lemma 17 of [22]):

▶ **Lemma 14.** *If $\|\cdot\|$ is γ-strongly convex, then $\|\cdot\|_*$ is $(1/\gamma)$-strongly smooth: $\frac{1}{2}(\|v+w\|_*^2 + \|v-w\|_*^2) \le \|v\|_*^2 + (1/\gamma)\|w\|_*^2$.*

Applying Lemma 14 inductively to $\mathbb{E}_{s_{1:m}}\left[\left\|\sum_{j=1}^m s_j v_j\right\|_*^2\right]$, we obtain

$$\mathbb{E}_{s_{1:m}}\left[\left\|\sum_{j=1}^m s_j v_j\right\|_*^2\right] \le \frac{1}{\gamma}\sum_{j=1}^m \|v_j\|_*^2 \le \frac{1}{\gamma}. \tag{11}$$

Combining with (10), we have the bound $B \le \sigma\sqrt{2/\gamma}$, which yields the desired result. ◀

3.1 Finding Resilient Cores when $\alpha \approx 1$

Lemma 11 together with Proposition 12 show that a $(\sigma, \frac{1}{2})$-resilient set has a core with bounded 2nd moments. One piece of looseness is that Proposition 12 only exploits resilience for $\epsilon = \frac{1}{2}$, and hence is not sensitive to the degree of (σ,ϵ)-resilience as $\epsilon \to 0$. In particular, it only yields a core S_0 of size $\frac{1}{2}|S|$, while we might hope to find a much larger core of size $(1-\epsilon)|S|$ for some small ϵ.

Here we tighten Proposition 12 to make use of finer-grained resilience information. Recall that we let $\sigma_*(\epsilon)$ denote the resilience over sets of size $(1-\epsilon)|S|$. For a given ϵ, our goal is to construct a core S_0 of size $(1-\epsilon)|S|$ with small second moments. The following key quantity will tell us how small the second moments can be:

$$\widetilde{\sigma}_*(\epsilon) \stackrel{\text{def}}{=} \sqrt{\int_{\epsilon/2}^{1/2} u^{-2}\sigma_*(u)^2 du}. \tag{12}$$

The following proposition says that $\widetilde{\sigma}_*$ controls the 2nd moments of S_0:

▶ **Proposition 15.** *Let S be any resilient set in a γ-strongly-convex norm. Then for any $\epsilon \le \frac{1}{2}$, there exists a core S_0 of size $(1-\epsilon)|S|$ with variance bounded by $\mathcal{O}\left(\widetilde{\sigma}_*^2(\epsilon)/\gamma\right)$.*

The proof is similar to Proposition 12, but requires more careful bookkeeping.

To interpret $\widetilde{\sigma}_*$, suppose that $\sigma_*(\epsilon) = \sigma\epsilon^{1-1/r}$ for some $r \in [1,2)$, which roughly corresponds to having bounded rth moments. Then $\widetilde{\sigma}_*^2(\epsilon) = \sigma^2\int_{\epsilon/2}^{1/2} u^{-2/r}du \le \frac{\sigma^2}{2/r-1}\left(\frac{2}{\epsilon}\right)^{2/r-1}$. If $r=1$ then a core of size $(1-\epsilon)|S|$ might require second moments as large as $\frac{\sigma^2}{\epsilon}$; on the other hand, as $r \to 2$ the second moments can be almost as small as σ^2. In general, $\widetilde{\sigma}_*(\epsilon)$ is $\mathcal{O}\left(\sigma\epsilon^{1/2-1/r}\right)$ if $r \in [1,2)$, is $\mathcal{O}\left(\sigma\sqrt{\log(1/\epsilon)}\right)$ if $r=2$, and is $\mathcal{O}(\sigma)$ if $r>2$.

4 Efficient Recovery Algorithms

We now turn our attention to the question of efficient algorithms. The main point of this section is to prove Theorem 7, which yields efficient robust mean estimation for a general class of norms.

4.1 Warm-Up: Recovery in ℓ_2-norm

We first prove a warm-up to Theorem 7 which focuses on the ℓ_2-norm. Our warm-up is:

▶ **Proposition 16.** *Let $x_1,\dots,x_n \in \mathbb{R}^d$, and let S be a subset of size αn with bounded variance in the ℓ_2-norm: $\lambda_{\max}(\frac{1}{|S|}\sum_{i\in S}(x_i-\mu)(x_i-\mu)^\top) \le \sigma^2$, where μ is the mean of S. Then there is an efficient randomized algorithm (Algorithm 1) which with probability $\Omega(\alpha)$ outputs a parameter $\hat{\mu}$ such that $\|\mu-\hat{\mu}\|_2 = \mathcal{O}\left(\frac{\sigma}{\alpha}\right)$. Moreover, if $\alpha = 1-\epsilon \ge \frac{3}{4}$ then $\|\mu-\hat{\mu}\|_2 = \mathcal{O}(\sigma\sqrt{\epsilon})$ with probability 1.*

Algorithm 1 Algorithm for recovering the mean of a set with bounded variance in ℓ_2-norm.

1: Initialize $c_i = 1$ for all $i = 1, \ldots, n$ and $\mathcal{A} = \{1, \ldots, n\}$.
2: Let $Y \in \mathbb{R}^{d \times d}$ and $W \in \mathbb{R}^{\mathcal{A} \times \mathcal{A}}$ be the maximizer/minimizer of the saddle point problem

$$\max_{\substack{Y \succeq 0, \\ \mathrm{tr}(Y) \leq 1}} \min_{\substack{0 \leq W_{ji} \leq \frac{4-\alpha}{\alpha(2+\alpha)n}, \\ \sum_j W_{ji} = 1}} \sum_{i \in \mathcal{A}} c_i (x_i - X_{\mathcal{A}} w_i)^\top Y (x_i - X_{\mathcal{A}} w_i). \tag{14}$$

3: Let $\tau_i^\star = (x_i - X_{\mathcal{A}} w_i)^\top Y (x_i - X_{\mathcal{A}} w_i)$.
4: **if** $\sum_{i \in \mathcal{A}} c_i \tau_i^\star > 4n\sigma^2$ **then**
5: For $i \in \mathcal{A}$, replace c_i with $\left(1 - \frac{\tau_i^\star}{\tau_{\max}}\right) c_i$, where $\tau_{\max} = \max_{i \in \mathcal{A}} \tau_i^\star$.
6: For all i with $c_i < \frac{1}{2}$, remove i from $\mathcal{A}$.
7: Go back to line 2.
8: **end if**
9: Let W_1 be the result of zeroing out all singular values of W that are greater than 0.9.
10: Let $Z = X_{\mathcal{A}} W_0$, where $W_0 = (W - W_1)(I - W_1)^{-1}$.
11: **if** $\mathrm{rank}(Z) = 1$ **then**
12: Output the average of the columns of $X_{\mathcal{A}}$.
13: **else**
14: Output a column of Z at random.
15: **end if**

At the heart of Algorithm 1 is the following optimization problem:

$$\begin{aligned} &\underset{W \in \mathbb{R}^{n \times n}}{\text{minimize}} \quad \|X - XW\|_2^2 \\ &\text{subject to} \quad 0 \leq W_{ji} \leq \frac{1}{\alpha n} \ \forall i, j, \quad \sum_j W_{ji} = 1 \ \forall i. \end{aligned} \tag{13}$$

Here $X \in \mathbb{R}^{d \times n}$ is the data matrix $[x_1 \ \cdots \ x_n]$ and $\|X - XW\|_2$ is the operator norm (maximum singular value) of $X - XW$. Note that (13) can be expressed as a semidefinite program; however, it can actually be solved more efficiently than this, via a singular value decomposition (see the full paper for details).

The idea behind (13) is to re-construct each x_i as an average of αn other x_j. Note that by assumption we can always re-construct each element of S using the mean of S, and have small error. Intuitively, any element that cannot be re-constructed well must not lie in S, and can be safely removed. We do a soft form of removal by maintaining weights c_i on the points x_i (initially all 1), and downweighting points with high reconstruction error. We also maintain an active set $\mathcal{A}$ of points with $c_i \geq \frac{1}{2}$.

Informally, Algorithm 1 for estimating μ takes the following form:

1. Solve the optimization problem (13).
2. If the optimum is $\gg \sigma^2 n$, then find the columns of X that are responsible for the optimum being large, and downweight them.
3. Otherwise, if the optimum is $\mathcal{O}(\sigma^2 n)$, then take a low rank approximation W_0 to W, and return a randomly chosen column of XW_0.

The hope in step 3 is that the low rank projection XW_0 will be close to μ for the columns belonging to S. The choice of operator norm is crucial: it means we can actually expect XW to be close to X (on the order of $\sigma\sqrt{n}$). In contrast, the Frobenius norm scales as $\sigma\sqrt{nd}$.

Finally, we note that $\|X - XW\|_2^2$ is equal to

$$\|X - XW\|_2^2 = \max_{Y \succeq 0, \operatorname{tr}(Y) \leq 1} \sum_{i=1}^{n} (x_i - Xw_i)^\top Y (x_i - Xw_i), \tag{15}$$

which is the form we use in Algorithm 1.

Proof (Proposition 16). We show two things: (1) that the outlier removal step removes many more outliers than good points, and (2) that many columns of XW_0 are close to μ.

Outlier removal. To analyze the outlier removal step (step 2 above, or lines 5-6 of Algorithm 1), we make use of the following general lemma:

▶ **Lemma 17.** *For any scalars τ_i and a, suppose that $\sum_{i \in \mathcal{A}} c_i \tau_i \geq 4a$ while $\sum_{i \in S \cap \mathcal{A}} c_i \tau_i \leq \alpha a$. Then the following invariants are preserved by lines 5-6 of Algorithm 1: (i) $\sum_{i \in S}(1 - c_i) \leq \frac{\alpha}{4} \sum_{i=1}^{n}(1 - c_i)$, and (ii) $|S \cap \mathcal{A}| \geq \frac{\alpha(2+\alpha)}{4-\alpha} n$.*

Lemma 17 says that we downweight points within S at least 4 times slower than we do overall (property i), and in particular we never remove too many points from S (property ii). This type of lemma is not new (cf. Lemma 4.5 of [3]) and its proof is deferred to the full paper.

We will show that we can take $a = n\sigma^2$ in Lemma 17, or in other words that $\sum_{i \in S \cap \mathcal{A}} c_i \tau_i^\star \leq \alpha n \sigma^2$. Let $\tau_i(w) = (x_i - X_{\mathcal{A}} w)^\top (x_i - X_{\mathcal{A}} w)$, and note that $\tau_i^\star = \tau_i(w_i) = \min\{\tau_i(w) \mid 0 \leq w_j \leq \frac{1}{\alpha n}, \sum_j w_j = 1\}$. This is because for a fixed Y, each of the w_i are optimized independently.

We can therefore bound $\tau_i^\star$ by substituting any feasible $\hat{w}_i$. We will choose $\hat{W}_{ji} = \frac{\mathbb{I}[j \in S \cap \mathcal{A}]}{|S \cap \mathcal{A}|}$, in which case $X_{\mathcal{A}} \hat{w}_i = \hat{\mu}$, where $\hat{\mu}$ is the average of x_j over $S \cap \mathcal{A}$. Then we have

$$\begin{aligned}
\sum_{i \in S \cap \mathcal{A}} c_i \tau_i^\star &\leq \sum_{i \in S \cap \mathcal{A}} c_i \tau_i(\hat{w}_i) && (16)\\
&\leq \sum_{i \in S \cap \mathcal{A}} c_i (x_i - \hat{\mu})^\top Y (x_i - \hat{\mu}) && (17)\\
&\overset{(i)}{\leq} \sum_{i \in S \cap \mathcal{A}} c_i (x_i - \mu)^\top Y (x_i - \mu) && (18)\\
&\leq \sum_{i \in S} (x_i - \mu)^\top Y (x_i - \mu) \leq \alpha n \sigma^2 \operatorname{tr}(Y) \leq \alpha n \sigma^2 && (19)
\end{aligned}$$

as desired; (i) is because the covariance around the mean ($\hat{\mu}$) is smaller than around any other point (μ).

Analyzing XW_0. By Lemma 17, we will eventually exit the if statement and obtain $Z = X_{\mathcal{A}} W_0$. It therefore remains to analyze Z; we will show in particular that $\|Z_{\mathcal{A} \cap S} - \mu \mathbb{1}^\top\|_F$ is small, where the subscript indicates restricting to the columns in $S \cap \mathcal{A}$. At a high level, it suffices to show that W_0 has low rank (so that Frobenius norm is close to spectral norm) and that XW_0 and X are close in spectral norm (note that X and $\mu \mathbb{1}^\top$ are close by assumption).

To bound $\operatorname{rank}(W_0)$, note that the constraints in (14) imply that $\|W\|_F^2 \leq \frac{4-\alpha}{\alpha(2+\alpha)}$, and so at most $\frac{4-\alpha}{0.81\alpha(2+\alpha)}$ singular values of W can be greater than 0.9. Importantly, at most 1 singular value can be greater than 0.9 if $\alpha \geq \frac{3}{4}$, and at most $\mathcal{O}(\frac{1}{\alpha})$ can be in general. Therefore, $\operatorname{rank}(W_0) \leq \mathcal{O}(\frac{1}{\alpha})$.

Next, we show that $X_{\mathcal{A}}$ and Z are close in operator norm. Indeed, $X_{\mathcal{A}} - Z = X_{\mathcal{A}}(I - W_0) = X_{\mathcal{A}}(I - W)(I - W_1)^{-1}$, hence:

$$\|X_{\mathcal{A}} - Z\|_2 = \|X_{\mathcal{A}}(I - W)(I - W_1)^{-1}\|_2 \tag{20}$$

$$\leq \|X_{\mathcal{A}}(I - W)\|_2 \|(I - W_1)^{-1}\|_2 \tag{21}$$

$$\overset{(i)}{\leq} 10\|X_{\mathcal{A}}(I - W)\|_2 \tag{22}$$

$$\overset{(ii)}{\leq} 10\sqrt{2}\|X_{\mathcal{A}}(I - W)\operatorname{diag}(c_{\mathcal{A}})^{1/2}\|_2 \overset{(iii)}{\leq} 20\sqrt{2n}\sigma. \tag{23}$$

Here (i) is because all singular values of W_1 are less than 0.9, (ii) is because $\operatorname{diag}(c_{\mathcal{A}})^{1/2} \succeq \frac{1}{\sqrt{2}} I$, and (iii) is by the condition in the if statement (line 4 of Algorithm 1), since the sum on line 4 is equal to $\|X_{\mathcal{A}}(I - W)\operatorname{diag}(c_{\mathcal{A}})^{1/2}\|_2^2$.

Combining the previous two observations, we have

$$\sum_{i \in S \cap \mathcal{A}} \|z_i - \mu\|_2^2 \leq (\operatorname{rank}(Z) + 1)\|[z_i - \mu]_{i \in S \cap \mathcal{A}}\|_2^2 \tag{24}$$

$$\leq (\operatorname{rank}(Z) + 1)\left(\|[z_i - x_i]_{i \in S \cap \mathcal{A}}\|_2 + \|[x_i - \mu]_{i \in S \cap \mathcal{A}}\|_2\right)^2 \tag{25}$$

$$\overset{(i)}{\leq} (\operatorname{rank}(Z) + 1)\left(20\sqrt{2n}\sigma + \sqrt{\alpha n}\sigma\right)^2 = \mathcal{O}\left(\tfrac{\sigma^2}{\alpha} n\right). \tag{26}$$

Here (i) uses the preceding bound on $\|X_{\mathcal{A}} - Z\|_2$, together with the 2nd moment bound $\|[x_i - \mu]_{i \in S}\|_2 \leq \sqrt{\alpha n}\sigma$. Note that $\operatorname{rank}(Z) \leq \operatorname{rank}(W_0) = \mathcal{O}(\frac{1}{\alpha})$.

Since $|S \cap \mathcal{A}| = \Omega(\alpha n)$ by Lemma 17, the average value of $\|z_i - \mu\|_2^2$ over $S \cap \mathcal{A}$ is $\mathcal{O}\big(\frac{\sigma^2}{\alpha^2}\big)$, and hence with probability at least $\frac{|S \cap \mathcal{A}|}{2|\mathcal{A}|} = \Omega(\alpha)$, a randomly chosen z_i will be within distance $\mathcal{O}\left(\frac{\sigma}{\alpha}\right)$ of μ, which completes the first part of Proposition 16.

For the second part, when $\alpha = 1 - \epsilon \geq \frac{3}{4}$, recall that we have $\operatorname{rank}(W_0) = 1$, and that $W_0 = (W - W_1)(I - W_1)^{-1}$. One can then verify that $\mathbb{1}^\top W_0 = \mathbb{1}^\top$. Therefore, $W_0 = u\mathbb{1}^\top$ for some u. Letting $\tilde{\mu} = X_{\mathcal{A}} u$, we have $\|X_{\mathcal{A}} - \tilde{\mu}\mathbb{1}^\top\|_2 \leq 20\sqrt{2n}\sigma$ by (23). In particular, $\mathcal{A}$ is resilient (around its mean) with $\sigma(\epsilon) \leq 20\sigma\sqrt{\frac{2\epsilon}{1-\epsilon}} \leq 40\sigma\sqrt{\epsilon}$ for $\epsilon \leq \frac{1}{2}$. Thus by the proof of Proposition 2 and the fact that $|\mathcal{A}| \geq |S \cap \mathcal{A}| \geq \frac{\alpha(2+\alpha)}{4-\alpha} n \geq (1 - \frac{5}{3}(1 - \alpha))n$, the mean of $\mathcal{A}$ is within $\mathcal{O}(\sigma\sqrt{1 - \alpha})$ of μ, as desired. ◀

4.2 General Case

We are now ready to prove our general algorithmic result, Theorem 7 from Section 1.3. For convenience we recall Theorem 7 here:

▶ **Theorem.** *Suppose that $x_1, \ldots, x_n$ contains a subset S of size $(1 - \epsilon)n$ whose variance around its mean μ is bounded by σ^2 in the norm $\|\cdot\|$. Also suppose that Assumption 6 (κ-approximability) holds for the dual norm $\|\cdot\|_*$. Then, if $\epsilon \leq \frac{1}{4}$, there is a polynomial-time algorithm whose output satisfies $\|\hat{\mu} - \mu\| = \mathcal{O}\big(\sigma\sqrt{\kappa\epsilon}\big)$.*

If, in addition, $\|\cdot\|$ is γ-strongly convex, then even if S only has size αn there is a polynomial-time algorithm such that $\|\hat{\mu} - \mu\| = \mathcal{O}\big(\frac{\sqrt{\kappa}\sigma}{\sqrt{\gamma}\alpha}\big)$ with probability $\Omega(\alpha)$.

Recall that bounded variance means that $\frac{1}{|S|}\sum_{i \in S}\langle x_i - \mu, v\rangle^2 \leq \sigma^2\|v\|_*^2$ for all $v \in \mathbb{R}^d$. There are two equivalent conditions to bounded variance that will be useful. The first is $\sup_{\|v\|_* \leq 1} v^\top \Sigma v \leq \sigma^2$ for all $v \in \mathbb{R}^d$, where $\Sigma = \frac{1}{|S|}\sum_{i \in S}(x_i - \mu)(x_i - \mu)^\top$; this is useful because Assumption 6 allows us to κ-approximate this supremum for any given Σ.

The second equivalent condition re-interprets σ in terms of a matrix norm. Let $\|\cdot\|_\psi$ denote the norm $\|\cdot\|$ above, and for a matrix M define the induced $2 \to \psi$-norm $\|M\|_{2 \to \psi}$ as

$\sup_{\|u\|_2 \le 1} \|Mu\|_\psi$. Then the set S has variance at most σ^2 if and only if $\|[x_i - \mu]_{i \in S}\|_{2\to\psi} \le \sqrt{|S|}\sigma$. This will be useful because induced norms satisfy helpful compositional properties such as $\|AB\|_{2\to\psi} \le \|A\|_{2\to\psi}\|B\|_2$.

The algorithm establishing Theorem 7 is almost identical to Algorithm 1, with two changes. The first change is that on line 4, the quantity $4n\sigma^2$ is replaced with $4\kappa n\sigma^2$, where κ is the approximation factor in Assumption 6. The second change is that in the optimization (14), the constraint $Y \succeq 0, \mathrm{tr}(Y) \le 1$ is replaced with $Y \in \mathcal{P}$, where $\mathcal{P}$ is the feasible set in Assumption 6. In other words, the only difference is that rather than finding the maximum eigenvalue, we κ-approximate the $2 \to \psi$ norm using Assumption 6. We therefore end up solving the saddle point problem

$$\max_{Y \in \mathcal{P}} \min_{W} \Big\{ \sum_{i \in \mathcal{A}} c_i (x_i - X_{\mathcal{A}} w_i)^\top Y (x_i - X_{\mathcal{A}} w_i) \mid 0 \le W_{ji} \le \frac{4-\alpha}{\alpha(2+\alpha)n}, \sum_j W_{ji} = 1 \Big\}. \quad (27)$$

Standard optimization algorithms such as Frank-Wolfe allow us to solve (27) to any given precision with a polynomial number of calls to the linear optimization oracle guaranteed by Assumption 6.

While Algorithm 1 essentially minimizes the quantity $\|X - XW\|_2^2$, this new algorithm can be thought of as minimizing $\|X - XW\|_{2\to\psi}^2$. However, for general norms computing the $2 \to \psi$ norm is NP-hard, and so we rely on a κ-approximate solution by optimizing over $\mathcal{P}$. We are now ready to prove Theorem 7.

Proof (Theorem 7). The proof is similar to Proposition 16, so we only provide a sketch of the differences. First, the condition of Lemma 17 still holds, now with a equal to $\kappa n\sigma^2$ rather than $n\sigma^2$ due to the approximation ratio κ. (This is why we needed to change line 4.)

Next, we need to modify equations (20-23) to hold for the $2 \to \psi$ norm rather than operator norm:

$$\begin{aligned}
\|X_{\mathcal{A}} - Z\|_{2\to\psi} &= \|X_{\mathcal{A}}(I - W)(I - W_1)^{-1}\|_{2\to\psi} && (28)\\
&\overset{(i)}{\le} \|X_{\mathcal{A}}(I - W)\|_{2\to\psi}\|(I - W_1)^{-1}\|_{2\to 2} && (29)\\
&\le 10\|X_{\mathcal{A}}(I - W)\|_{2\to\psi} && (30)\\
&\le 10\sqrt{2}\|X_{\mathcal{A}}(I - W)\,\mathrm{diag}(c_{\mathcal{A}})^{1/2}\|_{2\to\psi} \le 20\sqrt{2\kappa n}\sigma. && (31)
\end{aligned}$$

Here (i) is from the general fact $\|AB\|_{2\to\psi} \le \|A\|_{2\to\psi}\|B\|_{2\to 2}$, and the rest of the inequalities follow for the same reasons as in (20-23).

We next need to modify equations (24-26). This can be done with the following inequality:

▶ **Lemma 18.** *For any matrix $A = [a_1 \ \cdots \ a_n]$ of rank r and any γ-strongly convex norm $\|\cdot\|_\psi$, we have $\sum_{i=1}^n \|a_i\|_\psi^2 \le \frac{r}{\gamma}\|A\|_{2\to\psi}^2$.*

This generalizes the inequality $\|A\|_F^2 \le \mathrm{rank}(A) \cdot \|A\|_2^2$. Using Lemma 18 (proved below), we have

$$\begin{aligned}
\sum_{i \in S \cap \mathcal{A}} \|z_i - \mu\|_\psi^2 &\le \tfrac{\mathrm{rank}(Z)+1}{\gamma}\|[z_i - \mu]_{i \in S\cap\mathcal{A}}\|_{2\to\psi}^2 && (32)\\
&\le \tfrac{\mathrm{rank}(Z)+1}{\gamma}\left(\|[z_i - x_i]_{i \in S\cap\mathcal{A}}\|_{2\to\psi} + \|[x_i - \mu]_{i \in S\cap\mathcal{A}}\|_{2\to\psi}\right)^2 && (33)\\
&= \mathcal{O}\Big(\tfrac{\kappa\sigma^2 n}{\alpha\gamma}\Big). && (34)
\end{aligned}$$

The inequalities again follow for the same reasons as before. If we choose z_i at random, with probability $\Omega(\alpha)$ we will output a z_i with $\|z_i - \mu\|_\psi = \mathcal{O}\left(\frac{\sigma\sqrt{\kappa}}{\alpha\sqrt{\gamma}}\right)$. This completes the first part of the proposition.

For the second part, by the same reasoning as before we obtain $\tilde{\mu}$ with $\|X_{\mathcal{A}} - \tilde{\mu}\mathbb{1}^\top\|_{2\to\psi} = \mathcal{O}(\sqrt{n\kappa}\sigma)$, which implies that $\mathcal{A}$ is resilient with $\sigma_*(\epsilon) = \mathcal{O}(\sigma\sqrt{\kappa\epsilon})$ for $\epsilon \le \frac{1}{2}$. The mean of $\mathcal{A}$ will therefore be within distance $\mathcal{O}(\sigma\sqrt{\kappa\epsilon})$ of μ as before, which completes the proof. ◀

We finish by proving Lemma 18.

Proof (Lemma 18). Let $s \in \{-1, +1\}^n$ be a uniformly random sign vector. We will compare $\mathbb{E}_s[\|As\|_\psi^2]$ in two directions. Let P be the projection onto the span of A. On the one hand, we have $\|As\|_\psi^2 = \|APs\|_\psi^2 \le \|A\|_{2\to\psi}^2\|Ps\|_2^2$, and hence $\mathbb{E}_s[\|As\|_\psi^2] \le \mathbb{E}_s[\|Ps\|_2^2]\|A\|_{2\to\psi}^2 = \mathrm{rank}(A)\|A\|_{2\to\psi}^2$. On the other hand, similarly to (11) we have $\mathbb{E}_s[\|As\|_\psi^2] \ge (1/\gamma)\sum_{i=1}^n \|a_i\|_\psi^2$ by the strong convexity of the norm $\|\cdot\|_\psi$. Combining these yields the desired result. ◀

5 Robust Low-Rank Recovery

In this section we present results on rank-k recovery. We first justify the definition of rank-resilience (Definition 8) by showing that it is information-theoretically sufficient for (approximately) recovering the best rank-k subspace. Then, we provide an algorithm showing that this subspace can be recovered efficiently.

5.1 Information-Theoretic Sufficiency

Let $X_S = [x_i]_{i\in S}$. Recall that δ-rank-resilience asks that $\mathrm{col}(X_T) = \mathrm{col}(X_S)$ and $\|X_T^\dagger X_S\|_2 \le 2$ for $|T| \ge (1-\delta)|S|$. This is justified by the following:

▶ **Proposition 19.** *Let $S \subseteq [n]$ be a set of points of size $(1-\delta)n$ that is $\frac{\delta}{1-\delta}$-rank-resilient. Then it is possible to output a rank-k projection matrix P such that $\|(I-P)X_S\|_2 \le 2\sigma_{k+1}(X_S)$.*

Proof. Find the $\frac{\delta}{1-\delta}$-rank-resilient set S' of size $(1-\delta)n$ such that $\sigma_{k+1}(X'_S)$ is smallest, and let P be the projection onto the top k singular vectors of $X_{S'}$. Then we have $\|(I-P)X_{S'}\|_2 = \sigma_{k+1}(X_{S'}) \le \sigma_{k+1}(X_S)$. Moreover, if we let $T = S \cap S'$, we have $\|(I-P)X_T\|_2 \le \|(I-P)X_{S'}\|_2 \le \sigma_{k+1}(X_S)$ as well. By pigeonhole, $|T| \ge (1-2\delta)n = (1-\frac{\delta}{1-\delta})|S|$. Therefore $\mathrm{col}(X_T) = \mathrm{col}(X_S)$, and $\|(I-P)X_S\|_2 = \|(I-P)X_TX_T^\dagger X_S\|_2 \le \|(I-P)X_T\|_2\|X_T^\dagger X_S\|_2 \le 2\sigma_{k+1}(X_S)$ as claimed. ◀

5.2 Efficient Recovery

We next provide an algorithm for efficient recovery, given as Algorithm 2 below. The proof that it satisfies the guarantees of Theorem 9 is deferred to the full paper.

6 Finite-Sample Concentration

In this section we provide two general finite-sample concentration results that establish resilience with high probability. The first holds for arbitrary resilient distributions but has suboptimal sample complexity, while the latter is specialized to distributions with bounded variance and has near-optimal sample complexity.

Algorithm 2 Algorithm for recovering a rank-k subspace.

1: Initialize $c_i = 1$ for all $i = 1, \dots, n$ and $\mathcal{A} = \{1, \dots, n\}$. Set $\lambda = \frac{(1-\delta)n\sigma^2}{k}$.
2: Let $Y \in \mathbb{R}^{d\times d}$ and $Q \in \mathbb{R}^{\mathcal{A}\times\mathcal{A}}$ be the maximizer/minimizer of the saddle point problem

$$\max_{\substack{Y\succeq 0,\\ \mathrm{tr}(Y)\le 1}} \min_{Q\in\mathbb{R}^{n\times n}} \sum_{i\in\mathcal{A}} c_i[(x_i - X_{\mathcal{A}}q_i)^\top Y(x_i - X_{\mathcal{A}}q_i) + \lambda\|q_i\|_2^2]. \tag{35}$$

3: Let $\tau_i^\star = (x_i - X_{\mathcal{A}}q_i)^\top Y(x_i - X_{\mathcal{A}}q_i) + \lambda\|q_i\|_2^2$.
4: **if** $\sum_{i\in\mathcal{A}} c_i\tau_i^\star > 8n\sigma^2$ **then**
5: For $i \in \mathcal{A}$, replace c_i with $(1 - \frac{\tau_i^\star}{\tau_{\max}})c_i$, where $\tau_{\max} = \max_{i\in\mathcal{A}} \tau_i^\star$.
6: For all i with $c_i < \frac{1}{2}$, remove i from $\mathcal{A}$.
7: Go back to line 3.
8: **end if**
9: Let Q_1 be the result of zeroing out all singular values of Q greater than 0.9.
10: Output $P = X_{\mathcal{A}}Q_0X_{\mathcal{A}}^\dagger$, where $Q_0 = (Q - Q_1)(I - Q_1)^{-1}$.

6.1 Concentration for Resilient Distributions

Our first result, stated as Proposition 4 in Section 1.2, applies to any (σ, ϵ)-resilient distribution p; recall that p is (σ, ϵ)-resilient iff $\|\mathbb{E}[x \mid E] - \mu\| \le \sigma$ for any event E of probability $1 - \epsilon$.

We define the *covering number* of the unit ball in a norm $\|\cdot\|_*$ to be the minimum M for which there are vectors $v_1, \dots, v_M$, each with $\|v_j\|_* \le 1$, such that $\max_{j=1}^M \langle x, v_j\rangle \ge \frac{1}{2}\sup_{\|v\|_*\le 1}\langle x, v\rangle$ for all vectors $v \in \mathbb{R}^d$. Note that $\log M$ is a measure of the effective dimension of the unit ball, i.e. $\log M = \Theta(d)$ if $\|\cdot\|_*$ is the ℓ_∞ or ℓ_2 norm, while $\log M = \Theta(\log d)$ for the ℓ_1 norm.

We recall Proposition 4 for convenience:

▶ **Proposition.** *Suppose that a distribution p is (σ, ϵ)-resilient around its mean μ with $\epsilon < \frac{1}{2}$. Let B be such that $\mathbb{P}[\|x - \mu\| \ge B] \le \epsilon/2$. Also let M be the covering number of the unit ball in the dual norm $\|\cdot\|_*$.*

Then, given n samples $x_1, \dots, x_n \sim p$, with probability $1 - \delta - \exp(-\epsilon n/6)$ there is a subset T of $(1-\epsilon)n$ of the x_i that is (σ', ϵ)-resilient with $\sigma' = \mathcal{O}\Big(\sigma\Big(1 + \sqrt{\frac{\log(M/\delta)}{\epsilon^2 n}} + \frac{(B/\sigma)\log(M/\delta)}{n}\Big)\Big)$.

Proof. Let p' be the distribution of samples from p conditioned on $\|x - \mu\| \le B$. Note that p' is $(\sigma, \frac{\epsilon}{2})$-resilient since every event with probability $1 - \epsilon/2$ in p' is an event of probability $(1 - \epsilon/2)^2 \ge 1 - \epsilon$ in p. Moreover, with probability $1 - \exp(-\epsilon n/6)$, at least $(1 - \epsilon)n$ of the samples from p will come from p'. Therefore, we can focus on establishing resilience of the $n' = (1 - \epsilon)n$ samples from p'.

With a slight abuse of notation, let $x_1, \dots, x_{n'}$ be the samples from p'. Then to check resilience we need to bound $\|\frac{1}{|T|}\sum_{i\in T}(x_i - \mu)\|$ for all sets T of size at least $(1 - \epsilon)n'$. We will first use the covering $v_1, \dots, v_M$ to obtain

$$\Big\|\frac{1}{|T|}\sum_{i\in T}(x_i - \mu)\Big\| \le 2\max_{j=1}^M \frac{1}{|T|}\sum_{i\in T}\langle x_i - \mu, v_j\rangle. \tag{36}$$

The idea will be to analyze the sum over $\langle x_i - \mu, v_j\rangle$ for a fixed v_j and then union bound over the M possibilities. For a fixed v_j, we will split the sum into two components: those with small magnitude (roughly σ/ϵ) and those with large magnitude (between σ/ϵ and B). We can then bound the former with Hoeffding's inequality, and using resilience we will be

able to upper-bound the second moment of the latter, after which we can use Bernstein's inequality.

More formally, let $\tau = \frac{1-\epsilon}{\epsilon/4}\sigma$ and define

$$y_i = \langle x_i - \mu, v_j\rangle \mathbb{I}[|\langle x_i - \mu, v_j\rangle| < \tau], \tag{37}$$

$$z_i = \langle x_i - \mu, v_j\rangle \mathbb{I}[|\langle x_i - \mu, v_j\rangle| \geq \tau]. \tag{38}$$

Clearly $y_i + z_i = \langle x_i - \mu, v_j\rangle$. Also, we have $|y_i| \leq \tau$ almost surely, and $|z_i| \leq B$ almost surely (because $x_i \sim p'$ and hence $\langle x_i - \mu, v_j\rangle \leq \|x_i - \mu\| \leq B$). The threshold τ is chosen so that z_i is non-zero with probability at most $\epsilon/2$ under p (see Lemma 3).

Now, for any set T of size at least $(1-\epsilon)n'$, we have

$$\frac{1}{|T|}\sum_{i\in T}\langle x_i - \mu, v_j\rangle = \frac{1}{|T|}\sum_{i\in T} y_i + z_i \tag{39}$$

$$\leq \Big|\frac{1}{|T|}\sum_{i\in T} y_i\Big| + \frac{1}{|T|}\sum_{i\in T}|z_i| \tag{40}$$

$$\leq \Big|\frac{1}{|T|}\sum_{i=1}^{n'} y_i\Big| + \Big|\frac{1}{|T|}\sum_{i\notin T} y_i\Big| + \frac{1}{|T|}\sum_{i=1}^{n'}|z_i| \tag{41}$$

$$\leq \frac{1}{1-\epsilon}\Big|\frac{1}{n'}\sum_{i=1}^{n'} y_i\Big| + \frac{\epsilon}{1-\epsilon}\tau + \frac{1}{(1-\epsilon)n'}\sum_{i=1}^{n'}|z_i|. \tag{42}$$

The last step uses the fact that $|y_i| \leq \tau$ for all i. It thus suffices to bound $|\frac{1}{n'}\sum_{i=1}^{n'} y_i|$ as well as $\frac{1}{n'}\sum_{i=1}^{n'}|z_i|$.

For the y_i term, note that by resilience $\|\mathbb{E}[y_i]\| \leq \sigma$ (since y_i is sampled from p conditioned on $|\langle x_i - \mu, v_j\rangle| < \tau$ and $\|x_i - \mu\| \leq B$, which each occur with probabiliy at least $1 - \epsilon/2$). Then by Hoeffding's inequality, $|\frac{1}{n'}\sum_{i=1}^{n'} y_i| = \mathcal{O}(\sigma + \tau\sqrt{\log(2/\delta)/n'})$ with probability $1-\delta$.

For the z_i term, we note that $\mathbb{E}[|z_i|] = \mathbb{E}[\max(z_i, 0)] + \mathbb{E}[\max(-z_i, 0)]$. Let τ' be the ϵ-quantile of $\langle x_i - \mu, v_j\rangle$ under p, which by Lemma 3 is at most τ. Then we have

$$\mathbb{E}_p[\max(z_i, 0)] = \mathbb{E}_p[\langle x_i - \mu, v_j\rangle \mathbb{I}[\langle x_i - \mu, v_j\rangle \geq \tau]] \tag{43}$$

$$\leq \mathbb{E}_p[\langle x_i - \mu, v_j\rangle \mathbb{I}[\langle x_i - \mu, v_j\rangle \geq \tau']] \tag{44}$$

$$\overset{(i)}{\leq} \epsilon \cdot \frac{1-\epsilon}{\epsilon}\sigma = (1-\epsilon)\sigma, \tag{45}$$

where (i) is again Lemma 3. Then we have $\mathbb{E}_{p'}[\max(z_i, 0)] \leq \frac{1}{1-\epsilon}\mathbb{E}_p[\max(z_i, 0)] \leq \sigma$, and hence $\mathbb{E}_{p'}[|z_i|] \leq 2\sigma$ (as $\mathbb{E}[\max(-z_i, 0)] \leq \sigma$ by the same argument as above). Since $|z_i| \leq B$, we then have $\mathbb{E}[|z_i|^2] \leq 2B\sigma$.

Therefore, by Bernstein's inequality, with probability $1-\delta$ we have

$$\frac{1}{n'}\sum_{i=1}^{n'}|z_i| \leq \mathcal{O}\Big(\sigma + \sqrt{\frac{\sigma B\log(2/\delta)}{n'}} + \frac{B\log(2/\delta)}{n'}\Big) = \mathcal{O}\Big(\sigma + \frac{B\log(2/\delta)}{n'}\Big). \tag{46}$$

Taking a union bound over the v_j for both y and z, and plugging back into (42), we get that $\frac{1}{|T|}\sum_{i\in T}\langle x_i - \mu, v_j\rangle \leq \mathcal{O}\big(\sigma + \frac{\sigma}{\epsilon}\sqrt{\frac{\log(2M/\delta)}{n}} + \frac{B\log(2M/\delta)}{n}\big)$ for all T and v_j with probability $1-\delta$. Plugging back into (36), we get that $\|\frac{1}{|T|}\sum_{i\in T}(x_i - \mu)\| \leq \mathcal{O}\big(\sigma + \frac{\sigma}{\epsilon}\sqrt{\frac{\log(2M/\delta)}{n}} + \frac{B\log(2M/\delta)}{n}\big)$, as was to be shown. ◀

6.2 Concentration Under Bounded Covariance

In this section we state a stronger but more restrictive finite-sample bound giving conditions under which samples have bounded variance. It is a straightforward extension of Proposition B.1 of [3], so we defer the proof to the full paper.

▶ **Proposition 20.** *Suppose that a distribution p has bounded variance in a norm $\|\cdot\|_*$: $\mathbb{E}_{x\sim p}[\langle x-\mu, v\rangle^2] \le \sigma^2\|v\|_*^2$ for all $v \in \mathbb{R}^d$. Then, given n samples $x_1, \ldots, x_n \sim p$, with probability $1-\exp(-\epsilon^2 n/16)$ there is a subset T of $(1-\epsilon)n$ of the points such that*

$$\frac{1}{|T|}\sum_{i\in T}\langle x_i-\mu, v\rangle^2 \le (\sigma')^2\|v\|_*^2 \text{ for all } v\in\mathbb{R}^d, \text{ where } (\sigma')^2 = \frac{4\sigma^2}{\epsilon}\left(1+\frac{d}{(1-\epsilon)n}\right). \quad (47)$$

This that whenever a distribution p on $\mathbb{R}^d$ has bounded variance, if $n \ge d$ samples x_i are drawn from p then some large subset of the samples will have bounded variance as well.

References

1 S. Balakrishnan, S. S. Du, J. Li, and A. Singh. Computationally efficient robust sparse estimation in high dimensions. In *Conference on Learning Theory (COLT)*, pages 169–212, 2017.

2 J. Batson, D. A. Spielman, and N. Srivastava. Twice-Ramanujan sparsifiers. *SIAM Journal on Computing*, 41(6):1704–1721, 2012.

3 M. Charikar, J. Steinhardt, and G. Valiant. Learning from untrusted data. In *STOC*, 2017.

4 A. Decelle, F. Krzakala, C. Moore, and L. Zdeborová. Asymptotic analysis of the stochastic block model for modular networks and its algorithmic applications. *Physical Review E*, 84(6), 2011.

5 I. Diakonikolas, G. Kamath, D. Kane, J. Li, A. Moitra, and A. Stewart. Robust estimators in high dimensions without the computational intractability. In *FOCS*, 2016.

6 I. Diakonikolas, G. Kamath, D. Kane, J. Li, A. Moitra, and A. Stewart. Being robust (in high dimensions) can be practical. *arXiv*, 2017.

7 I. Diakonikolas, G. Kamath, D. M. Kane, J. Li, A. Moitra, and A. Stewart. Robustly learning a Gaussian: Getting optimal error, efficiently. *arXiv*, 2017.

8 I. Diakonikolas, D. Kane, and A. Stewart. Robust learning of fixed-structure Bayesian networks. *arXiv*, 2016.

9 I. Diakonikolas, D. M. Kane, and A. Stewart. Statistical query lower bounds for robust estimation of high-dimensional Gaussians and Gaussian mixtures. *arXiv*, 2016.

10 I. Diakonikolas, D. M. Kane, and A. Stewart. Learning geometric concepts with nasty noise. *arXiv*, 2017.

11 U. Haagerup. The best constants in the khintchine inequality. *Studia Mathematica*, 70(3):231–283, 1981.

12 D. Kane, S. Karmalkar, and E. Price. Robust polynomial regression up to the information theoretic limit. *arXiv*, 2017.

13 A. Khintchine. Über dyadische brüche. *Mathematische Zeitschrift*, 18:109–116, 1923.

14 A. R. Klivans, P. M. Long, and R. A. Servedio. Learning halfspaces with malicious noise. *Journal of Machine Learning Research*, 10:2715–2740, 2009.

15 P. Kothari and J. Steinhardt. Better agnostic clustering via tensor norms. *arXiv*, 2017.

16 K. A. Lai, A. B. Rao, and S. Vempala. Agnostic estimation of mean and covariance. In *FOCS*, 2016.

17 J. Li. Robust sparse estimation tasks in high dimensions. *arXiv*, 2017.

18 L. Massoulié. Community detection thresholds and the weak Ramanujan property. In *STOC*, pages 694–703, 2014.

19 M. Meister and G. Valiant. A data prism: Semi-verified learning in the small-alpha regime. *arXiv*, 2017.

20 E. Mossel, J. Neeman, and A. Sly. A proof of the block model threshold conjecture. *arXiv*, 2013.

21 Y. Nesterov. Semidefinite relaxation and nonconvex quadratic optimization. *Optimization methods and software*, 9:141–160, 1998.

22 S. Shalev-Shwartz. *Online Learning: Theory, Algorithms, and Applications*. PhD thesis, The Hebrew University of Jerusalem, 2007.

23 J. Steinhardt. Does robustness imply tractability? A lower bound for planted clique in the semi-random model. *arXiv*, 2017.

24 J. Steinhardt, G. Valiant, and M. Charikar. Avoiding imposters and delinquents: Adversarial crowdsourcing and peer prediction. In *NIPS*, 2016.

25 H. Xu, C. Caramanis, and S. Mannor. Principal component analysis with contaminated data: The high dimensional case. *arXiv*, 2010.

Recovering Structured Probability Matrices*†

Qingqing Huang[1], Sham M. Kakade[2], Weihao Kong[3], and Gregory Valiant[4]

1 **Massachusetts Institute of Technology, Cambridge, MA, USA**
qqh@mit.edu
2 **University of Washington, Seattle, WA, USA**
am@cs.washington.edu
3 **Stanford University, Stanford, CA, USA**
whkong@stanford.edu
4 **Stanford University, Stanford, CA, USA**
valiant@stanford.edu

Abstract

We consider the problem of accurately recovering a matrix $\mathbb{B}$ of size $M \times M$, which represents a probability distribution over M^2 outcomes, given access to an observed matrix of "counts" generated by taking independent samples from the distribution $\mathbb{B}$. How can structural properties of the underlying matrix $\mathbb{B}$ be leveraged to yield computationally efficient and information theoretically optimal reconstruction algorithms? When can accurate reconstruction be accomplished in the sparse data regime? This basic problem lies at the core of a number of questions that are currently being considered by different communities, including building recommendation systems and collaborative filtering in the sparse data regime, community detection in sparse random graphs, learning structured models such as topic models or hidden Markov models, and the efforts from the natural language processing community to compute "word embeddings". Many aspects of this problem—both in terms of learning and property testing/estimation and on both the algorithmic and information theoretic sides—remain open.

Our results apply to the setting where $\mathbb{B}$ has a low rank structure. For this setting, we propose an efficient (and practically viable) algorithm that accurately recovers the underlying $M \times M$ matrix using $\Theta(M)$ samples (where we assume the rank is a constant). This linear sample complexity is optimal, up to constant factors, in an extremely strong sense: even testing basic properties of the underlying matrix (such as whether it has rank 1 or 2) requires $\Omega(M)$ samples. Additionally, we provide an even stronger lower bound showing that distinguishing whether a sequence of observations were drawn from the uniform distribution over M observations versus being generated by a well-conditioned Hidden Markov Model with two hidden states requires $\Omega(M)$ observations, while our positive results for recovering $\mathbb{B}$ immediately imply that $\Omega(M)$ observations suffice to *learn* such an HMM. This lower bound precludes sublinear-sample hypothesis tests for basic properties, such as identity or uniformity, as well as sublinear sample estimators for quantities such as the entropy rate of HMMs.

1998 ACM Subject Classification I.2.6 Learning

Keywords and phrases Random matrices, matrix recovery, stochastic block model, Hidden Markov Models

Digital Object Identifier 10.4230/LIPIcs.ITCS.2018.46

* Sham Kakade acknowledges funding from the Washington Research Foundation for innovation in Data-intensive Discovery. Gregory and Weihao's contributions were supported by NSF CAREER Award CCF-1351108, and a Sloan Research Fellowship.

† full version at https://arxiv.org/abs/1602.06586

1 Introduction

Consider an unknown $M \times M$ matrix of probabilities $\mathbb{B}$, satisfying $\sum_{i,j} \mathbb{B}_{i,j} = 1$. Suppose one is given N independently drawn (i, j)-pairs, sampled according to the distribution defined by $\mathbb{B}$. How many draws are necessary to accurately recover $\mathbb{B}$? What can one infer about the underlying matrix based on these samples? How can one accurately test whether the underlying matrix possesses certain properties of interest? How do structural assumptions on $\mathbb{B}$ — for example, the assumption that $\mathbb{B}$ has low rank — affect the information theoretic or computational complexity of these questions? For the majority of these tasks, we currently lack both a basic understanding of the computational and information theoretic lay of the land, as well as algorithms that seem capable of achieving the information theoretic or computational limits.

This general question of making accurate inferences about a matrix of probabilities, given a matrix of observed "counts" of discrete outcomes, lies at the core of a number of problems that disparate communities have been tackling independently. On the theoretical side, these problems include both work on community detection in stochastic block models (where the goal is to infer the community memberships from an adjacency matrix of a graph that has been drawn according to an underlying matrix of probabilities expressing the community structure) as well as the line of work on recovering topic models, hidden Markov models (HMMs), and richer structured probabilistic models (where the model parameters can often be recovered using observed count data). On the practical side, these problems include work on computing low-rank approximations to sparsely sampled data, which arise in collaborative filtering and recommendation systems, as well as the recent work from the natural language processing community on understanding matrices of word co-occurrence counts for the purpose of constructing good "word embeddings". Additionally, work on latent semantic analysis and non-negative matrix factorization can also be recast in this setting.

In this work, we focus on this estimation problem where the probability matrix $\mathbb{B}$ possesses a particular low rank structure. While this estimation problem is rather specific, it generalizes the basic community detection problem and the problem of learning various common models encountered in natural language processing such as *probabilistic latent semantic analysis* [28]. Additionally, this problem encompasses the main technical challenge behind learning HMMs and topic models, in the sense that after $\mathbb{B}$ is accurately recovered, these learning problems have a number of parameters that is a function only of the number of topics/hidden states (which bounds the rank of $\mathbb{B}$ and is, in practical applications, at most a few hundred) as opposed to the the dictionary/alphabet size, M, which, in natural language settings is typically tens of thousands. Furthermore, this low rank case also provides a means to study how the relationships between property testing and estimation problems differ between this structured setting and the basic rank 1 setting that is equivalent to simply drawing i.i.d samples from a distribution supported on M elements.

We focus on the estimation of a low rank probability matrix $\mathbb{B}$ in the sparse data regime, near the information theoretic limit. In many practical scenarios involving sample counts, we seek algorithms capable of extracting the underlying structure in the sparsely sampled regime. To give two motivating examples, consider forming the matrix of word co-occurrences—the matrix whose rows and columns are indexed by the set of words, and whose (i, j)-th element consists of the number of times the i-th word follows the j-th word in a large corpus of text. In this context, the underlying probability matrix, $\mathbb{B}$, represents the distribution of bi-grams encountered in written english. In the context of recommendation system, one could consider a low rank matrix model, where the rows are indexed by customers, and the columns are indexed by products, with the (i, j)-th entry corresponding to the number of

times the i-th customer has purchased the j-th product. Here, the underlying probability matrix, $\mathbb{B}$, models the distribution from which each customer/product purchase is drawn. In both settings, the structure of the probability matrix underlying these observed counts contains insights into the two domains, and in both domains we only have relatively sparse data. This is inherent in many other natural scenarios involving heavy-tailed distributions (including genomic settings), where despite having massive datasets, a significant fraction of the domain is observed only a single time.

Similar estimation questions have been actively studied in the community detection literature, where the objective is to accurately recover the communities in the regime where the average degree (e.g. the row sums of the adjacency matrix) are constant. In contrast, the recent line of works for recovering highly structured models (such as topic models, HMMs, etc.) are only applicable to the *over-sampled* regime where the amount of data is well beyond the information theoretic limits. In these cases, achieving the information theoretic limits remains a widely open question. This work begins to bridge the divide between these recent algorithmic advances in both communities. We hope that the low rank probability matrix setting considered here serves as a jumping-off point for the more general questions of developing information theoretically optimal algorithms for estimating structured matrices and tensors in general, or recovering low-rank approximations to arbitrary probability matrices, in the sparse data regime. While the general settings are more challenging, we believe that some of our algorithmic techniques can be fruitfully extended.

In addition to developing algorithmic tools which we hope are applicable to a wider class of problems, a second motivation for considering this particular low rank case is that, with respect to distribution learning and property testing, the entire lay-of-the-land seems to change completely when the probability matrix $\mathbb{B}$ has rank larger than 1. In the rank 1 setting — where a sample consists of 2 *independent* draws from a distribution supported on $\{1, \ldots, M\}$ — the distribution can be learned using $\Theta(M)$ draws. Nevertheless, many properties of interest can be tested or estimated using a sample size that is *sublinear* in M[1]. However, even just in the case where the probability matrix is of rank 2, although the underlying matrix $\mathbb{B}$ can be represented with $O(M)$ parameters (and, as we show, it can also be accurately and efficiently recovered with $O(M)$ sample counts), sublinear sample property testing and estimation is generally impossible. This result begs a more general question: *what conditions must be true of a structured statistical setting in order for property testing to be easier than learning?*

1.1 Problem Formulation

We consider the following problem setup and notation:

- A vocabulary consisting of M "words", denoted by $\mathcal{M} = \{1, \ldots, M\}$.
- A low rank probability matrix $\mathbb{B}$, of size $M \times M$, with the following structure: $\mathbb{B} = \mathbb{P}\mathbb{W}\mathbb{P}^\top$, where $\mathbb{P}$ is an $M \times r$ non-negative matrix with column sums 1, and $\mathbb{W}$ is p.s.d. with $\sum_{i,j} \mathbb{W}_{i,j} = 1$.
- A set of N independent (i, j) pairs drawn according to $\mathbb{B}$, with the probability of drawing (i, j) given by $\mathbb{B}_{i,j}$.
- An $M \times M$ matrix of "counts", C, summarizing the frequencies of each (i, j) pair in the N draws.

[1] Distinguishing whether a distribution is uniform versus far from uniform can be accomplished using only $O(\sqrt{M})$ draws, testing whether two sets of samples were drawn from similar distributions can be done with $O(M^{2/3})$ draws, estimating the entropy of the distribution to within an additive ϵ can be done with $O(\frac{M}{\epsilon \log M})$ draws, etc.

Throughout, we will make frequent use of the Poissonization technique whereby we assume that the number of draws follows a Poisson distribution of expectation N. This renders $C_{i,j}$ independent of the other entries of the count matrix, simplifying analysis. Additionally, for both upper and lower bounds, with all but inverse exponential probability the $o(N)$ discrepancy between N and $Poi(N)$ contributes only to lower order terms.

Notation

Throughout the paper, we use the following standard shorthand notations. Denote $[n] \triangleq \{1, \ldots, n\}$. $\mathcal{I}$ denotes a subset of indices in $\mathcal{M}$. For a M-dimensional vector x, we use vector $x_{\mathcal{I}}$ to denote the elements of x restricted to the indices in $\mathcal{I}$; for two index sets $\mathcal{I}$, $\mathcal{J}$, and a $M \times M$ dimensional matrix X, we use $X_{\mathcal{I}\times\mathcal{J}}$ to denote the submatrix of X with rows restricting to indices in $\mathcal{I}$ and columns restricting to indices in $\mathcal{J}$.

We use $\mathrm{Poi}(\lambda)$ to denote a Poisson distribution with expectation λ; we use $\mathrm{Ber}(p)$ to denote a Bernoulli random variable with success probability $p \in [0, 1]$; and for a probability vector $x \in [0, 1]^M$ satisfying $\sum_i x_i = 1$ and an integer t, we use $\mathrm{Mul}(x; t)$ to denote the multinomial distribution over M outcomes corresponding to t draws from $[M]$ according to the distribution specified by the vector x.

1.2 Main Results

Our main result is the accurate recovery of a rank R matrix of the form described above in the linear data regime $N = O(M)$:

▶ **Theorem 1** (Upper bound for rank R, constant accuracy)**.** *Suppose we have access to N i.i.d. samples generated according to the a probability matrix $\mathbb{B} = \mathbb{P}\mathbb{W}\mathbb{P}^T$ with $\mathbb{P}$ an $M \times R$ nonnegative matrix with column sum 1, $\mathbb{W}$ an $R \times R$ p.s.d. matrix with entries summing to 1 and row sums bounded by $\sum_j \mathbb{W}_{i,j} \geq w_{min}$. For any constants $\epsilon > 0, \delta > 0$ and $N = \Theta(\frac{MR^2}{w_{min}^2 \epsilon^5} \log(1/\delta))$, there is an algorithm with poly$(M, \log(1/\delta))$ runtime that returns a rank R matrix $\widehat{\mathbb{B}}$ such that with probability at least $1 - \delta$:*

$$\|\widehat{\mathbb{B}} - \mathbb{B}\|_{\ell_1} \leq \epsilon.$$

We emphasize that our recovery is in terms of ℓ_1 distance, namely the total variation distance between the true distribution and the recovered distribution. In settings where there is a significant range in the row (or column) sums of $\mathbb{B}$, a spectral error bound might not be meaningful.

Much of the the difficulty in the algorithm is overcoming the fact that the row/column sums of $\mathbb{B}$ might be very non-uniform. Nevertheless, our result can be compared to the community detection setting with R communities (for which the row/column sums are completely uniform), for which accurate recovery can be efficiently achieved given $N = \Theta(MR^2)$ samples [20]. In our more general setting, we incur an extra factor of w_{min}^{-1}, whose removal might be possible with a more careful analysis of our approach.

1.2.1 Topic Models and Hidden Markov Models

One of the motivations for considering low rank structure of a probability matrix $\mathbb{B}$ is that this structure captures the structure of the matrix of expected bigrams generated by topic models [46, 28] and HMMs, as described below.

▶ **Definition 2.** An R-*topic model* over a vocabulary of size M is defined by a set of R distributions, $p^{(1)}, \ldots, p^{(R)}$ supported over M words, and a set of R corresponding topic *mixing weights* $w_1, \ldots, w_R$ with $\sum_i w_i = 1$. The process of drawing a bigram (i, j) consists of first randomly picking a topic $i \in [R]$ according to the distribution defined by the mixing weights, and then drawing two independent words from the distribution $p^{(i)}$ corresponding to the selected topic, i. Thus the probability of drawing a bigram (i, j) is $\sum_{k=1}^{R} w_R p^{(k)}(i) p^{(k)}(j)$, and the underlying distribution $\mathbb{B}$ over (i, j) pairs can be expressed as $\mathbb{B} = \mathbb{P}\mathbb{W}\mathbb{P}^\top$ with $\mathbb{P} = [p^{(1)}, \ldots, p^{(R)}]$, and $\mathbb{W} = diag(w_1, \ldots, w_R)$.

In the case of topic models, the decomposition of the matrix of bigram probabilities $\mathbb{B} = \mathbb{P}\mathbb{W}\mathbb{P}^\top$ has the desired form required by our Theorem 1, with $\mathbb{W}$ nonnegative and p.s.d., and hence the theorem guarantees an accurate recovery of $\mathbb{B}$, even in the sparse data regime. The recovery of the mixing weights $\{w_i\}$ and topic distributions $\{p^{(i)}\}$ from $\mathbb{B}$ requires an additional step, which will amount to solving a system of quadratic equations. Crucially, however, given the rank R matrix $\mathbb{B}$, the remaining problem becomes a problem only involving R^2 parameters—representing a linear combination of the R factors of $\mathbb{B}$ for each $p^{(i)}$—rather than recovering MR parameters.

▶ **Definition 3.** A *Hidden Markov model* with R hidden states and observations over an alphabet of size M is defined by an $R \times R$ transition matrix T, and R observation distributions $p^{(1)}, \ldots, p^{(R)}$. A sequence of observations is sampled as follows: select an initial state (e.g. according to the stationary distribution of the chain) then evolve the Markov chain according to the transition matrix T, drawing an observation from the ith distribution $p^{(i)}$ at each timestep in which the underlying chain is in state ith.

Assuming the Markov chain has stationary distribution $\pi_1, \ldots, \pi_R$, the probability of seeing a bigram (i, j) with symbol i observed at the kth timestep and symbol j observed at the $k + 1$st timestep, tends towards the following (i.e. assuming the chain is close to mixing by timestep k) rank R probability matrix $\mathbb{B} = \mathbb{P}\mathbb{W}\mathbb{P}^\top$, with $\mathbb{P} = [p^{(1)}, \ldots, p^{(R)}]$ and $\mathbb{W} = diag(\pi_1, \ldots, \pi_n)T$.

For HMMs, the low rank matrix of bigrams, $\mathbb{B} = \mathbb{P}\mathbb{W}\mathbb{P}^\top$, does *not* necessarily have the required form—specifically the mixing matrix $\mathbb{W}$ may not be p.s.d.—and it is unclear whether our approach can successfully recover such matrices. Nevertheless, with slightly more careful analysis, at least in certain cases the techniques yield tight results. For example, in the setting of an HMM with two hidden states, over an alphabet of size M, we can easily show that our techniques obtain an accurate reconstruction of the corresponding probability matrix $\mathbb{B}$, and then leverage that reconstruction together with a constant amount of tri-gram information to accurately learn the HMM:

▶ **Proposition 4.** *(Learning 2-state HMMs) Consider a sequence of observations given by a Hidden Markov Model with two hidden states and symmetric transition matrix with entries bounded away from 0. Assuming a constant ℓ_1 distance between the distributions of observations corresponding to the two states, there exists an algorithm which, given a sampled chain of length $N = \Omega(M/\epsilon^2)$, runs in time poly(M) and returns estimates of the transition matrix and two observation distributions that are accurate in ℓ_1 distance, with probability at least* $2/3$.

This probability of failure can be trivially boosted to $1 - \delta$ at the expense of an extra factor of $\log(1/\delta)$ observations.

1.2.2 Testing vs. Learning

Theorem 1 and Proposition 4 are tight in an extremely strong sense: for both the topic model and HMM settings, it is information theoretically impossible to perform even the most basic property tests using fewer than $\Theta(M)$ samples. For topic models, the community detection lower bounds [43][34][55] imply that $\Theta(M)$ bigrams are necessary to even distinguish between the case that the underlying model is the uniform distribution over bigrams versus the case of a R-topic model in which each topic has a unique subsets of M/R words with a constant fraction higher probability than the remaining words. More surprisingly, for k-state HMMs with $k \geq 2$, even if we permit an estimator to have more information than merely bigram counts, namely access to the *full sequence* of observations, we prove the following linear lower bound.

▶ **Theorem 5.** *There exists a constant $c > 0$ such that for sufficiently large M, given a sequence of observations from a HMM with two states and emission distributions p, q supported on M elements, even if the underlying Markov process is symmetric, with transition probability $1/4$, it is information theoretically impossible to distinguish the case that the two emission distributions, $p = q = Unif[M]$ from the case that $||p - q||_1 = 1$ with probability greater than $2/3$ using a sequence of fewer than cM observations.*

This immediately implies the following corollary for estimating the *entropy rate* of an HMM.

▶ **Corollary 6.** *There exists an absolute constant $c > 0$ such that given a sequence of observations from a HMM with two hidden states and emission distributions supported on M elements, a sequence of cM observations is information theoretically necessary to estimate the entropy rate to within an additive 0.5 with probability of success greater than $2/3$.*

These strong lower bounds for property testing and estimation are striking for several reasons. First, the core of our learning algorithm for 2-state HMMs (Proposition 4) is a matrix reconstruction step that uses only the set of bigram counts. Conceivably, it might be helpful to consider longer sequences of observations — even for HMMs that mix in constant time, there are detectable correlations between observations separated by $O(\log M)$ steps. Regardless, our lower bound shows that actually no additional information from such longer k-grams can be leveraged to yield sublinear sample property testing or estimation.

A second notable point is the apparent brittleness of sublinear property testing and estimation as we deviate from the standard (unstructured) i.i.d sampling setting. Indeed for nearly all distributional property estimation or testing tasks, including testing uniformity and estimating the entropy, sublinear-sample testing and estimation is possible in the i.i.d. sampling setting (e.g. [26, 52, 51]). In contrast to the i.i.d. setting in which estimation and testing require asymptotically fewer samples than *learning*, as the above results illustrate, even in the setting of an HMM with just two hidden states, learning and testing require comparable numbers of observations.

1.3 Related Work

As mentioned earlier, the general problem of reconstructing an underlying matrix of probabilities given access to a count matrix drawn according to the corresponding distribution, lies at the core of questions that are being actively pursued by several different communities. We briefly describe these questions, and their relation to the present work.

Community Detection. With the increasing prevalence of large scale social networks, there has been a flurry of activity from the algorithms and probability communities to both model structured random graphs, and understand how (and when it is possible) to examine a graph and infer the underlying structures that might have given rise to the observed graph. One of the most well studied community models is the *stochastic block model* [29]. In its most basic form, this model is parameterized by a number of individuals, M, and two probabilities, α, β. The model posits that the M individuals are divided into two equal-sized "communities", and such a partition defines the following random graph model: for each pair of individuals in the same community, the edge between them is present with probability α (independently of all other edges); for a pair of individuals in different communities, the edge between them is present with probability $\beta < \alpha$. Phrased in the notation of our setting, the adjacency matrix of the graph is generated by including each potential edge (i, j) independently, with probability $\mathbb{B}_{i,j}$, with $\mathbb{B}_{i,j} = \alpha$ or β according to whether i and j are in the same community. Note that $\mathbb{B}$ has rank 2 and is expressible as $\mathbb{B} = PWP^\top$ where $P = [p, q]$ for vectors $p = \frac{2}{M} I_1$ and $q = \frac{2}{M} I_2$ where I_1 is the indicator vector for membership in the first community, and I_2 is defined analogously, and W is the 2×2 matrix with $\alpha \frac{M^2}{4}$ on the diagonal and $\beta \frac{M^2}{4}$ on the off-diagonal.

What values of α, β, and M enable the community affiliations of all individuals to be accurately recovered with high probability? What values of α, β, and M allow for the graph to be distinguished from an Erdos-Renyi random graph (that has no community structure)? The crucial regime is where $\alpha, \beta = O(\frac{1}{M})$, and hence each person has a constant, or logarithmic expected degree. The naive spectral approaches will fail in this regime, as there will likely be at least one node with degree $\approx \log M / \log \log M$, which will ruin the top eigenvector. Nevertheless, in a sequence of works sparked by the paper of Friedman, and Szemeredi [24], the following punchline has emerged: the naive spectral approach will work, even in the constant expected degree setting, provided one first either removes, or at least diminishes the weight of these high-degree problem vertices (e.g. [23, 33, 42, 34, 35]). For both the *exact* recovery problem and the detection problem, the exact tradeoffs between α, β, and M were recently established, down to subconstant factors [43, 1, 38]. More recently, there has been further research investigating more complex stochastic block models, consisting of three or more components, components of unequal sizes, etc. (see e.g. [20, 2, 3]).

The community detection setting generates an adjacency matrix with entries in $\{0, 1\}$, choosing entry $C_{i,j} \leftarrow Bernoulli(\mathbb{B}_{i,j})$, as opposed to our setting where $C_{i,j}$ is drawn from the corresponding Poisson distribution. Nevertheless, the two models are extremely similar in the sparse regime considered in the community detection literature, since, when $\mathbb{B}_{i,j} = O(1/M)$, the corresponding Poisson and Bernoulli distributions have total variation distance $O(1/M^2)$.

Word Embeddings. On the more applied side, some of the most impactful advances in natural language processing over the past five years has been work on "word embeddings" [39, 37, 49, 10]. The main idea is to map every word w to a vector $v_w \in \mathbb{R}^d$ (typically $d \approx 500$) in such a way that the geometry of the vectors captures the semantics of the word.[2] One of the main constructions for such embeddings is to form the $M \times M$ matrix whose rows/columns are indexed by words, with (i, j)-th entry corresponding to the total number of times the i-th and j-th word occur next to (or near) each other in a large corpus of text (e.g. wikipedia).

[2] The goal of word embeddings is not just to cluster similar words, but to have semantic notions encoded in the geometry of the points: the example usually given is that the direction representing the difference between the vectors corresponding to "king" and "queen" should be similar to the difference between the vectors corresponding to "man" and "woman", or "uncle" and "aunt", etc.

The word embedding is then computed as the rows of the singular vectors corresponding to the top rank d approximation to this empirical count matrix.[3] These embeddings have proved to be extremely effective, particularly when used as a way to map text to features that can then be trained in downstream applications. Despite their successes, current embeddings seem to suffer from sampling noise in the count matrix (where many transformations of the count data are employed, e.g. see [48])—this is especially noticeable in the relatively poor quality of the embeddings for relatively rare words. The theoretical work [11] sheds some light on why current approaches are so successful, yet the following question largely remains: Is there a more accurate way to recover the best rank-d approximation of the underlying matrix than simply computing the best rank-d approximation for the (noisy) matrix of empirical counts?

Efficient Algorithms for Latent Variable Models. There is a growing body of work from the algorithmic side (as opposed to information theoretic) on how to recover the structure underlying various structured statistical settings. This body of work includes work on learning HMMs [31, 41, 19], recovering low-rank structure [9, 8, 15], and learning or clustering various structured distributions such as Gaussian mixture models [21, 54, 40, 14, 30, 32, 25]. A number of these methods essentially can be phrased as solving an inverse moments problem, and the work in [7] provides a unifying viewpoint for computationally efficient estimation for many of these models under a tensor decomposition perspective. In general, this body of work has focused on the computational issues and has considered these questions in the regime in which the amount of data is plentiful—well above the information theoretic limits.

On the practical side, the natural language processing community has considered a variety of generative and probabilistic models that fall into the framework we consider. These include work on *probabilistic latent semantic analysis* (see e.g. [28, 22]), including the popular *latent Dirichlet allocation* topic model [18]. Much of the algorithmic work on recovering these models is either of a heuristic nature (such as the EM framework), or focuses on computational efficiency in the regime in which data is plentiful (e.g. [6].

Sublinear Sample Testing and Estimation. In contrast to the work described in the previous section on efforts to devise computationally efficient algorithms for tackling complex structural settings in the "over–sampled" regime, there is also significant work establishing information theoretically optimal algorithms and (matching) lower bounds for estimation and distributional hypothesis testing in the most basic setting of independent samples drawn from (unstructured) distributions. This work includes algorithms for estimating basic statistical properties such as entropy [45, 27, 50, 52], support size [47, 50], distance between distributions [50, 52, 51], and various hypothesis tests, such as whether two distributions are very similar, versus significantly different [26, 12, 44, 53, 16], etc. While many of these results are optimal in a worst-case ("minimax") sense, there has also been recent progress on instance optimal (or "competitive") estimation and testing, e.g. [4, 5, 53], with stronger information theoretic optimality guarantees. There has also been a long line of work beginning with [17, 13] on these tasks in "simply structured" settings, e.g. where the domain of the distribution has a total ordering or where the distribution is monotonic or unimodal.

[3] A number of pre-processing steps have been considered, including taking the element-wise square roots of the entries, or logarithms of the entries, prior to computing the SVD.

2 Recovery Algorithm

To motivate our algorithms, it will be helpful to first consider the more naive approaches. Recall that we are given N samples drawn according to the probability matrix $\mathbb{B}$, with C denoting the matrix of empirical counts. By the Poisson assumption on sample size, we have that $C_{i,j} \sim \mathrm{Poi}(N\mathbb{B}_{i,j})$. Perhaps the most naive hope is to consider the rank R truncated SVD of the empirical matrix $\frac{1}{N}C$, which concentrates to $\mathbb{B}$ in Frobenius norm at $\frac{1}{\sqrt{N}}$ rate. Unfortunately, in order to achieve constant ℓ_1 error, this approach would require a sample complexity as large as $\Theta(M^2)$. Intuitively, this is because the rows and columns of C corresponding to words with larger marginal probabilities have higher row and column sums in expectation, as well as higher variances that undermine the spectral concentration of the matrix as a whole.

The above observation leads to the idea of pre-scaling the matrix so that every word (i.e. row/column) roughly has equal variance. Indeed, with the pre-scaling modification of the truncated SVD, one can likely improve the sample complexity of this approach to $\Theta(M \log M)$. To further reduce the sample complexity, it is worth considering what prevents the truncated SVD from achieving accurate recovery in the $N = \Theta(M)$ regime. Suppose the word marginals are roughly uniform, namely all in the order of $O(\frac{1}{M})$, the linear sample regime roughly corresponds to the stochastic block model setup where the expected row sums are all of order $d = \frac{N}{M} = \Omega(1)$. It is well-known that in this sparse regime, the adjacency matrix (in the graph setting), or the empirical count matrix C in our problem, does not concentrate to the expectation matrix in the spectral sense. Due to heavy rows/columns of sum $\Omega(\frac{\log M}{\log \log M})$, the leading eigenvectors are polluted by the local properties of these heavy rows/columns and do not reveal the global structure of the matrix/graph, which is precisely the desired information.

Fortunately, these heavy (empirical) rows/columns are the *only* impediment to spectral concentration in the linear sample size regime. Provided all rows/columns with observed weight significantly more than d are zeroed out, spectral concentration prevails. This simple idea of taming the heavy rows/columns was first introduced by [24], and analyzed in [23] and many other works. Recently in [35] and [36], the authors provided clean and clever proofs to show that *any* manner of "regularization"—removing entries from the heavy rows/columns until their row/column sums are bounded—essentially leads to the desired spectral concentration for the adjacency matrix of random graphs whose row/column sums are roughly uniform in expectation.

The challenge of applying this regularization approach in our more general setting is that the row/column expectations of C might be extremely non-uniform. If we try to "regularize", we will not know whether we are removing entries from rows that have small expected sum but happened to have a few extra entries, or if we are removing entries from a row that actually has a large expected sum (in which case such removal will be detrimental).

Our approach is to partition the vocabulary $\mathcal{M}$ into bins that have roughly uniform marginal probabilities, corresponding to partitioning the rows/columns into sets that have roughly equal (empirical) counts. Restricting our attention to the diagonal sub-blocks of $\mathbb{B}$ whose rows/columns consist of indices restricted to a single bin, the expected row and column sums are now roughly uniform. We can regularize (by removing abnormally heavy rows and columns) from each diagonal block separately to restore spectral concentration on each of these sub blocks. Now, we can apply truncated SVD to each diagonal sub block, recovering the column span of these blocks of $\mathbb{B}$. With the column spans of each bin, we can now "stitch" them together as a single large projection matrix P which has rank at most R

Algorithm 1: The algorithm to which Theorem 1 applies, which recovers rank R probability matrices in the linear data regime.

Input: $3N$ i.i.d. samples from the distribution $\mathbb{B}$ of dimension $M \times M$, where $N = O(\frac{MR^2}{w_{min}^2 \epsilon^5})$

(In each of the 3 steps, B refers to an independent copy of the normalized count matrix $\frac{1}{N}C$.)

Output: Rank R estimator $\widehat{\mathbb{B}}$ for $\mathbb{B}$

Step 1. (**Binning according to the empirical marginal probabilities**)

Set $\widehat{\rho}_i = \frac{\sum_{j=1}^{M}(C_{i,j}+C_{j,i})}{2N}$. Partition the vocabulary $\mathcal{M}$ into:

$$\mathcal{I}_0 = \left\{ i : \widehat{\rho}_i < \frac{1}{N} \right\}, \text{ and } \mathcal{I}_k = \left\{ i : \frac{e^{k-1}}{N} \leq \widehat{\rho}_i \leq \frac{e^k}{N} \right\}, \text{ for } k = 1, \ldots, \log N.$$

Sort the M words according to $\widehat{\rho}_i$ in ascending order. Define $\bar{\rho}_k = \frac{e^{k+1}}{N}$. For each bin $\mathcal{I}_k$, if $|\mathcal{I}_k| < 20e^{-\frac{3}{2}(k+1)}N$ set $\bar{\rho}_k$ to be 0. Let $k_0 = 4\log(\frac{c_0 R}{\epsilon\sqrt{w_{min}}}) + 16$, for an absolute constant c_0 which will be specified in the analysis, and set $\bar{\rho}_k$ to be 0 for all $k < k_0$. Define the following block diagonal matrix:

$$D = \begin{bmatrix} \bar{\rho}_1^{1/2} I_{|\mathcal{I}_1|} & & \\ & \ddots & \\ & & \bar{\rho}_{\log N}^{1/2} I_{|\mathcal{I}_{\log N}|} \end{bmatrix}. \tag{1}$$

Step 2. (**Estimate dictionary span in each bin**)

For each diagonal block $B_k = B_{\mathcal{I}_k \times \mathcal{I}_k}$, perform the following two steps:

1. (**Regularization**):

- If a row/column of B has sum exceeding $2\bar{\rho}_k$, set the entire row/column to 0.
- If a row/column of B_k has sum exceeding $\frac{2|\mathcal{I}_k|\bar{\rho}_k^2}{w_{min}}$, set the entire row/column to 0.

Denote the regularized block by $\widetilde{B}_k$.

2. (R-**SVD**): Define the $|\mathcal{I}_k| \times R$ matrix V_k to consist of the R top singular vectors of $\widetilde{B}_k$.

Step 3. (**Recover estimate for $\widehat{\mathbb{B}}$ accurate in ℓ_1**)

Define the following projection matrix:

$$P_V = \begin{bmatrix} P_{V_1} & & \\ & \ddots & \\ & & P_{V_{\log M}} \end{bmatrix}, \text{ where } P_{V_k} = V_k V_k^T. \tag{2}$$

Let $\widehat{\mathbb{B}}'$ be the rank-R truncated SVD of matrix $P_V D^{-1} B D^{-1} P_V$, and return $\widehat{\mathbb{B}} = D\widehat{\mathbb{B}}'D$.

times the number of bins, and roughly contains the column span of $\mathbb{B}$. We then project a new count matrix, C', obtained via a fresh partition of samples. As the projection is fairly low rank, it filters most of the sampling noise, leaving an accurate approximation of $\mathbb{B}$.

We summarize these basic ideas of Algorithm 1.

1. Given a batch of N samples, group words according to the empirical marginal probabilities,

so that in each bin consists of words whose (empirical) marginal probabilities, differ by at most a constant factor.

2. Given a second batch of N samples, zeros out the words that have abnormally large empirical marginal probabilities comparing to the expected marginal probabilities of words in their bin. Then consider the diagonal blocks of the empirical bigram counts matrix C, with rows and columns corresponding to the words in the same bin. We "regularize" each diagonal block in the empirical matrix by removing abnormally heavy rows and columns of the blocks, and then apply truncated SVD to estimate the column span of that diagonal block of $\mathbb{B}$.
3. With a third batch of N samples, project the empirical count matrix into the "stitched" column spans recovered in the previous step which yields an accurate estimate of $\mathrm{Diag}(\rho)^{-1/2}\mathbb{B}\mathrm{Diag}(\rho)^{-1/2}$ in spectral norm, where ρ denotes the vector of marginal probabilities. Since the estimate is accurate in spectral norm *after* scaling by the marginal probabilities, this spectral concentration of the scaled matrix easily translates into an ℓ_1 error bounds for the un-scaled matrix $\mathbb{B}$, as desired.

There are several potential concerns that arise in implementing the above high-level algorithm outline and establishing the correctness of the algorithm:

1. We do not have access to the exact marginal probabilities of each word. With a linear sample size, the recovered vector of marginal probabilities has only constant (expected) accuracy in ℓ_1 norm. Hence each bin, defined in terms of the empirical marginals, includes some non-negligible fraction of words with significantly larger (or smaller) marginal probabilities. When directly applied to the empirical bins with such "spillover" words, the existing results of "regularization" in [36] do not lead to the desired concentration result.
2. When we restrict our analysis to a diagonal block corresponding to a single bin, we throw away all the sample counts outside of that block. This greatly reduces the effective sample size, since a significant fraction of a word's marginal probability might be due to co-occurrences with words outside of its bin. It is not obvious that we retain enough samples in each diagonal block to guarantee meaningful estimation. [If the mixing matrix $\mathbb{W}$ in $\mathbb{B} = \mathbb{P}\mathbb{W}\mathbb{P}^\top$ is not p.s.d., this effect may be sufficiently severe so as to render these diagonal blocks essentially empty, foiling this approach.]
3. Finally, even if the "regularization" trick works for each diagonal block, we need to extract the useful information and "stitch" together this information from each block to provide an estimator for the entire matrix, including the off-diagonal blocks. Fortunately, the p.s.d assumption of the mixing matrix W ensures that sufficient information is contained in these diagonal blocks.

References

1 Emmanuel Abbe, Afonso S Bandeira, and Georgina Hall. Exact recovery in the stochastic block model. *arXiv preprint arXiv:1405.3267*, 2014.

2 Emmanuel Abbe and Colin Sandon. Community detection in general stochastic block models: fundamental limits and efficient recovery algorithms. *CoRR*, abs/1503.00609, 2015. `arXiv:1503.00609`.

3 Emmanuel Abbe and Colin Sandon. Detection in the stochastic block model with multiple clusters: proof of the achievability conjectures, acyclic bp, and the information-computation gap. *CoRR*, abs/1512.09080, 2015. `arXiv:1512.09080`.

4 J. Acharya, H. Das, A. Jafarpour, A. Orlitsky, and S. Pan. Competitive closeness testing. In *Conference on Learning Theory (COLT)*, 2011.

5 J. Acharya, H. Das, A. Jafarpour, A. Orlitsky, and S. Pan. Competitive classification and closeness testing. In *Conference on Learning Theory (COLT)*, 2012.

6 Anima Anandkumar, Dean P. Foster, Daniel J. Hsu, Sham Kakade, and Yi-Kai Liu. A spectral algorithm for latent dirichlet allocation. In Peter L. Bartlett, Fernando C. N. Pereira, Christopher J. C. Burges, Léon Bottou, and Kilian Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25: 26th Annual Conference on Neural Information Processing Systems 2012. Proceedings of a meeting held December 3-6, 2012, Lake Tahoe, Nevada, United States.*, pages 926–934, 2012. URL: `http://papers.nips.cc/paper/4637-a-spectral-algorithm-for-latent-dirichlet-allocation`.

7 Animashree Anandkumar, Rong Ge, Daniel Hsu, Sham M. Kakade, and Matus Telgarsky. Tensor decompositions for learning latent variable models. *Journal of Machine Learning Research*, 15:2773–2832, 2014. URL: `http://jmlr.org/papers/v15/anandkumar14b.html`.

8 Sanjeev Arora, Rong Ge, Ravindran Kannan, and Ankur Moitra. Computing a nonnegative matrix factorization–provably. In *Proceedings of the forty-fourth annual ACM symposium on Theory of computing*, pages 145–162. ACM, 2012.

9 Sanjeev Arora, Rong Ge, and Ankur Moitra. Learning topic models–going beyond svd. In *Foundations of Computer Science (FOCS), 2012 IEEE 53rd Annual Symposium on*, pages 1–10. IEEE, 2012.

10 Sanjeev Arora, Yuanzhi Li, Yingyu Liang, Tengyu Ma, and Andrej Risteski. Random walks on context spaces: Towards an explanation of the mysteries of semantic word embeddings. *CoRR*, abs/1502.03520, 2015. `arXiv:1502.03520`.

11 Sanjeev Arora, Yuanzhi Li, Yingyu Liang, Tengyu Ma, and Andrej Risteski. Random walks on context spaces: Towards an explanation of the mysteries of semantic word embeddings. *CoRR*, abs/1502.03520, 2015. `arXiv:1502.03520`.

12 T. Batu, L. Fortnow, R. Rubinfeld, W. D. Smith, and P. White. Testing closeness of discrete distributions. *Journal of the ACM (JACM)*, 60(1), 2013.

13 T. Batu, R. Kumar, and R. Rubinfeld. Sublinear algorithms for testing monotone and unimodal distributions. In *Symposium on Theory of Computing (STOC)*, pages 381–390, 2004.

14 Mikhail Belkin and Kaushik Sinha. Polynomial learning of distribution families. In *Foundations of Computer Science (FOCS), 2010 51st Annual IEEE Symposium on*, pages 103–112. IEEE, 2010.

15 Aditya Bhaskara, Moses Charikar, Ankur Moitra, and Aravindan Vijayaraghavan. Smoothed analysis of tensor decompositions. In *Proceedings of the 46th Annual ACM Symposium on Theory of Computing*, pages 594–603. ACM, 2014.

16 B. Bhattacharya and G. Valiant. Testing closeness with unequal sized samples. In *Neural Information Processing Systems (NIPS)*, 2015.

17 L. Birge. Estimating a density under order restrictions: Nonasymptotic minimax risk. *Annals of Statistics*, 15(3):995–1012, 1987.

18 David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022, 2003.

19 J. T. Chang. Full reconstruction of Markov models on evolutionary trees: Identifiability and consistency. *Mathematical Biosciences*, 137:51–73, 1996.

20 Peter Chin, Anup Rao, and Van Vu. Stochastic block model and community detection in the sparse graphs: A spectral algorithm with optimal rate of recovery. *CoRR*, abs/1501.05021, 2015. `arXiv:1501.05021`.

21 Sanjoy Dasgupta. Learning mixtures of gaussians. In *Foundations of Computer Science, 1999. 40th Annual Symposium on*, pages 634–644. IEEE, 1999.

22 Chris H. Q. Ding, Tao Li, and Wei Peng. Nonnegative matrix factorization and probabilistic latent semantic indexing: Equivalence chi-square statistic, and a hybrid method.

In *Proceedings, The Twenty-First National Conference on Artificial Intelligence and the Eighteenth Innovative Applications of Artificial Intelligence Conference, July 16-20, 2006, Boston, Massachusetts, USA*, pages 342–347. AAAI Press, 2006. URL: `http://www.aaai.org/Library/AAAI/2006/aaai06-055.php`.

23 Uriel Feige and Eran Ofek. Spectral techniques applied to sparse random graphs. *Random Structures & Algorithms*, 27(2):251–275, 2005.

24 Joel Friedman, Jeff Kahn, and Endre Szemeredi. On the second eigenvalue of random regular graphs. In *Proceedings of the twenty-first annual ACM symposium on Theory of computing*, pages 587–598. ACM, 1989.

25 Rong Ge, Qingqing Huang, and Sham M. Kakade. Learning mixtures of gaussians in high dimensions. In *Proceedings of the Symposium on Theory of Computing, STOC 2015,*, 2015.

26 O. Goldreich and D. Ron. On testing expansion in bounded-degree graphs. In *Technical Report TR00-020, Electronic Colloquium on Computational Complexity*, 2000.

27 S. Guha, A. McGregor, and S. Venkatasubramanian. Streaming and sublinear approximation of entropy and information distances. In *Proceedings of the ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2006.

28 Thomas Hofmann. Probabilistic latent semantic indexing. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 50–57. ACM, 1999.

29 Paul W Holland, Kathryn Blackmond Laskey, and Samuel Leinhardt. Stochastic blockmodels: First steps. *Social networks*, 5(2):109–137, 1983.

30 Daniel Hsu and Sham M Kakade. Learning mixtures of spherical gaussians: moment methods and spectral decompositions. In *Proceedings of the 4th conference on Innovations in Theoretical Computer Science*, pages 11–20. ACM, 2013.

31 Daniel Hsu, Sham M Kakade, and Tong Zhang. A spectral algorithm for learning hidden markov models. *Journal of Computer and System Sciences*, 78(5):1460–1480, 2012.

32 Adam Tauman Kalai, Ankur Moitra, and Gregory Valiant. Efficiently learning mixtures of two gaussians. In *Proceedings of the 42nd ACM symposium on Theory of computing*, pages 553–562. ACM, 2010.

33 Raghunandan H Keshavan, Sewoong Oh, and Andrea Montanari. Matrix completion from a few entries. In *Information Theory, 2009. ISIT 2009. IEEE International Symposium on*, pages 324–328. IEEE, 2009.

34 Florent Krzakala, Cristopher Moore, Elchanan Mossel, Joe Neeman, Allan Sly, Lenka Zdeborová, and Pan Zhang. Spectral redemption in clustering sparse networks. *Proceedings of the National Academy of Sciences*, 110(52):20935–20940, 2013.

35 Can Le, Elizaveta Levina, and Roman Vershynin. Sparse random graphs: regularization and concentration of the laplacian. *arXiv preprint arXiv:1502.03049*, 2015.

36 Can M. Le and Roman Vershynin. Concentration and regularization of random graphs. *CoRR*, abs/1506.00669, 2015. `arXiv:1506.00669`.

37 Omer Levy and Yoav Goldberg. Neural word embedding as implicit matrix factorization. In Zoubin Ghahramani, Max Welling, Corinna Cortes, Neil D. Lawrence, and Kilian Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pages 2177–2185, 2014. URL: `http://papers.nips.cc/paper/5477-neural-word-embedding-as-implicit-matrix-factorization`.

38 Laurent Massoulié. Community detection thresholds and the weak ramanujan property. In *Proceedings of the 46th Annual ACM Symposium on Theory of Computing*, pages 694–703. ACM, 2014.

39 Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781, 2013. `arXiv:1301.3781`.

40 Ankur Moitra and Gregory Valiant. Settling the polynomial learnability of mixtures of gaussians. In *Foundations of Computer Science (FOCS), 2010 51st Annual IEEE Symposium on*, pages 93–102. IEEE, 2010.

41 E. Mossel and S. Roch. Learning nonsingular phylogenies and hidden Markov models. *Annals of Applied Probability*, 16(2):583–614, 2006.

42 Elchanan Mossel, Joe Neeman, and Allan Sly. Stochastic block models and reconstruction. *arXiv preprint arXiv:1202.1499*, 2012. `arXiv:1202.1499`.

43 Elchanan Mossel, Joe Neeman, and Allan Sly. Consistency thresholds for binary symmetric block models. *CoRR*, abs/1407.1591, 2014. `arXiv:1407.1591`.

44 S. on Chan, I. Diakonikolas, G. Valiant, and P. Valiant. Optimal algorithms for testing closeness of discrete distributions. In *Proceedings of the ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1193–1203, 2014.

45 L. Paninski. Estimating entropy on m bins given fewer than m samples. *IEEE Transactions on Information Theory*, 50(9):2200–2203, 2004.

46 Christos H Papadimitriou, Hisao Tamaki, Prabhakar Raghavan, and Santosh Vempala. Latent semantic indexing: A probabilistic analysis. In *Proceedings of the seventeenth ACM SIGACT-SIGMOD-SIGART symposium on Principles of database systems*, pages 159–168. ACM, 1998.

47 S. Raskhodnikova, D. Ron, A. Shpilka, and A. Smith. Strong lower bounds for approximating distribution support size and the distinct elements problem. *SIAM Journal on Computing*, 39(3):813–842, 2009.

48 Karl Stratos, Michael Collins, and Daniel Hsu. Model-based word embeddings from decompositions of count matrices. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015, July 26-31, 2015, Beijing, China, Volume 1: Long Papers*, 2015.

49 Karl Stratos, Michael Collins Do-Kyum Kim, and Daniel Hsu. A spectral algorithm for learning class-based n-gram models of natural language. In *Proceedings of the 30th Conference on Uncertainty in Artificial Intelligence*, 2014.

50 G. Valiant and P. Valiant. Estimating the unseen: an $n/\log n$-sample estimator for entropy and support size, shown optimal via new clts. In *Symposium on Theory of Computing (STOC)*, 2011.

51 G. Valiant and P. Valiant. The power of linear estimators. In *Symposium on Foundations of Computer Science (FOCS)*, 2011.

52 G. Valiant and P. Valiant. Estimating the unseen: improved estimators for entropy and other properties. In *Neural Information Processing Systems (NIPS)*, 2013.

53 G. Valiant and P. Valiant. An automatic inequality prover and instance optimal identity testing. In *IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 51–60, 2014.

54 Santosh Vempala and Grant Wang. A spectral algorithm for learning mixture models. *Journal of Computer and System Sciences*, 68(4):841–860, 2004.

55 Anderson Y. Zhang and Harrison H. Zhou. Minimax rates of community detection in stochastic block models. *CoRR*, abs/1507.05313, 2015. URL: `http://arxiv.org/abs/1507.05313`, `arXiv:1507.05313`.

Learning Discrete Distributions from Untrusted Batches

Mingda Qiao[1] **and Gregory Valiant**[*2]

1 **Institute for Interdisciplinary Information Sciences, Tsinghua University, Beijing, China**
qmd14@mails.tsinghua.edu.cn

2 **Computer Science Department, Stanford University, Stanford, USA**
valiant@stanford.edu

Abstract

We consider the problem of learning a discrete distribution in the presence of an ϵ fraction of malicious data sources. Specifically, we consider the setting where there is some underlying distribution, p, and each data source provides a batch of $\geq k$ samples, with the guarantee that at least a $(1-\epsilon)$ fraction of the sources draw their samples from a distribution with total variation distance at most η from p. We make no assumptions on the data provided by the remaining ϵ fraction of sources – this data can even be chosen as an adversarial function of the $(1-\epsilon)$ fraction of "good" batches. We provide two algorithms: one with runtime exponential in the support size, n, but polynomial in k, $1/\epsilon$ and $1/\eta$ that takes $O((n+k)/\epsilon^2)$ batches and recovers p to error $O(\eta + \epsilon/\sqrt{k})$. This recovery accuracy is information theoretically optimal, to constant factors, even given an infinite number of data sources. Our second algorithm applies to the $\eta = 0$ setting and also achieves an $O(\epsilon/\sqrt{k})$ recover guarantee, though it runs in $\text{poly}((nk)^k)$ time. This second algorithm, which approximates a certain tensor via a rank-1 tensor minimizing ℓ_1 distance, is surprising in light of the hardness of many low-rank tensor approximation problems, and may be of independent interest.

1998 ACM Subject Classification Probability and Statistics

Keywords and phrases robust statistics, information-theoretic optimality

Digital Object Identifier 10.4230/LIPIcs.ITCS.2018.47

1 Introduction

Consider the following real-world problem: suppose there are millions of people texting away on their phones, and you wish to learn the distribution of words corresponding to a given mis-typed word, or the distribution of words that follow a given sequence, etc. The challenge of this setup is twofold. First, each person provides far too little data to accurately learn these distributions based solely on one person's data, hence a successful learning or estimation algorithm must combine data from different sources. Second, these sources are heterogeneous – some people have wider fingers than others, and the nature of typos likely differs between people. Further complicating this heterogeneity is the very real possibility that a small but not negligible fraction of the sources might be operated by adversarial agents whose goal is to embarrass the learning algorithm either as a form of corporate sabotage or as a publicity stunt. In much the same way as "Google bombing" or "link bombing" was used to associate specific websites with certain terms – perhaps the most famous of which

* Gregory's work is supported by ONR award N00014-17-1-2562 and NSF CAREER award CCF-1351108.

resulting in the George W. Bush biography being the top Google and Yahoo! hit when searching for the term "miserable failure" back in Sept. 2006 – it seems likely that certain groups of people would collectively attempt to influence the auto-correct or auto-suggest responses to certain misspellings.

This general problem of learning or estimation given data supplied by a large number of individuals has gained attention in the more practical communities under the keyword *Federated Learning* (see e.g. [24, 17] and Google research blog post [23]), and raises a number of pertinent questions: What notions of privacy can be maintained while leveraging everyone's data? Do we train a single model or try to personalize models for each user?

In this work, we consider a basic yet fundamental problem in this space: learning a discrete distribution given access to batches of samples, where an unknown $(1-\epsilon)$ fraction of batches are drawn i.i.d. from distributions that each has total variation (equivalently, ℓ_1) distance at most η from some target distribution, p, and we make no assumptions about the data contained in the remaining ϵ fraction of batches. The data in this "bad" ϵ fraction of the batches can even be chosen adversarially as a function of the "good" data. This problem is also a natural problem in the space of "robust statistics", and the recent line of recent work from the theory community on estimation and learning with untrusted data (e.g. [20, 7, 31, 6, 30, 8]), and we outline these connections in Section 1.2.

We begin by summarizing our main results and discuss the connections with related work. We conclude the introduction with a discussion of several relevant directions for future research.

1.1 Summary of Results

The following theorem characterizes our tight information theoretic result for robustly learning a discrete distribution in the setting where a certain fraction of batches of data are arbitrarily corrupted:

▶ **Theorem 1.** *Let p denote a distribution supported on n domain elements, and fix parameters $\epsilon \in (0, 1/900)$, $\eta > 0$, failure probability $\delta \in (0,1)$, and integer $k \geq 1$. Suppose we have access to $m = O((n + k + \log(1/\delta))/\epsilon^2)$ batches of data such that at least $m(1-\epsilon)$ of the batches consist of $\geq k$ i.i.d. draws from some distribution with ℓ_1 distance at most η from p. There exists an algorithm that runs in time $\mathrm{poly}\,(2^n, k, 1/\epsilon, 1/\eta, \log(1/\delta))$ and, with probability at least $1-\delta$, returns a distribution $\hat{p}$ satisfying $\|p-\hat{p}\|_1 = O(\eta+\epsilon/\sqrt{k})$, where the "$O$" notation hides an absolute constant.*

The recovery guarantees of the above theorem are information theoretically optimal, even given infinite data, as the following lower bound formalizes:

▶ **Theorem 2.** *In the setup of Theorem 1 for integers $k \geq 1$ and $n \geq 2$ and parameters $\epsilon \in (0, 1/2)$ and $\eta \in [0, 1/4)$, no algorithm can, with probability greater than $1/2$, return a distribution $\hat{p}$ satisfying $\|p - \hat{p}\|_1 < 2\eta + \epsilon/\sqrt{2k}$, even in the limit as the number of batches, m, tends to infinity.*

To provide some intuition for these results, consider the setting where $\eta = 0$, namely where the $(1-\epsilon)$ fraction of "good" batches all consist of k i.i.d. draws from p. In this case, the above results claim an optimal recovery error of $\Theta(\epsilon/\sqrt{k})$. Intuitively, this is due to the following "tensorization" property of the total variation distance, which we show in Appendix B: given two distributions p and q with $\|p-q\|_1 = \alpha = O(1/\sqrt{k})$, the ℓ_1 distance between their kth tensor products, $\|p^{\otimes k} - q^{\otimes k}\|_1$, is at least $\Theta(\alpha\sqrt{k})$. Here, the kth tensor product $p^{\otimes k}$ denotes the n^k sized object, indexed by a k-tuple $i_1, \ldots, i_k$, with

$p^{\otimes k}_{i_1,\ldots,i_k} = \prod_{j=1}^{k} p(i_j)$. This lower bound on the distance between the kth tensor products implies that, even though the adversary can alter the observed k-fold product distribution of p by ϵ (w.r.t. the ℓ_1 or total variation distance), any two k-tensors with distance $\leq \epsilon$ that each correspond to the kth tensor products of distributions will correspond to distributions whose distance is only $O(\epsilon/\sqrt{k})$, motivating our claimed recovery guarantees.

The algorithm to which the guarantees of Theorem 1 apply proceeds by reducing the problem at hand to $\leq 2^n$ instances of the problem of learning a Bernoulli random variable in this setting of untrusted batches. Specifically, for each of the $\leq 2^n$ possible subsets of the observed domain elements, the algorithm attempts to estimate the probability mass that p assigns to that set. The algorithm then combines these $\leq 2^n$ estimates to estimate p. In Section 1.3, we discuss the possibility of an efficient variant of this algorithm that proceeds via estimating these weights for only a polynomial number of subsets. We also note that the requirement that $\epsilon \leq 1/900$ can be relaxed slightly at the expense of readability of the proof.

Our proof of the lower bound (Theorem 2) proceeds by providing a construction of an explicit pair of indistinguishable instances, one corresponding to a distribution p, one corresponding to a distribution q, with $\|p-q\|_1 = 4\eta + 2\epsilon/\sqrt{2k}$. Each instance consists of two parts – a distribution from which the "good" batches of data are drawn, and a distribution over k-tuples of samples from which the ϵ fraction of "bad" batches are drawn. For the pair of instances constructed, the mixture of the good and bad batches corresponding to p is identical to the mixture of the good and bad batches corresponding to q.

In addition to the algorithm of Theorem 1, which runs in time exponential in the support size, we provide a second algorithm in the setting where $\eta = 0$, with runtime $(nk)^{O(k)}$ that also achieves the information theoretically optimal recovery error of $O(\epsilon/\sqrt{k})$.

▶ **Theorem 3.** *As in the setup of Theorem 1, given $m = (nk)^{O(k)} \log(1/\delta)/\epsilon^2$ batches of samples, of which a $(1-\epsilon)$ fraction consist of $\geq k$ i.i.d. draws from a distribution p, supported on $\leq n$ elements, there is an algorithm that runs in time $\mathrm{poly}((nk)^k, 1/\epsilon, \log(1/\delta))$ which, with probability at least $1-\delta$, returns a distribution $\hat{p}$ such that $\|p-\hat{p}\|_1 = O(\epsilon/\sqrt{k})$.*

The algorithm to which the above theorem applies proceeds by forming the k-tensor corresponding to the empirical distribution over the m batches, with each batch regarded as a k-tuple. The algorithm then efficiently computes a rank-1 k-tensor which approximates this empirical tensor in the (element-wise) ℓ_1 sense. The ability to efficiently (in time polynomial in the size of the k-tensor) compute this rank-1 approximation is rather surprising, and crucially leverages the structure of the empirical k-tensor guaranteed by the setup of our problem. Indeed the general problem of finding the best rank-1 approximation (in either an ℓ_1 or ℓ_2 sense) of a k-tensor is NP-hard for $k \geq 3$ [13]. Additionally, even for various "nice" random distributions of 3-tensors, efficient algorithms for related rank-1 approximation problems would yield efficient algorithms for recovering planted cliques of size $o(\sqrt{n})$ from $G(n, 1/2)$ [9].[1]

1.2 Related Work

There are a number of relevant lines of related work, including work from the theoretical computer science and information theory communities on learning, estimating and testing properties of distributions or collections of distributions, the classical work on "robust

[1] Specifically, [9] shows that given an $n \times n \times n$ 3-tensor A constructed from an instance of planted-clique, the ability to efficiently find a unit vector v that nearly maximizes $\max_{\|v\|=1} \sum_{i,j,k} v_i v_j v_k A_{i,j,k}$ would yield an algorithm for finding cliques of size at least $O(n^{1/3}\,\mathrm{polylog}(n))$ planted in $G(n, 1/2)$.

statistics", the recent line of work from the computer science community on robust learning and estimation with untrusted data and "adversarial" machine learning, and the very recent work from the more applied community on "federated learning". From a technical perspective, there are also connections between our work and the body of work on tensor factorizations, and low rank matrix and tensor approximations with respect to the ℓ_1 norm. We briefly summarize these main lines of related work.

1.2.1 Learning and Testing Discrete Distributions

The problem of learning a discrete distribution given access to independent samples has been intensely studied over the past century (see, e.g. [10, 18] and the references therein). For distributions supported on n elements, given access to k i.i.d. samples, minimax optimal loss rates as a function of n and k have been considered for various loss functions, including KL-divergence [4], ℓ_2 [15], χ^2 loss [15]. For ℓ_1 error (equivalently, "total variation distance" or "statistical distance"), it is easy to show that the expected worst-case error is $\Theta(\sqrt{n/k})$, and the recent work [15] establishes the exact first-order coefficients of this loss. Beyond this worst-case setting, there has also been significant work considering this learning problem with the goal of developing "instance optimal" algorithms that leverage whatever structure might be present in the given distribution (see e.g. [28, 35]).

In the context of work on testing distributional properties from the theoretical computer science community, the work closest to this current paper is the work of Levi, Ron, and Rubinfeld [21], which considers the following task: given access to independent draws from each of m distributions, distinguish whether all m distributions are identical (or very close), versus the case that there is significant variation between the distributions – namely that the average distance to any single distribution is at least a constant. We note that in this work, as in much of the distribution testing literature, the results are extremely sensitive to assumption that in the "yes" case, the distributions are all extremely close to one distribution. For example, the task of distinguishing whether the distributions have average distance at most α_1 from a single distribution, versus having average distance at least α_2, seems to be a significantly harder problem when α_1 is not asymptotically smaller than α_2.[2] In a related vein, the recent work [32] considers the setting of drawing k i.i.d. samples from each of m possibly heterogeneous distributions, and shows that the *set* of such distributions can be accurately learned. For example, if each distribution is a Bernoulli random variable with the ith distribution corresponding to some probability p_i, the histogram of the p_i's can be recovered to error $O(1/k)$, rather than the $\Theta(1/\sqrt{k})$ that would be given by the empirical estimates. Both [21] and [32], however, crucially rely on the assumption that the data consists of independent draws from the m distributions, and the algorithms and results do not extend to the present setting where some non-negligible fraction of the data is arbitrary/adversarial.

1.2.2 Robust Statistics, Learning with Unreliable Data, and Adversarial Learning

The problem of estimation and learning in the presence of contaminated or outlying data points has a history of study in the Statistics community, dating back to early work of Tukey [33] (see also the surveys [14, 12]). Recently, these problems have gained attention

[2] For example, testing whether two distributions supported on $\leq n$ elements are identical, vs have distance at least 0.1 requires $\Theta(n^{2/3})$ samples, whereas distinguishing whether the two distributions have distance ≤ 0.1 versus ≥ 0.9 requires $\Theta(n/\log n)$ samples. [2, 5, 34].

from both the applied and theoretical computer science communities. On the applied side, the interest in "adversarial" machine learning stems from the very real vulnerabilities of many deployed learning systems to "data poisoning" attacks in which an adversary selectively alters or plants a small amount of carefully chosen training data that significantly impacts the resulting trained model (see e.g. [27, 26, 36]).

On the theory side, recent work has revisited some of the classical robust statistics settings with an eye towards 1) establishing the information theoretic dependencies between the fraction of corrupted data, dimension of the problem, and achievable accuracy of returned model or estimate, and 2) developing computationally tractable algorithms that approach or achieve these information theoretic bounds. These recent works include [20, 7, 8] who consider the problem of robustly estimating the mean and covariance of high-dimensional Gaussians (and other distributions such as product distributions), and [31] who consider a sparse ratings aggregation setting. Other learning problems, such as robustly learning halfspaces [16], robust linear regression [3] and more general convex optimization [6] have also been considered in similar settings where some fraction of the data is drawn from a distribution of interest, and no assumptions are made about the remaining data. This latter work and [25] also considers several models for which strong positive results can be attained even when the majority of the data is arbitrary (i.e. $\epsilon > 1/2$).

The present setting, in which data arrives in batches, with some batches corresponding to "good" data represents a practically relevant instance of this more general robust learning problem, and we are not aware of previous work from the theory community that explicitly considers it.[3] The recent attention on "federated learning" [24, 17, 23] focuses on a similar setting where data is presented in batches (corresponding to individual users), although the current emphasis is largely on privacy and communication concerns, as opposed to robustness.

1.2.3 Low Rank Approximations in ℓ_1

The algorithm underlying Theorem 3 efficiently computes a rank-1 approximation of the empirical k-tensor of data, where the recovered rank-1 tensor is close in the (element-wise) ℓ_1 sense. Both the problems of computing the best low-rank ℓ_1 approximation of a *matrix*, and the problems of computing the best rank-1 approximation of a k-tensor for $k \geq 3$ are NP-hard [11, 13]. Nevertheless, recent work has provided efficient algorithms for returning good (in a competitive-analysis sense) low-rank approximations of matrices in the ℓ_1 norm [29], and for efficiently recovering low-rank approximations of special classes of tensors, including those with random or "incoherent" factors (see e.g. [19, 22]). Still, for many tensor decomposition problems over various families of random nearly low-rank tensors – such as those obtained from instances of planted-clique [9] – these decomposition problems remain mysterious.

1.3 Future directions and discussion

Two concrete open directions raised by this work are 1) understanding the computational hardness of this basic learning question, and 2) extending the information theoretic and algorithmic results to the setting where a *minority* of the batches of data are "good", which generalizes the problem of robustly learning a mixture of distributions. Before discussing these problems, we note that there are a number of other practically relevant related questions,

[3] One could view each batch of k samples as a single k-dimensional draw from a product of multinomial distributions, or as a k-sparse sample from an n-dimensional product distribution, although the results of previous work do not yield strong results for these settings.

including the extent to which this problem can be solved while maintaining some notion of privacy for each batch/user, and considering other basic learning and estimation tasks in this batched robust setting.

1.3.1 An Efficient Algorithm?

Recall that the algorithm of Theorem 1 has a linear sample complexity (treating ϵ as fixed), yet requires a runtime exponential in n. The algorithm of Theorem 3 requires more data, but runs in time $(nk)^{O(k)}$, and only applies to the clean but less practically relevant setting where $\eta = 0$. One natural question is whether there exists an algorithm that achieves both the linear sample complexity and the $(nk)^{O(k)}$ runtime. Alternately, does there exist an algorithm with data requirements and runtime that are polynomial in all the parameters, $n, k, 1/\epsilon$, and achieves the optimal recovery guarantees even in the $\eta = 0$ regime? Or are there natural barriers to such efficiency or connections to other problems that we believe to be computationally intractable?

One natural approach to yielding an efficient algorithm is via a variant of our first algorithm. At a high level, our algorithm proceeds by recovering $\epsilon/\sqrt{k}$-accurate estimates of the probability mass of each of the 2^n possible subsets of the n domain elements, and then recovering a distribution consistent with these estimates via a linear program. A natural approach to improving the running time of the algorithm is to find an efficient separation oracle for this linear program. Alternately, one could even imagine recovering $\epsilon/\sqrt{k}$-accurate estimates of the mass of a random $\text{poly}(k, n)$ sized set of subsets of the domain and then using these to recover the distribution. If the error in each estimate were independent, say distributed according to $N(0, \epsilon^2/k)$, then a polynomial (and even linear!) number of such measurements would suffice to recover the distribution to error $O(\epsilon/\sqrt{k})$. On the other hand, if an adversary is allowed to choose the errors in each measurement, trivially, an exponential number would be required. In our setting, however, the adversaries are somewhat limited in their ability to corrupt the measurements, and it is plausible that such an approach could work, perhaps both in practice and theory.

1.3.2 The small-α regime

Our positive results (Theorems 1 and 3) assume that the fraction of adversarial data, ϵ, is relatively small (at most of order 10^{-3}). While this assumption can be slightly relaxed by carefully adjusting the constants in the proofs, it is also natural to ask: what can we do if ϵ is much larger and even close to 1? This regime is examined in [6, 25]. It is clearly impossible to estimate the distribution in question to a nontrivial precision in the setting where $\epsilon \geq 1/2$ due to the symmetry between the real distribution and the one chosen by the adversary. Nevertheless, two learning frameworks were proposed for this case in [6]: the *list-decodable learning* framework (first introduced by Balcan et al. [1]), in which the learning algorithm is allowed to output multiple answers to the learning problem (perhaps $1/(1-\epsilon)$), and the *semi-verified* model, where a small amount of reliable/verified data is available. It seems plausible that variants of our algorithms that attempt to recover a rank $1/(1-\epsilon)$ approximation to the empirical tensor might be successful, though the algorithm and analysis are certainly not immediate.

1.4 Organization of Paper

In Section 2 we establish our information theoretic lower bound, Theorem 2, which bounds the accuracy of the recovered distribution and applies even in the infinite data setting. Section 3 describes our information theoretically optimal algorithm that has runtime exponential in the

support size, n, and establishes Theorem 1. Section 4 describes our more efficient algorithm, with runtime $(nk)^{O(k)}$, which directly approximates the k-tensor of observations via a rank-1 tensor, establishing Theorem 3. We conclude this section with a brief summary of the notation that will be used throughout the remainder of the paper.

1.5 Notation

Throughout, we use p to denote a discrete distribution of support size at most n. Without loss of generality, we assume that the support of p is $[n] = \{1, 2, \ldots, n\}$. Thus, p can also be interpreted as a probability vector $(p_1, p_2, \ldots, p_n) \in \mathbb{R}^n$, where p_i denotes the probability of element i. For a set $S \subseteq [n]$, we adopt the shorthand notation $p(S) = \sum_{i \in S} p_i$ to denote the total probability mass assigned to elements of S.

Let $p^{\otimes k}$ denote the k-fold product distribution corresponding to p, which is the kth tensor power of vector p. Each entry of $p^{\otimes k}$ is indexed by k indices $i_1, i_2, \ldots, i_k \in [n]$, and

$$p^{\otimes k}_{i_1, i_2, \ldots, i_k} = p_{i_1} p_{i_2} \cdots p_{i_k}.$$

More generally, a n^k-tensor is a k-dimensional array in which each entry is indexed by k elements in $[n]$. The *marginal* of n^k-tensor A is the vector $a \in \mathbb{R}^n$ defined as

$$a_i = \sum_{i_2, \ldots, i_k \in [n]} A_{i, i_2, \ldots, i_k}.$$

The i-th *slice* of A is the n^{k-1}-tensor obtained by restricting the first index of A to i.

A probability vector (resp. probability tensor) is a vector (resp. tensor) whose entries are nonnegative and sum to 1. Note that a probability n^k-vector defines a probability distribution on $[n]^k$. Moreover, its marginal is the marginal distribution of the first component, and its i-th slice (after normalization) is the conditional distribution of the other $k-1$ components given that the first component equals i.

The learning algorithm is given m batches of data, each of which is a k-tuple in $[n]^k$. Among the m batches, $m(1-\epsilon)$ are "good" in the sense that each of them is drawn from $\tilde{p}^{\otimes k}$ for some distribution $\tilde{p}$ with $\Delta(p, \tilde{p}) \leq \eta$.[4] The other $m\epsilon$ batches are chosen arbitrarily after the $m(1-\epsilon)$ good batches are drawn. The objective is to output a distribution q such that $\Delta(p, q)$ is small. Here $\Delta(p, q)$ stands for the total variation distance between p and q, i.e., one half of the ℓ_1-distance between vectors p and q:

$$\Delta(p, q) \coloneqq \frac{1}{2} \|p - q\|_1 = \frac{1}{2} \sum_{i=1}^{n} |p_i - q_i| = \max_{S \subseteq [n]} [p(S) - q(S)].$$

2 Information Theoretic Lower Bound

In this section we establish Theorem 2, showing that it is impossible to learn p to an $o(\eta + \epsilon/\sqrt{k})$ precision in ℓ_1 distance, even for distributions supported on 2 domain elements. This lower bound holds even if the learning algorithm is given unlimited computation power and access to infinitely many batches, i.e., the input of the algorithm is simply the mixture distribution $(1-\epsilon)P + \epsilon N$, where P is a mixture of k-fold product distributions of distributions that are η-close to p in the total variation distance, and N is a probability n^k-tensor that corresponds to the distribution of the adversarial batches.

[4] The distribution $\tilde{p}$ may vary for different good batches.

▶ **Lemma 4.** *For integer $k \geq 1$ and parameters $\epsilon \in (0, 1/2)$ and $\eta \in [0, 1/4)$, there are Bernoulli distributions p, q, p', q', together with two probability 2^k-tensors $N^{(p)}$ and $N^{(q)}$, such that:*

1. $\Delta(p, q) = 2\eta + \epsilon/\sqrt{2k}$.
2. $\Delta(p, p') = \Delta(q, q') = \eta$. *item* $(1-\epsilon)p'^{\otimes k} + \epsilon N^{(p)} = (1-\epsilon)q'^{\otimes k} + \epsilon N^{(q)}$.

Lemma 4 implies that given the distribution $(1-\epsilon)p'^{\otimes k} + \epsilon N^{(p)} = (1-\epsilon)q'^{\otimes k} + \epsilon N^{(q)}$ as input, the algorithm cannot determine whether the underlying distribution is p or q. This establishes the $\Omega(\eta + \epsilon/\sqrt{k})$ lower bound in Theorem 2.

Proof of Lemma 4. Since $\epsilon/\sqrt{2k} < \epsilon < 1/2$, by Lemma 11, for Bernoulli distributions p' and q' with means $(1 - \epsilon/\sqrt{2k})/2$ and $(1 + \epsilon/\sqrt{2k})/2$, it holds that $\Delta(p', q') = \epsilon/\sqrt{2k}$ and $\Delta(p'^{\otimes k}, q'^{\otimes k}) \leq \epsilon$. Let p and q be Bernoulli distributions with means $(1 - \epsilon/\sqrt{2k})/2 - \eta$ and $(1 + \epsilon/\sqrt{2k})/2 + \eta$.[5] Then the distributions clearly satisfy the first two conditions.

Let A be the entrywise maximum of $p'^{\otimes k}$ and $q'^{\otimes k}$, i.e., $A_i = \max(p'^{\otimes k}_i, q'^{\otimes k}_i)$ for every $i \in [2]^k$. Let α denote the sum of entries in A. Then, $\alpha = 1 + \Delta(p'^{\otimes k}, q'^{\otimes k}) \leq 1 + \epsilon \leq 1/(1-\epsilon)$. Define $N^{(p)} = \left[A/\alpha - (1-\epsilon)p'^{\otimes k}\right]/\epsilon$ and $N^{(q)} = \left[A/\alpha - (1-\epsilon)q'^{\otimes k}\right]/\epsilon$. Then the third condition is met, and it remains to prove that $N^{(p)}$ and $N^{(q)}$ are probability tensors.

Note that the elements in $N^{(p)}$ sum to $(\alpha/\alpha - (1-\epsilon)\cdot 1)/\epsilon = 1$. Moreover, since $\alpha \leq 1/(1-\epsilon)$, $N^{(p)}_i = A_i/\alpha - (1-\epsilon)p'^{\otimes k}_i \geq (1-\epsilon)(A_i - p'^{\otimes k}_i) \geq 0$ for any $i \in [2]^k$. This shows that $N^{(p)}$ and $N^{(q)}$ are probability tensors and finishes the proof. ◀

3 An Information Theoretically Optimal Algorithm

In this section, we present an algorithm that approximates the distribution to an information theoretically optimal $O(\eta + \epsilon/\sqrt{k})$ accuracy using $O((n + k + \ln(1/\delta))/\epsilon^2)$ batches. The algorithm runs in time $\mathrm{poly}(2^n, k, 1/\epsilon, 1/\eta, \ln(1/\delta))$. In particular, the algorithm is computationally efficient if the distribution has a relatively small support.

Our approach is to reduce the problem to learning Bernoulli distributions: we estimate the probability mass of any subset of the support to an $O(\eta + \epsilon/\sqrt{k})$ accuracy, and output a distribution that is consistent with the measurements. Then the total variation distance between our output and the true distribution would also be upper bounded by $O(\eta + \epsilon/\sqrt{k})$.

In the following we define a subroutine that efficiently estimates $p(S)$, the probability mass that p assigns to set $S \subseteq [n]$. Given batches $x_1, x_2, \ldots, x_m \in [n]^k$, the algorithm counts the number of batches that contain exactly i elements in S $(0 \leq i \leq k)$ and obtains a distribution f^S over $\{0, 1, \ldots, k\}$. Then it outputs $(i+2)\eta$ for some $0 \leq i \leq 1/\eta - 4$ as the estimation, if there is a mixture of binomial distributions with success probabilities in interval $[i\eta, (i+4)\eta]$ such that its ℓ_1 distance to f^S is upper bounded by $O(\epsilon)$.

Note that the mathematical program (8) can be transformed to an equivalent linear program, and therefore can be efficiently solved.

The following lemma bounds the difference between the estimation $\mathsf{BinomialEst}((x_i)_{i\in[m]}, S, \epsilon, \eta)$ and $p(S)$.

▶ **Lemma 5.** *Suppose that $S \subseteq [n]$, $\epsilon \in (0, 1/900)$, and $\eta, \delta \in (0, 1)$. For some $m = O((k + \ln(1/\delta))/\epsilon^2)$, $x_1, x_2, \ldots, x_m$ are m batches chosen as in the setup of Theorem 1. With probability $1 - \delta$ over the randomness of the $m(1-\epsilon)$ good batches,*

$$\left|\mathsf{BinomialEst}((x_i)_{i\in[m]}, S, \epsilon, \eta) - p(S)\right| \leq 3\eta + 60\epsilon/\sqrt{k}.$$

[5] p and q are well defined since $\epsilon/(2\sqrt{2k}) + \eta < 1/4 + 1/4 = 1/2$.

Algorithm 1: BinomialEst$((x_i)_{i\in[m]}, S, \epsilon, \eta)$

Input: Batches $x_1, x_2, \ldots, x_m \in [n]^k$, set $S \subseteq [n]$, and parameters ϵ, η.
Output: An estimation of $p(S)$.

```
for i = 1, 2, ..., m do
    cnt_i ← Σ_{j=1}^{k} 𝕀{x_{i,j} ∈ S};
for i = 0, 1, ..., k do
    f_i^S ← (1/m) Σ_{j=1}^{m} 𝕀{cnt_j = i};
tot ← 4η/(ϵ/k);
for i = 0, 1, ..., 1/η − 4 do
    return (i + 2)η if the following mathematical program is feasible:
```

$$\begin{aligned} \text{find} \quad & \alpha_0, \alpha_1, \ldots, \alpha_{\text{tot}} \\ \text{subject to} \quad & \Delta\left(\sum_{j=0}^{\text{tot}} \alpha_j B\left(k, i\eta + j\epsilon/k\right), f^S\right) \le 2\epsilon \\ & \sum_{j=0}^{\text{tot}} \alpha_j = 1 \\ & \alpha_j \ge 0, \ \forall j \in \{0, 1, \ldots, \text{tot}\} \end{aligned} \tag{1}$$

Here the "O" notation hides an absolute constant.

Proof of Lemma 5. Let $I \subseteq [m]$ denote the indices of the $m(1-\epsilon)$ good batches and let $\overline{I}$ be its complement. Define distributions $\hat{p}^S$ and δ^S over $\{0, 1, \ldots, k\}$ as: $\hat{p}_i^S = \frac{1}{|I|}\sum_{j\in I}\mathbb{I}\{\text{cnt}_j = i\}$ and $\delta_i^S = \frac{1}{|\overline{I}|}\sum_{j\in\overline{I}}\mathbb{I}\{\text{cnt}_j = i\}$. Then the distribution f^S defined in Algorithm 1 can be rewritten as $f^S = (1-\epsilon)\hat{p}^S + \epsilon\delta^S$.

Concentration of $\hat{p}^S$. For each good batch $j \in I$, let θ_j denote the probability of set S in the actual distribution from which batch j is drawn. By the assumption that the underlying distribution of each good batch is η-close to the target p, we have $|\theta_j - p(S)| \le \eta$. Define p^S as the uniform mixture of binomial distributions $B(k, \theta_j)$ for $j \in I$, i.e., $p^S = \frac{1}{|I|}\sum_{j\in I} B(k, \theta_j)$. Since $\hat{p}^S$ denotes the frequency of $|I| = \Omega(m)$ samples, exactly one of which is drawn from $B(k, \theta_j)$ for each $j \in I$, for some $m = O((k + \ln(1/\delta))/\epsilon^2)$, it holds with probability $1-\delta$ that $\Delta(p^S, \hat{p}^S) \le \epsilon/2$. It follows that

$$\Delta(p^S, f^S) \le \Delta(p^S, \hat{p}^S) + \Delta(\hat{p}^S, f^S) = \Delta(p^S, \hat{p}^S) + \epsilon\Delta(\hat{p}^S, \delta^S) \le 3\epsilon/2.$$

Feasibility of mathematical program (8). Let $i^* = \lfloor p(S)/\eta \rfloor - 2$. We show that with probability $1-\delta$, the mathematical program (8) is feasible for $i = i^*$.
Since $|(i^*+2)\eta - p(S)| \le \eta$ and $|\theta_j - p(S)| \le \eta$, it holds for any $j \in I$ that $i^*\eta \le \theta_j \le (i^*+4)\eta$. Let $\tilde{\theta}_j$ be the value among $\{i^*\eta + t\epsilon/k : t \in \{0, 1, \ldots, \text{tot}\}\}$ that is closest to θ_j. By definition, we have $|\tilde{\theta}_j - \theta_j| \le \epsilon/(2k)$, which implies that

$$\Delta(B(k, \theta_j), B(k, \tilde{\theta}_j)) \le k \cdot \epsilon/(2k) = \epsilon/2.$$

Let $\tilde{p}^S = \frac{1}{|I|}\sum_{j\in I} B(k,\tilde{\theta}_j)$. Then we have

$$\Delta(p^S,\tilde{p}^S) \le \frac{1}{|I|}\sum_{j\in I}\Delta(B(k,\theta_j),B(k,\tilde{\theta}_j)) \le \epsilon/2,$$

and it follows that with probability $1-\delta$,

$$\Delta(\tilde{p}^S,f^S) \le \Delta(p^S,\tilde{p}^S)+\Delta(p^S,f^S) \le \epsilon/2+3\epsilon/2 = 2\epsilon.$$

Note that this naturally defines a feasible solution to the mathematical program (8) for $i=i^*$: for each $t\in\{0,1,\dots,\mathrm{tot}\}$, $\alpha_t = \frac{1}{|I|}\sum_{j\in I}\mathbb{I}\left\{\tilde{\theta}_j = i^*\eta + t\epsilon/k\right\}$.

Approximation guarantee. Let x_0 denote $\mathsf{BinomialEst}((x_i)_{i\in[m]},S,\epsilon,\eta)$. In the following we prove that, with probability $1-\delta$, x_0 is a good approximation of $p(S)$.

Recall that $p^S = \frac{1}{|I|}\sum_{j\in I}B(k,\theta_j)$ is a mixture of binomial distributions with success probabilities in $[p(S)-\eta,p(S)+\eta]$. Moreover, by definition of procedure $\mathsf{BinomialEst}$, f^S is 2ϵ-close to a mixture of binomial distributions with success probabilities that lie in $[x_0-2\eta,x_0+2\eta]$. Then the inequality $\Delta(p^S,f^S)\le 3\epsilon/2$, which holds with probability $1-\delta$, further implies that the two mixtures above are 4ϵ-close to each other.

Let $\epsilon' = 60\epsilon/\sqrt{k}$. Note that our assumption $\epsilon < 1/900$ implies that $\epsilon' < 1/(15\sqrt{k})$. Now suppose for a contradiction that $|x_0-p(S)| > 3\eta+\epsilon'$. Without loss of generality, we have $x_0 > p(S)+3\eta+\epsilon'$, or equivalently, $x_0-2\eta > p(S)+\eta+\epsilon'$. Applying the contrapositive of Lemma 13 with parameters $\epsilon=\epsilon'$, $p=p(S)+\eta$ and $q=x_0-2\eta$ gives a contradiction. ◀

Now we prove our main theorem by Lemma 5 and a reduction to the estimation of Bernoulli random variables.

Proof of Theorem 1. We compute $\hat{p}(S) = \mathsf{BinomialEst}((x_i)_{i\in[m]},S)$ for each $S\subseteq[n]$, and then output an arbitrary feasible solution (if any) to the following linear program:

$$\begin{aligned} \text{find}\quad & q\in\mathbb{R}^n \\ \text{subject to}\quad & \left|\sum_{i\in S} q_i - \hat{p}(S)\right| \le 3\eta+60\epsilon/\sqrt{k},\ \forall S\subseteq[n] \\ & \sum_{i=1}^n q_i = 1 \\ & q_i \ge 0,\ \forall i\in[n] \end{aligned} \tag{2}$$

The algorithm described above involves solving a linear program with n variables and $O(2^n)$ constraints, as well as 2^n calls to the subroutine $\mathsf{BinomialEst}$, each of which takes polynomial time. Thus the whole algorithm runs in $\mathrm{poly}(2^n,k,1/\epsilon,1/\eta,\log(1/\delta))$ time.

Let $\delta_0 = \delta/2^n$. By Lemma 5 and a union bound, for some $m = O\left((k+\ln(1/\delta_0))/\epsilon^2\right) = O\left((n+k+\ln(1/\delta))/\epsilon^2\right)$, it holds with probability $1-2^n\cdot\delta_0 = 1-\delta$ that, for any $S\subseteq[n]$, $|\hat{p}(S)-p(S)| \le 3\eta+60\epsilon/\sqrt{k}$. This implies that linear program (2) is feasible with probability $1-\delta$. Moreover, let q be any feasible solution to (2). Then it holds for any $S\subseteq[n]$ that

$$|p(S)-q(S)| \le |p(S)-\hat{p}(S)|+|q(S)-\hat{p}(S)| \le 6\eta+120\epsilon/\sqrt{k}.$$

And hence, from the definition of total variation distance, $\Delta(p,q) = O(\eta+\epsilon/\sqrt{k})$. ◀

4 A Tensor-Based Algorithm

In this section, we present a different algorithm that efficiently learns the distribution in the small-batch regime where k is a constant, and where all the good batches are drawn from the actual distribution p, i.e., $\eta = 0$. The algorithm works on the *frequency tensor* A defined by the m batches, i.e., $A_{i_1,i_2,\ldots,i_k}$ is the fraction of batches whose set of samples equal $(i_1, i_2, \ldots, i_k)$.

The following recursive function $\mathsf{DistSet}(n, k, A)$ takes tensor A and outputs a set of "guesses" of the distribution.

Algorithm 2: $\mathsf{DistSet}(n, k, A)$

Input: n, k, and n^k-tensor A.
Output: A set of n-dimensional vectors.

```
if k = 1 then
    return {A};
for i = 1, 2, ..., n do
    A_i ← the normalized i-th slice of A;
    S_i ← DistSet(n, k − 1, A_i);
a ← the marginal of A;
return S_1 ∪ S_2 ∪ ··· ∪ S_n ∪ {a};
```

The following lemma states that given sufficiently many batches, $\mathsf{DistSet}(n, k, A)$ contains a vector that is $O(\epsilon/k)$-close to the real distribution p in the total variation distance.

▶ **Lemma 6.** *Suppose that $\epsilon \in (0, 1/2)$ and vector p is a probability distribution on $[n]$. Let $A \in \mathbb{R}^{n^k}$ be the frequency tensor of m batches, among which $m(1-\epsilon)$ are drawn from $p^{\otimes k}$ and the other $m\epsilon$ batches are arbitrary and may depend on the $m(1-\epsilon)$ good batches. Then, with probability $1-\delta$ (over the randomness in the good batches),*

$$\min_{q \in \mathsf{DistSet}(n,k,A)} \Delta(p, q) \le \frac{6\epsilon}{k} + O\left(\sqrt{\frac{n^k \cdot k! \cdot (n + k \ln n + \ln(1/\delta))}{m}}\right).$$

Proof of Lemma 6. Let $\delta_0 = \delta/(n+1)^{k-1}$. We recursively define functions $f_{n,1}(\epsilon, m)$ through $f_{n,k}(\epsilon, m)$ for $\epsilon, m > 0$ as follows:

$$f_{n,1}(\epsilon, m) = 3\epsilon + C\sqrt{\frac{n + \ln(1/\delta_0)}{m}} \tag{3}$$

and for $t \ge 2$,

$$f_{n,t}(\epsilon, m) = \max\left(\frac{3\epsilon}{t} + C\sqrt{\frac{n + \ln(1/\delta_0)}{m}}, f_{n,t-1}\left(1 - (1-\epsilon)^{(t-1)/t}, \frac{m}{nt}\right)\right). \tag{4}$$

Here C is an absolute constant to be determined later.

The following claim states that $\min_{q \in \mathsf{DistSet}(n,k,A)} \Delta(p, q)$ is upper bounded by $f_{n,k}(\epsilon, m(1-\epsilon))$ with high probability.

▶ **Claim 1.** *Suppose that $t \in [k]$, $\epsilon \in (0, 1/2)$, and $A \in \mathbb{R}^{n^t}$ is the frequency tensor of a data set, in which at least m batches are drawn from $p^{\otimes t}$, and the fraction of these batches is at least $1-\epsilon$. Then, with probability $1 - (n+1)^{t-1} \cdot \delta_0$, it holds that*

$$\min_{q \in \mathsf{DistSet}(n,t,A)} \Delta(p, q) \le f_{n,t}(\epsilon, m).$$

The following claim further upper bounds $f_{n,k}(\epsilon, m)$.

▶ **Claim 2.** *For any $\epsilon \in (0, 1/2)$ and $m > 0$,*

$$f_{n,k}(\epsilon, m) \le \frac{6\epsilon}{k} + C\sqrt{\frac{n^k \cdot k! \cdot (n + \ln(1/\delta_0))}{m}}.$$

Claims 1 and 2 together imply that with probability $1 - (n+1)^{k-1} \cdot \delta_0 = 1 - \delta$, it holds that

$$\begin{aligned}\min_{q \in \mathsf{DistSet}(n,k,A)} \Delta(p, q) &\le f_{n,k}(\epsilon, m(1-\epsilon)) \\ &\le \frac{6\epsilon}{k} + O\left(\sqrt{\frac{n^k \cdot k! \cdot (n + k\ln n + \ln(1/\delta))}{m}}\right).\end{aligned}$$

◀

Lemma 6 implies that given

$$m = O(n^k \cdot k! \cdot (n + k\ln n + \ln(1/\delta)) \cdot k^2/\epsilon^2) = (nk)^{O(k)} \ln(1/\delta)/\epsilon^2$$

batches, the minimum distance is upper bounded by $O(\epsilon/k)$. Now we prove Theorem 3 by leveraging this approximation guarantee of procedure DistSet.

Proof of Theorem 3. Let $q_0 = \operatorname{argmin}_{q \in \mathsf{DistSet}(n,k,A)} \Delta(A, q^{\otimes k})$ and $q_1 = \operatorname{argmin}_{q \in \mathsf{DistSet}(n,k,A)} \Delta(p, q)$. Note that q_0 can be computed from A in $(nk)^{O(k)}/\epsilon^2$ time. In the following we prove that q_0 is a good approximation of p. By Lemma 6, for some $m = (nk)^{O(k)} \ln(1/\delta)/\epsilon^2$, it holds with probability $1 - \delta/2$ that $\Delta(p, q_1) \le 7\epsilon/k$, and thus, $\Delta(p^{\otimes k}, {q_1}^{\otimes k}) \le k\Delta(p, q_1) \le 7\epsilon$. We write A as $A = (1-\epsilon)\hat{P} + \epsilon N$, where $\hat{P}$ and N are the frequency tensor of the "good" and "bad" batches, respectively. Since $\hat{P}$ denotes the frequency among $m(1-\epsilon) = \Omega(m)$ samples drawn from $p^{\otimes k}$, and the support of $p^{\otimes k}$ is of size n^k, for some $m = O\left((n^k + \ln(1/\delta))/\epsilon^2\right)$, it holds with probability $1 - \delta/2$ that $\Delta(\hat{P}, p^{\otimes k}) \le \epsilon$. Therefore,

$$\Delta(A, p^{\otimes k}) \le \Delta(A, \hat{P}) + \Delta(\hat{P}, p^{\otimes k}) = \epsilon\Delta(\hat{P}, N) + \Delta(\hat{P}, p^{\otimes k}) \le 2\epsilon.$$

By a union bound, it holds with probability $1 - \delta$ that

$$\Delta(A, {q_1}^{\otimes k}) \le \Delta(A, p^{\otimes k}) + \Delta(p^{\otimes k}, {q_1}^{\otimes k}) \le 2\epsilon + 7\epsilon = 9\epsilon.$$

Furthermore, by definition of q_0, $\Delta(A, {q_0}^{\otimes k}) \le \Delta(A, {q_1}^{\otimes k}) \le 9\epsilon$, and thus

$$\Delta(p^{\otimes k}, {q_0}^{\otimes k}) \le \Delta(A, p^{\otimes k}) + \Delta(A, {q_0}^{\otimes k}) \le 2\epsilon + 9\epsilon = 11\epsilon.$$

Let $\epsilon' = 165\epsilon/\sqrt{k} < 1/(15\sqrt{k})$. Since $\Delta(p^{\otimes k}, {q_0}^{\otimes k}) \le 11\epsilon = \epsilon'\sqrt{k}/15$, applying the contrapositive of Lemma 12 with parameter ϵ' yields that, with probability at least $1 - \delta$, $\Delta(p, q_0) \le \epsilon' = O(\epsilon/\sqrt{k})$. ◀

References

1 Maria-Florina Balcan, Avrim Blum, and Santosh Vempala. A discriminative framework for clustering via similarity functions. In *Symposium on Theory of Computing (STOC)*, pages 671–680, 2008.

2 Tugkan Batu, Lance Fortnow, Ronitt Rubinfeld, Warren D Smith, and Patrick White. Testing that distributions are close. In *Foundations of Computer Science, 2000. Proceedings. 41st Annual Symposium on*, pages 259–269. IEEE, 2000.

3 K. Bhatia, P. Jain, and P. Kar. Robust regression via hard thresholding. In *Advances in Neural Information Processing Systems (NIPS)*, pages 721–729, 2015.

4 Dietrich Braess and Thomas Sauer. Bernstein polynomials and learning theory. *Journal of Approximation Theory*, 128(2):187–206, 2004.

5 Siu-On Chan, Ilias Diakonikolas, Gregory Valiant, and Paul Valiant. Optimal algorithms for testing closeness of discrete distributions. In *Proceedings of the twenty-fifth annual ACM-SIAM symposium on Discrete algorithms*, pages 1193–1203. Society for Industrial and Applied Mathematics, 2014.

6 Moses Charikar, Jacob Steinhardt, and Gregory Valiant. Learning from untrusted data. In *Symposium on Theory of Computing (STOC)*, pages 47–60, 2017.

7 Ilias Diakonikolas, Gautam Kamath, Daniel M Kane, Jerry Li, Ankur Moitra, and Alistair Stewart. Robust estimators in high dimensions without the computational intractability. In *Foundations of Computer Science (FOCS)*, pages 655–664, 2016.

8 Ilias Diakonikolas, Gautam Kamath, Daniel M Kane, Jerry Li, Ankur Moitra, and Alistair Stewart. Being robust (in high dimensions) can be practical. *arXiv preprint arXiv:1703.00893*, 2017.

9 Alan Frieze and Ravi Kannan. A new approach to the planted clique problem. In *LIPIcs-Leibniz International Proceedings in Informatics*, volume 2. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2008.

10 E Gilbert. Codes based on inaccurate source probabilities. *IEEE Transactions on Information Theory*, 17(3):304–314, 1971.

11 Nicolas Gillis and Stephen A Vavasis. On the complexity of robust pca and l1-norm low-rank matrix approximation. *arXiv preprint arXiv:1509.09236*, 2015.

12 Frank R Hampel, Elvezio M Ronchetti, Peter J Rousseeuw, and Werner A Stahel. *Robust statistics: the approach based on influence functions*, volume 114. John Wiley & Sons, 2011.

13 Christopher J Hillar and Lek-Heng Lim. Most tensor problems are np-hard. *Journal of the ACM (JACM)*, 60(6):45, 2013.

14 Peter J Huber and Elvezio M Ronchetti. *Robust statistics*, volume 2. John Wiley & Sons, 2009. doi:10.1002/9780470434697.

15 Sudeep Kamath, Alon Orlitsky, Dheeraj Pichapati, and Ananda Theertha Suresh. On learning distributions from their samples. In *Conference on Learning Theory (COLT)*, pages 1066–1100, 2015.

16 Adam R Klivans, Philip M Long, and Rocco A Servedio. Learning halfspaces with malicious noise. *Journal of Machine Learning Research*, 10(Dec):2715–2740, 2009.

17 Jakub Konečný, H. Brendan McMahan, Felix X. Yu, Peter Richtarik, Ananda Theertha Suresh, and Dave Bacon. Federated learning: Strategies for improving communication efficiency. In *NIPS Workshop on Private Multi-Party Machine Learning*, 2016. URL: https://arxiv.org/abs/1610.05492.

18 Raphail Krichevsky and Victor Trofimov. The performance of universal encoding. *IEEE Transactions on Information Theory*, 27(2):199–207, 1981.

19 Joseph B Kruskal. Three-way arrays: rank and uniqueness of trilinear decompositions, with application to arithmetic complexity and statistics. *Linear algebra and its applications*, 18(2):95–138, 1977.

20 Kevin A Lai, Anup B Rao, and Santosh Vempala. Agnostic estimation of mean and covariance. In *Foundations of Computer Science (FOCS)*, pages 665–674, 2016.

21 Reut Levi, Dana Ron, and Ronitt Rubinfeld. Testing properties of collections of distributions. *Theory of Computing*, 9(8):295–347, 2013.

22 Tengyu Ma, Jonathan Shi, and David Steurer. Polynomial-time tensor decompositions with sum-of-squares. In *Foundations of Computer Science (FOCS), 2016 IEEE 57th Annual Symposium on*, pages 438–446. IEEE, 2016.

23 Brendan McMahan and Daniel Ramage. `https://research.google.com/pubs/pub44822.html`, 2017.

24 H Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, et al. Communication-efficient learning of deep networks from decentralized data. *arXiv preprint arXiv:1602.05629*, 2016.

25 Michela Meister and Gregory Valiant. A data prism: Semi-verified learning in the small-alpha regime. *arXiv preprint arXiv:1708.02740*, 2017.

26 Blaine Nelson, Battista Biggio, and Pavel Laskov. Understanding the risk factors of learning in adversarial environments. In *Proceedings of the 4th ACM workshop on Security and artificial intelligence*, pages 87–92. ACM, 2011.

27 James Newsome, Brad Karp, and Dawn Song. Paragraph: Thwarting signature learning by training maliciously. In *International Workshop on Recent Advances in Intrusion Detection*, pages 81–105. Springer, 2006.

28 Alon Orlitsky and Ananda Theertha Suresh. Competitive distribution estimation: Why is good-turing good. In *Advances in Neural Information Processing Systems*, pages 2143–2151, 2015.

29 Zhao Song, David P Woodruff, and Peilin Zhong. Low rank approximation with entrywise l1-norm error. *Algorithms*, 1:2, 2017.

30 Jacob Steinhardt, Moses Charikar, and Gregory Valiant. Resilience: A criterion for learning in the presence of arbitrary outliers. *arXiv preprint arXiv:1703.04940*, 2017.

31 Jacob Steinhardt, Gregory Valiant, and Moses Charikar. Avoiding imposters and delinquents: Adversarial crowdsourcing and peer prediction. In *Advances in Neural Information Processing Systems (NIPS)*, pages 4439–4447, 2016.

32 Kevin Tian, Weihao Kong, and Gregory Valiant. Learning populations of parameters. In *Neural Information Processing Systems (to appear)*, 2017.

33 John W Tukey. A survey of sampling from contaminated distributions. *Contributions to probability and statistics*, 2:448–485, 1960.

34 Gregory Valiant and Paul Valiant. Estimating the unseen: an n/log (n)-sample estimator for entropy and support size, shown optimal via new clts. In *Proceedings of the forty-third annual ACM symposium on Theory of computing*, pages 685–694. ACM, 2011.

35 Gregory Valiant and Paul Valiant. Instance optimal learning of discrete distributions. In *Proceedings of the forty-eighth annual ACM symposium on Theory of Computing*, pages 142–155. ACM, 2016.

36 Huang Xiao, Battista Biggio, Blaine Nelson, Han Xiao, Claudia Eckert, and Fabio Roli. Support vector machines under adversarial label contamination. *Neurocomputing*, 160:53–62, 2015.

A Technical Lemmas

We prove a few useful technical lemmas.

▶ **Lemma 7.** *For any $\epsilon \in [0, 1/2]$ and $\alpha \in [0, 1]$, $(1-\epsilon)^\alpha \geq 1 - 2\alpha\epsilon$.*

Proof of Lemma 7. Let $f(x) = (1-x)^\alpha - (1-2\alpha x)$. Since $f'(x) = \alpha[2 - (1-x)^{\alpha-1}] \geq 0$ for any $x \in [0, 1/2]$ and $\alpha \in [0, 1]$, it holds for any $\epsilon \in [0, 1/2]$ that $f(\epsilon) \geq f(0) = 0$, which proves the lemma. ◀

▶ **Lemma 8.** *For any $a > 0$, the function $(1 - a/x)^x$ is increasing on $(a, +\infty)$.*

Proof of Lemma 8. Let $f(x) = x \ln(1 - a/x)$. Then for $x \in (a, +\infty)$, we have $f'(x) = \ln\left(1 - \frac{a}{x}\right) + \frac{a}{x-a}$ and $f''(x) = -\frac{a^2}{x(x-a)^2} < 0$. Since $\lim_{x\to+\infty} f'(x) = 0$, f' is positive on $(a, +\infty)$, which proves the lemma. ◀

▶ **Lemma 9.** *Suppose that n and m are positive integers such that $m \geq \max(n, 2)$. Then for any $\alpha \in [0, 1.1\sqrt{n}]$, $\left(1+\frac{\alpha}{n}\right)^n \left(1-\frac{\alpha}{m}\right)^m \geq \frac{1}{7}$.*

Proof of Lemma 9. For $x \in [0, 1.1\sqrt{n}]$, define

$$f_{n,m}(x) = n \ln\left(1+\frac{x}{n}\right) + m \ln\left(1-\frac{x}{m}\right).$$

Note that $f_{n,m}$ is well-defined since $x \leq 1.1\sqrt{n} \leq 1.1\sqrt{m} < m$. Moreover,

$$f'_{n,m}(x) = 1/(1+x/n) - 1/(1-x/m) \leq 0.$$

Thus, it remains to prove the inequality for $\alpha = 1.1\sqrt{n}$, i.e.,

$$\left(1+\frac{1.1}{\sqrt{n}}\right)^n \left(1-\frac{1.1\sqrt{n}}{m}\right)^m \geq \frac{1}{7}. \tag{5}$$

If $n \geq 2$, we lower bound the lefthand side of (5) by

$$\begin{aligned}\left(1+\frac{1.1}{\sqrt{n}}\right)^n \left(1-\frac{1.1\sqrt{n}}{m}\right)^m &\geq \left(1+\frac{1.1}{\sqrt{n}}\right)^n \left(1-\frac{1.1\sqrt{n}}{n}\right)^n \\ &= \left(1-\frac{1.21}{n}\right)^n \\ &\geq \left(1-\frac{1.21}{2}\right)^2 \\ &\geq \frac{1}{7}.\end{aligned}$$

Here the first and third steps follow from Lemma 8. For $n = 1$, we have

$$\left(1+\frac{1.1}{\sqrt{n}}\right)^n \left(1-\frac{1.1\sqrt{n}}{m}\right)^m = 2.1 \times \left(1-\frac{1.1}{m}\right)^m \geq 2.1 \times \left(1-\frac{1.1}{2}\right)^2 \geq \frac{1}{7}. \quad \blacktriangleleft$$

▶ **Lemma 10.** *For any $k \in \mathbb{N}$ and $t \in \{1, \ldots, k-1\}$,*

$$\binom{k}{t}\left(\frac{t}{k}\right)^t \left(\frac{k-t}{k}\right)^{k-t} \geq \frac{1}{3\sqrt{t}}.$$

Proof of Lemma 10. Stirling's approximation states that $\sqrt{2\pi n}\cdot(n/e)^n \leq n! \leq e\sqrt{n}\cdot(n/e)^n$ for any positive integer n. Thus,

$$\binom{k}{t} = \frac{k!}{t!(k-t)!} \geq \frac{\sqrt{2\pi k}}{e^2\sqrt{t(k-t)}} \cdot \frac{(k/e)^k}{(t/e)^t[(k-t)/e]^{k-t}} = \frac{\sqrt{2\pi}}{e^2} \cdot \sqrt{\frac{k}{t(k-t)}} \cdot \frac{k^k}{t^t(k-t)^{k-t}}.$$

We conclude that

$$\binom{k}{t}\left(\frac{t}{k}\right)^t \left(\frac{k-t}{k}\right)^{k-t} \geq \frac{\sqrt{2\pi}}{e^2} \cdot \sqrt{\frac{k}{t(k-t)}} \geq \frac{1}{3\sqrt{t}}. \quad \blacktriangleleft$$

B Tensorization of the Total Variation Distance

In this section we prove two inequalities regarding the total variation distance between k-fold product distributions.

▶ **Lemma 11.** *For any $\epsilon \in (0, 1/2)$ and $k \in \mathbb{N}$, there exist Bernoulli distributions P and Q such that:*
1. $\Delta(P, Q) = \epsilon$.
2. $\Delta(P^{\otimes k}, Q^{\otimes k}) \le \epsilon\sqrt{2k}$.

In particular, Bernoulli distributions with means $(1-\epsilon)/2$ and $(1+\epsilon)/2$ satisfy the above conditions.

Proof of Lemma 11. Let P and Q be Bernoulli distributions with means $(1-\epsilon)/2$ and $(1+\epsilon)/2$, respectively. Clearly, we have $\Delta(P, Q) = \epsilon$ and

$$\mathrm{KL}(P, Q) = \epsilon \ln \frac{1+\epsilon}{1-\epsilon} \le \frac{2\epsilon^2}{1-\epsilon} \le 4\epsilon^2.$$

Here the second step applies the inequality $\ln(1+x) \le x$. By Pinsker's inequality,

$$\Delta(P^{\otimes k}, Q^{\otimes k}) \le \sqrt{\frac{1}{2}\mathrm{KL}(P^{\otimes k}, Q^{\otimes k})} = \sqrt{\frac{k}{2}\mathrm{KL}(P, Q)} \le \epsilon\sqrt{2k}. \qquad \blacktriangleleft$$

The next lemma shows that for sufficiently small ϵ, the $O(\sqrt{k})$ ratio between $\Delta(P^{\otimes k}, Q^{\otimes k})$ and $\Delta(P, Q)$ in Lemma 11 is tight (up to a constant factor).

▶ **Lemma 12.** *Suppose that $k \in \mathbb{N}$ and $\epsilon \in \left(0, 1/(15\sqrt{k})\right)$. P and Q are two distributions on the same support such that $\Delta(P, Q) \ge \epsilon$. Then, $\Delta(P^{\otimes k}, Q^{\otimes k}) \ge \frac{\epsilon\sqrt{k}}{15}$.*

To prove Lemma 12, we have the following weaker claim for the special case of Bernoulli distributions.

▶ **Lemma 13.** *Suppose that $k \in \mathbb{N}$, $\epsilon \in (0, 1/(15\sqrt{k}))$, $p, q \in [0, 1]$ and $q - p \ge \epsilon$. $\tilde{P}$ is a mixture of binomial distributions with k trials and success probabilities in $[0, p]$, i.e., $\tilde{P} = \sum_{i=1}^{\mathrm{tot}} \alpha_i B(k, p_i)$ for some nonnegative weights $\alpha_1, \ldots, \alpha_{\mathrm{tot}}$ and probabilities $p_1, \ldots, p_{\mathrm{tot}}$ in $[0, p]$, such that $\sum_{i=1}^{\mathrm{tot}} \alpha_i = 1$. $\tilde{Q}$ is a mixture of binomial distributions with k trials and success probabilities in $[q, 1]$. Then, $\Delta(\tilde{P}, \tilde{Q}) \ge \frac{\epsilon\sqrt{k}}{15}$.*

We prove Lemma 12 by a reduction to Bernoulli distributions.

Proof of Lemma 12. Suppose that P and Q share the support $[n]$. Define $\pi : [n] \to \{0, 1\}$ as $\pi(x) = \begin{cases} 1, & P(x) \ge Q(x), \\ 0, & P(x) < Q(x). \end{cases}$ Let p and q the means of $\pi(P)$ and $\pi(Q)$. Without loss of generality, $p \le q$. By construction, we have $|p - q| = \Delta(P, Q) \ge \epsilon$, and

$$\begin{aligned} \Delta(B(k, p), B(k, q)) &= \Delta((\pi(P_1), \pi(P_2), \ldots, \pi(P_k)), (\pi(Q_1), \pi(Q_2), \ldots, \pi(Q_k))) \\ &\le \Delta((P_1, P_2, \ldots, P_k), (Q_1, Q_2, \ldots, Q_k)) \\ &= \Delta(P^{\otimes k}, Q^{\otimes k}), \end{aligned}$$

where $(P_i)_{i\in[k]}$ and $(Q_i)_{i\in[k]}$ are independent copies of P and Q. Here the second step applies the data processing inequality. Applying Lemma 13 with $\tilde{P} = B(k, p)$ and $\tilde{Q} = B(k, q)$ yields that $\Delta(P^{\otimes k}, Q^{\otimes k}) \ge \Delta(B(k, p), B(k, q)) \ge \epsilon\sqrt{k}/15$. $\blacktriangleleft$

Now we turn to the more technical proof of Lemma 13.

Proof of Lemma 13. If $k < 10$, the inequality trivially follows from $\Delta(\tilde{P}, \tilde{Q}) \geq |p - q| > \epsilon\sqrt{k}/15$. Thus we assume that $k \geq 10$ in the following proof. Without loss of generality, we have $p \leq 1/2$ (otherwise we prove the lemma for $p' = 1 - q \leq 1/2$ and $q' = 1 - p$).

Let $t = \lfloor p(k-1) \rfloor$. Note that $t \leq (k-1)/2$ and $t \leq p(k-1) < t + 1$. Define function $f(x) = \sum_{j=0}^{t} \binom{k}{j} x^j (1-x)^{k-j}$. Note that

$$\begin{aligned} f'(x) &= k \sum_{j=0}^{t} \left[\binom{k-1}{j-1} x^{j-1}(1-x)^{k-j} - \binom{k-1}{j} x^j (1-x)^{k-j-1} \right] \\ &= -k \binom{k-1}{t} x^t (1-x)^{k-1-t} \leq 0, \end{aligned} \tag{6}$$

and thus f is non-increasing.

Since $f(x)$ is the probability that the binomial distribution $B(k, x)$ assigns to set $\{0, 1, \ldots, t\}$, $\tilde{P}(\{0, 1, \ldots, t\})$ can be written as a weighted average of $f(p_i)$'s for $p_1, p_2, \ldots \in [0, p]$. Similarly, $\tilde{Q}(\{0, 1, \ldots, t\})$ is a weighted average of $f(q_i)$'s for $q_1, q_2, \ldots \in [q, 1]$. Since $\tilde{P}(\{0, 1, \ldots, t\}) - \tilde{Q}(\{0, 1, \ldots, t\})$ is a lower bound on $\Delta(\tilde{P}, \tilde{Q})$, it remains to show that $f(p_i) - f(q_i) \geq \epsilon\sqrt{k}/15$ for any $p_i \leq p$ and $q_i \geq q$. The monotonicity of f and the fact that $q \geq p + \epsilon$ further imply that it suffices to prove $f(p) - f(p + \epsilon) \geq \epsilon\sqrt{k}/15$. We prove the inequality in the following two cases.

Case 1: $t = 0$. In this case, we have $0 \leq p < 1/(t-1)$. Note that

$$f(p) - f(p + \epsilon) = -\epsilon f'(x) = \epsilon k (1-x)^{k-1}$$

for some $x \in (p, p + \epsilon)$. If $\epsilon \leq 1/k$, we have $x \leq p + \epsilon \leq 2/(k-1)$, and

$$f(p) - f(p + \epsilon) \geq \epsilon k \left(1 - \frac{2}{k-1}\right)^{k-1} \geq \epsilon k \left(1 - \frac{2}{9}\right)^{9} \geq \frac{\epsilon k}{10}. \tag{7}$$

Here the second step follows from Lemma 8 and the assumption that $k \geq 10$. It then follows that $f(p) - f(p + \epsilon) \geq \epsilon\sqrt{k}/15$.
If $\epsilon \geq 1/k$, by Inequality (7) and the assumption that $\epsilon < 1/(15\sqrt{k})$,

$$f(p) - f(p + \epsilon) \geq f(p) - f(p + 1/k) \geq \frac{1}{10} \geq \frac{\epsilon\sqrt{k}}{15}.$$

Case 2: $t > 0$. Let $x_0 = t/(k-1)$. By Equation (6) and Lemma 10, we have

$$|f'(x_0)| = k \binom{k-1}{t} \left(\frac{t}{k-1}\right)^t \left(\frac{k-1-t}{k-1}\right)^{k-1-t} \geq \frac{k}{3\sqrt{t}}.$$

For any $x \in [p, p + \epsilon]$, we can write $x = (t + \alpha)/(k-1)$ for some $\alpha \geq 0$. Then,

$$\frac{|f'(x)|}{|f'(x_0)|} = \left(1 + \frac{\alpha}{t}\right)^t \left(1 - \frac{\alpha}{k-1-t}\right)^{k-1-t}$$

Since $t \leq (k-1)/2$ and $k \geq 10$, we have $k - 1 - t \geq \max(t, (k-1)/2) \geq \max(t, 2)$. Applying Lemma 9 with $n = t$ and $m = k - 1 - t$ yields that

$$\left(1 + \frac{\alpha}{t}\right)^t \left(1 - \frac{\alpha}{k-1-t}\right)^{k-1-t} \geq \frac{1}{7}$$

for any $\alpha \in [0, 1.1\sqrt{t}]$. Consequently, $|f'(x)| \geq \frac{|f'(x_0)|}{7} \geq \frac{k}{21\sqrt{t}}$ for any $x \in [x_0, x_0 + 1.1\sqrt{t}/(k-1)]$.

Let l be the length of the intersection of $[p, p+\epsilon]$ and $[x_0, x_0 + 1.1\sqrt{t}/(k-1)]$. By our choice of t, we have $t \leq p(k-1) < t+1$, and thus, $x_0 \leq p < x_0 + 1/(k-1)$. This implies that

$$l = \min(p + \epsilon, x_0 + 1.1\sqrt{t}/(k-1)) - p \geq \min(\epsilon, \sqrt{t}/[10(k-1)]).$$

Therefore, we conclude that

$$f(p) - f(p+\epsilon) \geq l \cdot \frac{k}{21\sqrt{t}} \geq \frac{1}{21} \min\left(\frac{\epsilon k}{\sqrt{t}}, \frac{1}{10}\right) \geq \frac{1}{21} \min\left(\epsilon\sqrt{2k}, \frac{1}{10}\right) \geq \frac{\epsilon\sqrt{k}}{15}.$$

Here the last step holds since our assumption $\epsilon < 1/(15\sqrt{k})$ implies that $\epsilon\sqrt{2k} < 1/10$. ◀

C Missing Proofs from Section 4

C.1 Proof of Claim 1

Proof of Claim 1. By assumption, the frequency tensor A can be written as $A = (1-\epsilon)\hat{P} + \epsilon N$, where $\hat{P}$ and N are probability n^t-tensors. In particular, $\hat{P}$ denotes the frequency among the m "good" samples drawn from $p^{\otimes t}$, while N denotes the frequency among the other batches. We prove the lemma by induction on t.

Base case. When $t = 1$, we have $\mathsf{DistSet}(n, t, A) = \{A\}$, and thus,

$$\min_{q \in \mathsf{DistSet}(n,t,A)} \Delta(p, q) = \Delta(p, A) \leq \Delta(p, \hat{P}) + \Delta(\hat{P}, A) = \Delta(p, \hat{P}) + \epsilon\Delta(\hat{P}, N) \leq \Delta(p, \hat{P}) + \epsilon.$$

It is well-known that with probability $1 - \delta_0$, $\Delta(p, \hat{P}) \leq C\sqrt{\frac{n + \ln \delta_0^{-1}}{m}}$ for some absolute constant C. Therefore,

$$\min_{q \in \mathsf{DistSet}(n,t,A)} \Delta(p, q) \leq C\sqrt{\frac{n + \ln \delta_0^{-1}}{m}} + \epsilon \leq f_{n,1}(\epsilon, m).$$

Inductive step. Let $\hat{p}$ and δ be the marginals of $\hat{P}$ and N, respectively. Then the marginal of A is given by $a = (1-\epsilon)\hat{p} + \epsilon\delta$. Moreover, the i-th slice of A after normalization is given by

$$A_i = \frac{(1-\epsilon)\hat{p}_i \cdot \hat{P}_i + \epsilon\delta_i \cdot N_i}{(1-\epsilon)\hat{p}_i + \epsilon\delta_i} = (1 - \epsilon_i')\hat{P}_i + \epsilon_i' N_i,$$

where $\epsilon_i' = \frac{\epsilon\delta_i}{(1-\epsilon)\hat{p}_i + \epsilon\delta_i}$. Moreover, A_i contains $m_i' = m \cdot \hat{p}_i$ samples drawn from $p^{\otimes(t-1)}$. Let $\alpha = (1 - \sqrt[t]{1-\epsilon})/\epsilon$ and $\beta = 1/t$. We consider the following two cases.

Case 1: For every $i \in [n]$, either $\delta_i/\hat{p}_i \geq 1 - \alpha$ or $\hat{p}_i \leq \beta/n$.

Since $a \in \mathsf{DistSet}(n, t, A)$, we have

$$\min_{q \in \mathsf{DistSet}(n,t,A)} \Delta(p, q) \leq \Delta(p, a) \leq \Delta(p, \hat{p}) + \Delta(\hat{p}, a) = \Delta(p, \hat{p}) + \epsilon\Delta(\hat{p}, \delta).$$

Moreover, since for any $i \in [n]$, either $\delta_i \geq (1-\alpha)\hat{p}_i$ or $\hat{p}_i \leq \beta/n$, it holds that $\hat{p}_i - \delta_i \leq \alpha\hat{p}_i + \beta/n$, and thus

$$\Delta(\hat{p}, \delta) = \sum_{i=1}^{n} \max(\hat{p}_i - \delta_i, 0) \leq \sum_{i=1}^{n} (\alpha\hat{p}_i + \beta/n) = \alpha + \beta.$$

It follows that

$$\min_{q \in \mathsf{DistSet}(n,t,A)} \Delta(p,q) \leq \Delta(p,\hat{p}) + \epsilon(\alpha+\beta) = \Delta(p,\hat{p}) + 1 - \sqrt[t]{1-\epsilon} + \epsilon/t \leq \frac{3\epsilon}{t} + \Delta(p,\hat{p}).$$

Here the last step $\sqrt[t]{1-\epsilon} \geq 1 - 2\epsilon/t$ follows from Lemma 7.

Case 2: For some $i \in [n]$, both $\delta_i/\hat{p}_i < 1-\alpha$ and $\hat{p}_i > \beta/n$ hold. In this case, we have

$$\epsilon_i' = \frac{\epsilon(\delta_i/\hat{p}_i)}{1 - \epsilon + \epsilon(\delta_i/\hat{p}_i)} \leq \frac{\epsilon(1-\alpha)}{1-\epsilon\alpha} = 1 - (1-\epsilon)^{(t-1)/t}$$

and $m_i' = m \cdot \hat{p}_i \geq \beta m/n = m/(nt)$. We have the following bound:

$$\min_{q \in \mathsf{DistSet}(n,t,A)} \Delta(p,q) \leq \min_{q \in \mathsf{DistSet}(n,t-1,A_i)} \Delta(p,q).$$

Combining the two cases shows that $\min_{q \in \mathsf{DistSet}(n,t,A)} \Delta(p,q)$ is upper bounded by either $3\epsilon/t + \Delta(p,\hat{p})$ or $\min_{q \in \mathsf{DistSet}(n,t-1,A_i)} \Delta(p,q)$ for some $i \in [n]$ such that $\epsilon_i' \leq 1 - (1-\epsilon)^{(t-1)/t}$ and $m_i' \geq m/(nt)$. According to the induction hypothesis, for each $i \in [n]$, it holds with probability $1 - (n+1)^{t-2} \cdot \delta_0$ that

$$\min_{q \in \mathsf{DistSet}(n,t-1,A_i)} \Delta(p,q) \leq f_{n,t-1}\left(\epsilon_i', m_i'\right).$$

Moreover, with probability $1-\delta_0$, $\Delta(p,\hat{p}) \leq C\sqrt{\frac{n+\ln\delta_0^{-1}}{m}}$. By a union bound, with probability $1 - n \cdot (n+1)^{t-2}\delta_0 - \delta_0 \geq 1 - (n+1)^{t-1}\delta_0$, it holds that

$$\begin{aligned}
&\min_{q \in \mathsf{DistSet}(n,t,A)} \Delta(p,q) \\
&\leq \max\left(\frac{3\epsilon}{t} + C\sqrt{\frac{n+\ln\delta_0^{-1}}{m}}, f_{n,t-1}\left(1-(1-\epsilon)^{(t-1)/t}, \frac{m}{nt}\right)\right) \\
&= f_{n,t}(\epsilon, m).
\end{aligned}$$

This completes the inductive step. ◀

C.2 Proof of Claim 2

Proof of Claim 2. Define $(\epsilon_k, m_k) = (\epsilon, m)$ and $(\epsilon_{t-1}, m_{t-1}) = \big(1-(1-\epsilon_t)^{(t-1)/t}, m_t/(nt)\big)$. Then by Equations (3) and (4),

$$\begin{aligned}
&f_{n,k}(\epsilon_k, m_k)\\
&\quad= \max\left(\frac{3\epsilon_k}{k} + C\sqrt{\frac{n+\ln\delta_0^{-1}}{m_k}}, f_{n,k-1}(\epsilon_{k-1}, m_{k-1})\right)\\
&\quad= \max\left(\frac{3\epsilon_k}{k} + C\sqrt{\frac{n+\ln\delta_0^{-1}}{m_k}}, \frac{3\epsilon_{k-1}}{k-1} + C\sqrt{\frac{n+\ln\delta_0^{-1}}{m_{k-1}}}, f_{n,k-2}(\epsilon_{k-2}, m_{k-2})\right)\\
&\quad= \cdots\\
&\quad= \max_{t\in[k]}\left(\frac{3\epsilon_t}{t} + C\sqrt{\frac{n+\ln\delta_0^{-1}}{m_t}}\right).
\end{aligned}$$

Moreover, by Lemma 7 and a simple induction, $\epsilon_t = 1-(1-\epsilon)^{t/k} \le 2t\epsilon/k$ and $m_t = m/(n^{k-t}A_k^{k-t})$, where A_n^m denotes $n(n-1)\cdots(n-m+1)$. It follows that for any $t \in [k]$,

$$\begin{aligned}
\frac{3\epsilon_t}{t} + C\sqrt{\frac{n+\ln\delta_0^{-1}}{m_t}} &\le \frac{3}{t}\cdot\frac{2t\epsilon}{k} + C\sqrt{\frac{n^{k-t}A_k^{k-t}(n+\ln\delta_0^{-1})}{m}}\\
&\le \frac{6\epsilon}{k} + C\sqrt{\frac{n^k\cdot k!\cdot(n+\ln\delta_0^{-1})}{m}}.
\end{aligned}$$

◀

Competing Bandits: Learning Under Competition*

Yishay Mansour[1], Aleksandrs Slivkins[2], and Zhiwei Steven Wu[3]

1 **Tel Aviv University, Tel Aviv, Israel**
`mansour.yishay@gmail.com`
2 **Microsoft Research, New York City, USA**
`slivkins@microsoft.com`
3 **Microsoft Research, New York City, USA**
`zhiww@microsoft.com`

Abstract

Most modern systems strive to learn from interactions with users, and many engage in *exploration*: making potentially suboptimal choices for the sake of acquiring new information. We initiate a study of the interplay between *exploration and competition*—how such systems balance the exploration for learning and the competition for users. Here the users play three distinct roles: they are customers that generate revenue, they are sources of data for learning, and they are self-interested agents which choose among the competing systems.

In our model, we consider competition between two multi-armed bandit algorithms faced with the same bandit instance. Users arrive one by one and choose among the two algorithms, so that each algorithm makes progress if and only if it is chosen. We ask whether and to what extent competition incentivizes the adoption of better bandit algorithms. We investigate this issue for several models of user response, as we vary the degree of rationality and competitiveness in the model. Our findings are closely related to the "competition vs. innovation" relationship, a well-studied theme in economics.

1998 ACM Subject Classification F.1.1 Models of Computation

Keywords and phrases machine learning, game theory, competition, exploration, rationality

Digital Object Identifier 10.4230/LIPIcs.ITCS.2018.48

1 Introduction

Learning from interactions with users is ubiquitous in modern customer-facing systems, from product recommendations to web search to spam detection to content selection to fine-tuning the interface. Many systems purposefully implement *exploration*: making potentially suboptimal choices for the sake of acquiring new information. Randomized controlled trials, a.k.a. A/B testing, are an industry standard, with a number of companies such as *Optimizely* offering tools and platforms to facilitate them. Many companies use more sophisticated exploration methodologies based on *multi-armed bandits*, a well-known theoretical framework for exploration and making decisions under uncertainty.

Systems that engage in exploration typically need to compete against one another; most importantly, they compete for users. This creates an interesting tension between *exploration* and *competition*. In a nutshell, while exploring may be essential for improving the service tomorrow, it may degrade quality and make users leave *today*, in which case there will be no users to learn from! Thus, users play three distinct roles: they are customers that generate

* This research has been done while Y. Mansour was co-affiliated with Microsoft Research, and while Z.S. Wu was a graduate student at University of Pennsylvania.

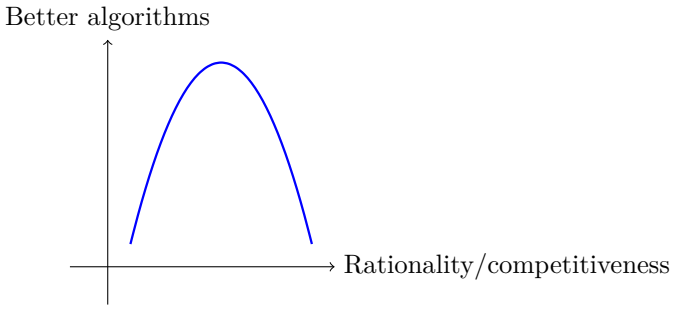


Figure 1 Inverted-U relationship between rationality/competitiveness and algorithms.

revenue, they generate data for the systems to learn from, and they are self-interested agents which choose among the competing systems.

We initiate a study of the interplay between *exploration* and *competition*. The main high-level question is: **whether and to what extent competition incentivizes adoption of better exploration algorithms**. This translates into a number of more concrete questions. While it is commonly assumed that better learning technology always helps, is this so for our setting? In other words, would a better learning algorithm result in higher utility for a principal? Would it be used in an equilibrium of the "competition game"? Also, does competition lead to better social welfare compared to a monopoly? We investigate these questions for several models, as we vary the capacity of users to make rational decisions (*rationality*) and the severity of competition between the learning systems (*competitiveness*). The two are controlled by the same "knob" in our models; such coupling is not unusual in the literature, *e.g.,* see [18].

On a high level, our contributions can be framed in terms of the "inverted-U relationship" between rationality/competitiveness and the quality of adopted algorithms (see Figure 1).

Our model. We define a game in which two firms (*principals*) simultaneously engage in exploration and compete for users (*agents*). These two processes are interlinked, as exploration decisions are experienced by users and informed by their feedback. We need to specify several conceptual pieces: how the principals and agents interact, what is the machine learning problem faced by each principal, and what is the information structure. Each piece can get rather complicated in isolation, let alone jointly, so we strive for simplicity. Thus, the basic model is as follows:

- A new agent arrives in each round $t = 1, 2, \ldots$, and chooses among the two principals. The principal chooses an action (*e.g.,* a list of web search results to show to the agent), the user experiences this action, and reports a reward. All agents have the same "decision rule" for choosing among the principals given the available information.
- Each principal faces a very basic and well-studied version of the multi-armed bandit problem: for each arriving agent, it chooses from a fixed set of actions (a.k.a. *arms*) and receives a reward drawn independently from a fixed distribution specific to this action.
- Principals simultaneously announce their learning algorithms before round 1, and cannot change them afterwards. There is a common Bayesian prior on the rewards (but the realized reward distributions are not observed by the principals or the agents). Agents do not receive any other information. Each principal only observes agents that chose him.

Technical results. Our results depend crucially on agents' "decision rule" for choosing among the principals. The simplest and perhaps the most obvious rule is to select the principal which maximizes their expected utility; we refer to it as `HardMax`. We find that `HardMax` is not conducive to adopting better algorithms. In fact, each principal's dominant strategy is to do no purposeful exploration whatsoever, and instead always choose an action that maximizes expected reward given the current information; we call this algorithm `DynamicGreedy`. While this algorithm may potentially try out different actions over time and acquire useful information, it is known to be dramatically bad in many important cases of multi-armed bandits — precisely because it does not explore on purpose, and may therefore fail to discover best/better actions. Further, we show that `HardMax` is very sensitive to tie-breaking when both principals have exactly the same expected utility according to agents' beliefs. If tie-breaking is probabilistically biased — say, principal 1 is always chosen with probability strictly larger than $\frac{1}{2}$ — then this principal has a simple "winning strategy" no matter what the other principal does.

We relax `HardMax` to allow each principal to be chosen with some fixed baseline probability. One intuitive interpretation is that there are "random agents" who choose a principal uniformly at random, and each arriving agent is either `HardMax` or "random" with some fixed probability. We call this model `HardMax&Random`. We find that better algorithms help in a big way: a sufficiently better algorithm is guaranteed to win all non-random agents after an initial learning phase. While the precise notion of "sufficiently better algorithm" is rather subtle, we note that commonly known "smart" bandit algorithms typically defeat the commonly known "naive" ones, and the latter typically defeat `DynamicGreedy`. However, there is a substantial caveat: one can defeat any algorithm by interleaving it with `DynamicGreedy`. This has two undesirable corollaries: a better algorithm may sometimes lose, and a pure Nash equilibrium typically does not exist.

We further relax the decision rule so that the probability of choosing a given principal varies smoothly as a function of the difference between principals' expected rewards; we call it `SoftMax`. For this model, the "better algorithm wins" result holds under much weaker assumptions on what constitutes a better algorithm. This is the most technical result of the paper. The competition in this setting is necessarily much more relaxed: typically, both principals attract approximately half of the agents as time goes by (but a better algorithm may attract slightly more).

All results extend to a much more general version of the multi-armed bandit problem in which the principal may observe additional feedback before and/or after each decision, as long as the feedback distribution does not change over time. In most results, principal's utility may depend on both the market share and agents' rewards.

Economic interpretation. The inverted-U relationship between the severity of competition among firms and the quality of technologies that they adopt is a familiar theme in the economics literature (*e.g.,* [2, 41]).[1] We find it illuminating to frame our contributions in a similar manner, as illustrated in Figure 1.

[1] The literature frames this relationship as one between "competition" and "innovation". In this context, "innovation" refers to adoption of a better technology, at a substantial R&D expense to a given firm. It is not salient whether similar ideas and/or technologies already exist outside the firm. It is worth noting that adoption of exploration algorithms tends to require substantial R&D effort in practice, even if the algorithms themselves are well-known in the research literature; see [1] for an example of such R&D effort.

Our models differ in terms of rationality in agents' decision-making: from fully rational decisions with `HardMax` to relaxed rationality with `HardMax&Random` to an even more relaxed rationality with `SoftMax`. The same distinctions also control the severity of competition between the principals: from cut-throat competition with `HardMax` to a more relaxed competition with `HardMax&Random`, to an even more relaxed competition with `SoftMax`. Indeed, with `HardMax` you lose all customers as soon as you fall behind in performance, with `HardMax&Random` you get some small market share no matter what, and with `SoftMax` you are further guaranteed a market share close to $\frac{1}{2}$ as long as your performance is not much worse than the competition. The uniform choice among principals corresponds to no rationality and no competition.

We identify the inverted-U relationship in the spirit of Figure 1 that is driven by the rationality/competitiveness distinctions outlined above: from `HardMax` to `HardMax&Random` to `SoftMax` to `Uniform`. We also find another, technically different inverted-U relationship which zeroes in on the `HardMax&Random` model. We vary rationality/competitiveness inside this model, and track the marginal utility of switching to a better algorithm.

These inverted-U relationships arise for a fundamentally different reason, compared to the existing literature on "competition vs. innovation." In the literature, better technology always helps in a competitive environment, other things being equal. Thus, the tradeoff is between the costs of improving the technology and the benefits that the improved technology provides in the competition. Meanwhile, we find that a better exploration algorithm may sometimes perform much worse under competition, even in the absence of R&D costs.

Discussion. We capture some pertinent features of reality while ignoring some others for the sake of tractability. Most notably, we assume that agents do not receive any information about other agents' rewards after the game starts. In the final analysis, this assumption makes agents' behavior independent of a particular realization of the Bayesian prior, and therefore enables us to summarize each learning algorithm via its Bayesian-expected rewards (as opposed to detailed performance on the particular realizations of the prior). Such summarization is essential for formulating lucid and general analytic results, let alone proving them. It is a major open question whether one can incorporate signals about other agents' rewards and obtain a tractable model.

We also make a standard assumption that agents are myopic: they do not worry about how their actions impact their future utility. In particular, they do not attempt to learn over time, to second-guess or game future agents, or to manipulate principal's learning algorithm. We believe this is a typical case in practice, in part because agent's influence tend to be small compared to the overall system. We model this simply by assuming that each agent only arrives once.

Much of the challenge in this paper, both conceptual and technical, was in setting up the right model and the matching theorems, and not only in proving the theorems. Apart from making the modeling choices described above, it was crucial to interpret the results and intuitions from the literature on multi-armed bandits so as to formulate meaningful assumptions on bandit algorithms and Bayesian priors which are productive in our setting.

Open questions. How to incorporate signals about the other agents' rewards? One needs to reason about how exact or coarse these signals are, and how the agents update their beliefs after receiving them. Also, one may need to allow principals' learning algorithms to respond to updates about the other principal's performance. (Or not, since this is not how learning algorithms are usually designed!) A clean, albeit idealized, model would be that (i)

each agent learns her exact expected reward from each principal before she needs to choose which principal to go to, but (ii) these updates are invisible to the principals. Even then, one needs to argue about the competition on particular realizations of the Bayesian prior, which appears very challenging.

Another promising extension is to heterogeneous agents. Then the agents' choices are impacted by their idiosyncratic signals/beliefs, instead of being entirely determined by priors and/or signals about the average performance. It would be particularly interesting to investigate the emergence of *specialization*: whether/when an algorithm learns to target specific population segments in order to compete against a more powerful "incumbent".

Map of the paper. We survey related work (Section 2), lay out the model and preliminaries (Section 3), and proceed to analyze the three main models, `HardMax`, `HardMax&Random` and `SoftMax` (in Sections 4, 5, and 6, respectively). We discuss economic implications in Section 7. Appendix A provides some pertinent background on multi-armed bandits. Appendix B gives a broad example to support an assumption in our model.

2 Related work

Multi-armed bandits (*MAB*) is a particularly elegant and tractable abstraction for tradeoff between *exploration* and *exploitation*: essentially, between acquisition and usage of information. MAB problems have been studied in Economics, Operations Research and Computer Science for many decades; see [13, 20, 39] for background on regret-minimizing and Bayesian formulations, respectively. A discussion of industrial applications of MAB can be found in [1].

The literature on MAB is vast and multi-threaded. The most related thread concerns regret-minimizing MAB formulations with IID rewards [29, 4]. This thread includes "smart" MAB algorithms that combine exploration and exploitation, such as UCB1 [4] and Successive Elimination [16], and "naive" MAB algorithms that separate exploration and exploitation, including explore-first and ϵ-Greedy *e.g.,* see [39].

The three-way tradeoff between exploration, exploitation and incentives has been studied in several other settings: incentivizing exploration in a recommendation system [14, 17, 28, 30, 11, 9, 31], dynamic auctions *e.g.,* [3, 10, 25], pay-per-click ad auctions with unknown click probabilities *e.g.,* [8, 15, 7], coordinating search and matching by self-interested agents [27], as well as human computation *e.g.,* [22, 19, 38].

[12, 26, 21] studied models with self-interested agents jointly performing exploration, with no principal to coordinate them.

There is a superficial similarity (in name only) between this paper and the line of work on "dueling bandits" *e.g.,* [43, 44]. The latter is not about competing bandit algorithms, but rather about scenarios where in each round two arms are chosen to be presented to a user, and the algorithm only observes which arm has "won the duel".

Our setting is closely related to the "dueling algorithms" framework [24] which studies competition between two principals, each running an algorithm for the same problem. However, this work considers algorithms for offline / full input scenarios, whereas we focus on online machine learning and the explore-exploit-incentives tradeoff therein. Also, this work specifically assumes binary payoffs (*i.e.,* win or lose) for the principals.

Other related work in economics. The competition vs. innovation relationship and the inverted-U shape thereof have been introduced in a classic book [37], and remained an important theme in the literature ever since *e.g.,* [2, 41]. Production costs aside, this

literature treats innovation as a priori beneficial for the firm. Our setting is very different, as innovation in exploration algorithms may potentially hurt the firm.

A line of work on *platform competition*, starting with [36], concerns competition between firms (*platforms*) that improve as they attract more users (*network effect*); see [42] for a recent survey. This literature is not concerned with *innovation*, and typically models network effects exogenously, whereas in our model network effects are endogenous: they are created by MAB algorithms, an essential part of the model.

Relaxed versions of rationality similar to ours are found in several notable lines of work. For example, "random agents" (a.k.a. noise traders) can side-step the "no-trade theorem" [32], a famous impossibility result in financial economics. The `SoftMax` model is closely related to the literature on *product differentiation*, starting from [23], see [34] for a notable later paper.

There is a large literature on non-existence of equilibria due to small deviations (which is related to the corresponding result for `HardMax`&`Random`), starting with [35] in the context of health insurance markets. Notable recent papers [40, 6] emphasize the distinction between `HardMax` and versions of `SoftMax`.

3 Our model and preliminaries

Principals and agents. There are two principals and T agents. The game proceeds in rounds (we will sometimes refer to them as *global rounds*). In each round $t \in [T]$, the following interaction takes place. A new agent arrives and chooses one of the two principals. The principal chooses a recommendation: an action $a_t \in A$, where A is a fixed set of actions (same for both principals and all rounds). The agent follows this recommendation, receives a reward $r_t \in [0, 1]$, and reports it back to the principal.

The rewards are i.i.d. with a common prior. More formally, for each action $a \in A$ there is a parametric family $\psi_a(\cdot)$ of reward distributions, parameterized by the mean reward μ_a. (The paradigmatic case is 0-1 rewards with a given expectation.) The mean reward vector $\mu = (\mu_a : a \in A)$ is drawn from prior distribution $\mathcal{P}_{\texttt{mean}}$ before round 1. Whenever a given action $a \in A$ is chosen, the reward is drawn independently from distribution $\psi_a(\mu_a)$. The prior $\mathcal{P}_{\texttt{mean}}$ and the distributions $(\psi_a(\cdot) : a \in A)$ constitute the (full) Bayesian prior on rewards, denoted $\mathcal{P}$.

Each principal commits to a learning algorithm for making recommendations. This algorithm follows a protocol of *multi-armed bandits* (*MAB*). Namely, the algorithm proceeds in time-steps:[2] each time it is called, it outputs a chosen action $a \in A$ and then inputs the reward for this action. The algorithm is called only in global rounds when the corresponding principal is chosen.

The information structure is as follows. The prior $\mathcal{P}$ is known to everyone. The mean rewards μ_a are not revealed to anybody. Each agent knows both principals' algorithms, and the global round when (s)he arrives, *but not* the rewards of the previous agents. Each principal is completely unaware of the rounds when the other is chosen.

Some terminology. The two principals are called "Principal 1" and "Principal 2". The algorithm of principal $i \in \{1, 2\}$ is called "algorithm i" and denoted $\texttt{alg}_i$. The agent in global round t is called "agent t"; the chosen principal is denoted i_t.

Throughout, $\mathbb{E}[\cdot]$ denotes expectation over all applicable randomness.

[2] These time-steps will sometimes be referred to as *local steps/rounds*, so as to distinguish them from "global rounds" defined before. We will omit the local vs. local distinction when clear from the context.

Bayesian-expected rewards. Consider the performance of a given algorithm $\texttt{alg}_i$, $i \in \{1,2\}$, when it is run in isolation (*i.e.,* without competition, just as a bandit algorithm). Let $\texttt{rew}_i(n)$ denote its Bayesian-expected reward for the n-th step.

Now, going back to our game, fix global round t and let $n_i(t)$ denote the number of global rounds before t in which this principal is chosen. Then:

$$\mathbb{E}[r_t \mid \text{principal } i \text{ is chosen in round } t \text{ and } n_i(t) = n] = \texttt{rew}_i(n+1) \quad (\forall n \in \mathbb{N}).$$

Agents' response. Each agent t chooses principal i_t as as follows: it chooses a distribution over the principals, and then draws independently from this distribution. Let p_t be the probability of choosing principal 1 according to this distribution. Below we specify p_t; we need to be careful so as to avoid a circular definition.

Let $\mathcal{I}_t$ be the information available to agent t before the round. Assume $\mathcal{I}_t$ suffices to form posteriors for quantities $n_i(t)$, $i \in \{1,2\}$, denote them by $\mathcal{N}_{i,t}$. Note that the Bayesian expected reward of each principal i is a function only of the number rounds he was chosen by the agents, so the posterior mean reward for each principal i can be written as

$$\texttt{PMR}_i(t) := \mathbb{E}[r_t \mid \mathcal{I}_t \text{ and } i_t = i] = \mathbb{E}[\texttt{rew}_i(n_i(t)+1) \mid \mathcal{I}_t] = \mathop{\mathbb{E}}_{n \sim \mathcal{N}_{i,t}}[\texttt{rew}_i(n+1)].$$

This quantity represents the posterior mean reward for principal i at round t, according to information $\mathcal{I}_t$; hence the notation $\texttt{PMR}$. In general, probability p_t is defined by the posterior mean rewards $\texttt{PMR}_i(t)$ for both principals. We assume a somewhat more specific shape:

$$p_t = f_{\texttt{resp}}\left(\texttt{PMR}_1(t) - \texttt{PMR}_2(t) \right). \tag{1}$$

Here $f_{\texttt{resp}} : [-1,1] \to [0,1]$ is the *response function*, which is the same for all agents. We assume that the response function is known to all agents.

To make the model well-defined, it remains to argue that information $\mathcal{I}_t$ is indeed sufficient to form posteriors on $n_1(t)$ and $n_2(t)$. This can be easily seen using induction on t.

Since all agents arrive with identical information (other than knowing which global round they arrive in), it follows that all agents have identical posteriors for $n_{i,t}$ (for a given principal i and a given global round t). This posterior is denoted $\mathcal{N}_{i,t}$.

Response functions. We use the response function $f_{\texttt{resp}}$ to characterize the amount of rationality and competitiveness in our model. We assume that $f_{\texttt{resp}}$ is monotonically non-decreasing, is larger than $\frac{1}{2}$ on the interval $(0,1]$, and smaller than $\frac{1}{2}$ on the interval $[-1,0)$. Beyond that, we consider three specific models, listed in the order of decreasing rationality and competitiveness (see Figure 2):

- $\texttt{HardMax}$: $f_{\texttt{resp}}$ equals 0 on the interval $[-1,0)$ and 1 on the interval $(0,1]$. In other words, the agents will deterministically choose the principal with the higher posterior mean reward.
- $\texttt{HardMax\&Random}$: $f_{\texttt{resp}}$ equals ϵ_0 on the interval $[-1,0)$ and $1-\epsilon_0$ on the interval $(0,1]$, where $\epsilon_0 \in (0, \frac{1}{2})$ are some positive constants. In words, each agent is a $\texttt{HardMax}$ agent with probability $1-2\epsilon_0$, and with the remaining probability she makes a random choice.
- $\texttt{SoftMax}$: $f_{\texttt{resp}}(\cdot)$ lies in the interval $[\epsilon_0, 1-\epsilon_0]$, $\epsilon_0 > 0$, and is "smooth" around 0 (in the sense defined precisely in Section 6).

We say that $f_{\texttt{resp}}$ is *symmetric* if $f_{\texttt{resp}}(-x) + f_{\texttt{resp}}(x) = 1$ for any $x \in [0,1]$. This implies *fair tie-breaking*: $f_{\texttt{resp}}(0) = \frac{1}{2}$.

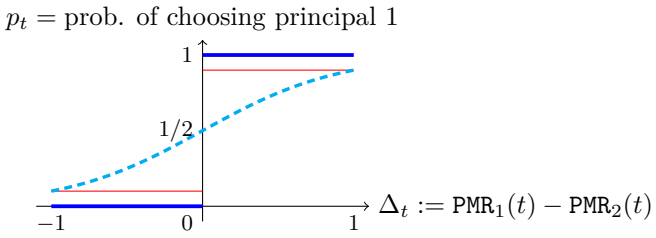


Figure 2 The three models for agents' response function: `HardMax` is thick blue, `HardMax&Random` is slim red, and `SoftMax` is the dashed curve.

MAB algorithms. We characterize the inherent quality of an MAB algorithm in terms of its *Bayesian Instantaneous Regret* (henceforth, `BIR`), a standard notion from machine learning:

$$\texttt{BIR}(n) := \mathop{\mathbb{E}}_{\mu\sim\mathcal{P}_{\texttt{mean}}}\left[\max_{a\in A}\mu_a\right] - \texttt{rew}(n), \tag{2}$$

where $\texttt{rew}(n)$ is the Bayesian-expected reward of the algorithm for the n-th step, when the algorithm is run in isolation. We are primarily interested in how `BIR` scales with n; we treat K, the number of arms, as a constant unless specified otherwise.

We will emphasize several specific algorithms or classes thereof:

- "smart" MAB algorithms that combine exploration and exploitation, such as UCB1 [4] and Successive Elimination [16]. These algorithms achieve $\texttt{BIR}(n) \leq \tilde{O}(n^{-1/2})$ for all priors and all (or all but a very few) steps n. This bound is known to be tight for any fixed n.[3]
- "naive" MAB algorithms that separate exploration and exploitation, such as Explore-then-Exploit and ϵ-Greedy. These algorithms have dedicated rounds in which they explore by choosing an action uniformly at random. When these rounds are known in advance, the algorithm suffers constant `BIR` in such rounds. When the "exploration rounds" are instead randomly chosen by the algorithm, one can usually guarantee an inverse-polynomial upper bound `BIR`, but not as good as the one above: namely, $\texttt{BIR}(n) \leq \tilde{O}(n^{-1/3})$. This is the best possible upper bound on `BIR` for the two algorithms mentioned above.
- `DynamicGreedy`: at each step, recommends the best action according to the current posterior: an action a with the highest posterior expected reward $\mathbb{E}[\mu_a \mid \mathcal{I}]$, where $\mathcal{I}$ is the information available to the algorithm so far. `DynamicGreedy` has (at least) a constant `BIR` for some reasonable priors, *i.e.,* $\texttt{BIR}(n) > \Omega(1)$.
- `StaticGreedy`: always recommends the prior best action,*i.e.,* an action a with the highest prior mean reward $\mathbb{E}_{\mu\sim\mathcal{P}_{\texttt{mean}}}[\mu_a]$. This algorithm typically has constant `BIR`.

We focus on MAB algorithms such that $\texttt{BIR}(n)$ is non-increasing; we call such algorithms *monotone*. While some reasonable MAB algorithms may occasionally violate monotonicity, they can usually be easily modified so that monotonicity violations either vanish altogether, or only occur at very specific rounds (so that agents are extremely unlikely to exploit them in practice).

More background and examples can be found in Appendix A. In particular, we prove that `DynamicGreedy` is monotone.

[3] This follows from the lower-bound analysis in [5].

Competition game between principals. Some of our results explicitly study the game between the two principals. We model it as a simultaneous-move game: before the first agent arrives, each principal commits to an MAB algorithm. Thus, choosing a pure strategy in this game corresponds to choosing an MAB algorithm (and, implicitly, announcing this algorithm to the agents).

Principal's utility is primarily defined as the market share, *i.e.,* the number of agents that chose this principal. Principals are risk-neutral, in the sense that they optimize their expected utility.

Assumptions on the prior. We make some technical assumptions for the sake of simplicity. First, each action a has a positive probability of being the best action according to the prior:

$$\forall a \in A: \quad \Pr_{\mu\sim\mathcal{P}_{\texttt{mean}}}[\mu_a > \mu_{a'}\ \forall a' \in A] > 0. \tag{3}$$

Second, posterior mean rewards of actions are pairwise distinct almost surely. That is, the history h at any step of an MAB algorithm[4] satisfies

$$\mathbb{E}[\mu_a \mid h] \neq \mathbb{E}[\mu_{a'} \mid h] \quad \forall a, a' \in A, \tag{4}$$

except at a set of histories of probability 0. In particular, prior mean rewards of actions are pairwise distinct: $\mathbb{E}[\mu_a] \neq \mathbb{E}[\mu'_a]$ for any $a, a' \in A$.

We provide two examples for which property (4) is 'generic', in the sense that it can be enforced almost surely by a small random perturbation of the prior. Both examples focus on 0-1 rewards and priors $\mathcal{P}_{\texttt{mean}}$ that are independent across arms. The first example assumes Beta priors on the mean rewards, and is very easy.[5] The second example assumes that mean rewards have a finite support, see Appendix B for details.

Some more notation. Without loss of generality, we label actions as $A = [K]$ and sort them according to their prior mean rewards, so that $\mathbb{E}[\mu_1] > \mathbb{E}[\mu_2] > \ldots > \mathbb{E}[\mu_K]$.

Fix principal $i \in \{1, 2\}$ and (local) step n. The arm chosen by algorithm $\texttt{alg}_i$ at this step is denoted $a_{i,n}$, and the corresponding $\texttt{BIR}$ is denoted $\texttt{BIR}_i(n)$. History of $\texttt{alg}_i$ up to this step is denoted $H_{i,n}$.

Write $\texttt{PMR}(a \mid E) = \mathbb{E}[\mu_a \mid E]$ for posterior mean reward of action a given event E.

3.1 Generalizations

Our results can be extended compared to the basic model described above.

First, unless specified otherwise, our results allow a more general notion of principal's utility that can depend on both the market share and agents' rewards. Namely, principal i collects $U_i(r_t)$ units of utility in each global round t when she is chosen (and 0 otherwise), where $U_i(\cdot)$ is some fixed non-decreasing function with $U_i(0) > 0$. In a formula,

$$U_i := \sum_{t=1}^{T} \mathbf{1}_{\{i_t=i\}} \cdot U_i(r_r). \tag{5}$$

[4] The *history* of an MAB algorithm at a given step comprises the chosen actions and the observed rewards in all previous steps in the execution of this algorithm.

[5] Suppose the rewards are Bernouli r.v. and the mean reward μ_a for each arm a is drawn from some Beta distribution $\text{Beta}(\alpha_a, \beta_a)$. Given any history that contains h_a number of heads and t_a number of tails from arm a, the posterior mean reward is $\frac{\alpha_a+h_a}{\alpha_a+h_a+\beta_a+t_a}$. Note that h_a and t_a take integer values. Therefore, perturbing the parameters α_a and β_a independently with any continuous noise will induce a prior with property (4) with probability 1.

Second, our results carry over, with little or no modification of the proofs, to much more general versions of MAB, as long as it satisfies the i.i.d. property. In each round, an algorithm can see a *context* before choosing an action (as in *contextual bandits*) and/or additional feedback other than the reward after the reward is chosen (as in, *e.g., semi-bandits*), as long as the contexts are drawn from a fixed distribution, and the (reward, feedback) pair is drawn from a fixed distribution that depends only on the context and the chosen action. The Bayesian prior $\mathcal{P}$ needs to be a more complicated object, to make sure that `PMR` and `BIR` are well-defined. Mean rewards may also have a known structure, such as Lipschitzness, convexity, or linearity; such structure can be incorporated via $\mathcal{P}$. All these extensions have been studied extensively in the literature on MAB, and account for a substantial segment thereof; see [13] for background and details.

3.2 Chernoff Bounds

We use an elementary concentration inequality known as *Chernoff Bounds*, in a formulation from [33].

▶ **Theorem 1** (Chernoff Bounds). *Consider n i.i.d. random variables $X_1 \dots X_n$ with values in $[0,1]$. Let $X = \frac{1}{n}\sum_{i=1}^{n} X_i$ be their average, and let $\nu = \mathbb{E}[X]$. Then:*

$$\min\left(\Pr[X - \nu > \delta\nu], \quad \Pr[\nu - X > \delta\nu]\right) < e^{-\nu n \delta^2/3} \quad \text{for any } \delta \in (0,1).$$

4 Full rationality (HardMax)

In this section, we will consider the version in which the agents are fully rational, in the sense that their response function is `HardMax`. We show that principals are not incentivized to *explore*— *i.e.,* to deviate from `DynamicGreedy`. The core technical result is that if one principal adopts `DynamicGreedy`, then the other principal loses all agents as soon as he deviates.

To make this more precise, let us say that two MAB algorithms *deviate* at (local) step n if there is an action $a \in A$ and a set of step-n histories of positive probability such that any history h in this set is feasible for both algorithms, and under this history the two algorithms choose action a with different probability.

▶ **Theorem 2.** *Assume* `HardMax` *response function with fair tie-breaking. Assume that* $\texttt{alg}_1$ *is* `DynamicGreedy`*, and* $\texttt{alg}_2$ *deviates from* `DynamicGreedy` *starting from some (local) step $n_0 < T$. Then all agents in global rounds $t \geq n_0$ select principal 1.*

▶ **Corollary 3.** *The competition game between principals has a unique Nash equilibirium: both principals choose* `DynamicGreedy`*.*

▶ Remark. This corollary holds under a more general model which allows time-discounting: namely, the utility of each principal i in each global round t is $U_{i,t}(r_t)$ if this principal is chosen, and 0 otherwise, where $U_{i,t}(\cdot)$ is an arbitrary non-decreasing function with $U_{i,t}(0) > 0$.

4.1 Proof of Theorem 2

The proof starts with two auxiliary lemmas: that deviating from `DynamicGreedy` implies a strictly smaller Bayesian-expected reward, and that `HardMax` implies a "sudden-death" property: if one agent chooses principal 1 with certainty, then so do all subsequent agents do. We re-use both lemmas in later sections, so we state them in sufficient generality.

▶ **Lemma 4.** *Assume that* $\texttt{alg}_1$ *is* `DynamicGreedy`*, and* $\texttt{alg}_2$ *deviates from* `DynamicGreedy` *starting from some (local) step* $n_0 < T$*. Then* $\texttt{rew}_1(n_0) > \texttt{rew}_2(n_0)$*. This holds for any response function* $f_{\texttt{resp}}$*.*

Lemma 4 does not rely on any particular shape of the response function because it only considers the performance of each algorithm without competition.

Proof of Lemma 4. Since the two algorithms coincide on the first $n_0 - 1$ steps, it follows by symmetry that histories H_{1,n_0} and H_{2,n_0} have the same distribution. We use a *coupling argument*: w.l.o.g., we assume the two histories coincide, $H_{1,n_0} = H_{2,n_0} = H$.

At local step n_0, `DynamicGreedy` chooses an action $a_{1,n_0} = a_{1,n_0}(H)$ which maximizes the posterior mean reward given history H: for any realized history $h \in \texttt{support}(H)$ and any action $a \in A$

$$\texttt{PMR}(a_{1,n_0} \mid H = h) \geq \texttt{PMR}(a \mid H = h). \tag{6}$$

By assumption (4), it follows that

$$\texttt{PMR}(a_{1,n_0} \mid H = h) > \texttt{PMR}(a \mid H = h) \quad \text{for any } h \in \texttt{support}(H) \text{ and } a \neq a_{1,n_0}(h). \tag{7}$$

Since the two algorithms deviate at step n_0, there is a set $S \subset \texttt{support}(H)$ of step-n_0 histories such that $\Pr[S] > 0$ and any history $h \in S$ satisfies $\Pr[a_{2,n_0} \neq a_{1,n_0} \mid H = h] > 0$. Combining this with (7), we deduce that

$$\texttt{PMR}(a_{1,n_0} \mid H = h) > \mathbb{E}\left[\mu_{a_{2,n_0}} \mid H = h\right] \quad \text{for each history } h \in S. \tag{8}$$

Using (6) and (8) and integrating over realized histories h, we obtain $\texttt{rew}_1(n_0) > \texttt{rew}_2(n_0)$. ◀

▶ **Lemma 5.** *Consider* `HardMax` *response function with* $f_{\texttt{resp}}(0) \geq \frac{1}{2}$*. Suppose* $\texttt{alg}_1$ *is monotone, and* $\texttt{PMR}_1(t_0) > \texttt{PMR}_2(t_0)$ *for some global round* t_0*. Then* $\texttt{PMR}_1(t) > \texttt{PMR}_2(t)$ *for all subsequent rounds* t*.*

Proof. Let us use induction on round $t \geq t_0$, with the base case $t = t_0$. Let $\mathcal{N} = \mathcal{N}_{1,t_0}$ be the agents' posterior distribution for n_{1,t_0}, the number of global rounds before t_0 in which principal 1 is chosen. By induction, all agents from t_0 to $t-1$ chose principal 1, so $\texttt{PMR}_2(t_0) = \texttt{PMR}_2(t)$. Therefore,

$$\texttt{PMR}_1(t) = \mathop{\mathbb{E}}_{n\sim\mathcal{N}}[\texttt{rew}_1(n+1+t-t_0)] \geq \mathop{\mathbb{E}}_{n\sim\mathcal{N}}[\texttt{rew}_1(n+1)] = \texttt{PMR}_1(t_0) > \texttt{PMR}_2(t_0) = \texttt{PMR}_2(t),$$

where the first inequality holds because $\texttt{alg}_1$ is monotone, and the second one is the base case. ◀

Proof of Theorem 2. Since the two algorithms coincide on the first $n_0 - 1$ steps, it follows by symmetry that $\texttt{rew}_1(n) = \texttt{rew}_2(n)$ for any $n < n_0$. By Lemma 4, $\texttt{rew}_1(n_0) > \texttt{rew}_2(n_0)$.

Recall that $n_i(t)$ is the number of global rounds $s < t$ in which principal i is chosen, and $\mathcal{N}_{i,t}$ is the agents' posterior distribution for this quantity. By symmetry, each agent $t < n_0$ chooses a principal uniformly at random. It follows that $\mathcal{N}_{1,n_0} = \mathcal{N}_{2,n_0}$ (denote both distributions by $\mathcal{N}$ for brevity), and $\mathcal{N}(n_0 - 1) > 0$. Therefore:

$$\begin{aligned}
\texttt{PMR}_1(n_0) &= \mathop{\mathbb{E}}_{n\sim\mathcal{N}}[\texttt{rew}_1(n+1)] = \sum_{n=0}^{n_0-1} \mathcal{N}(n)\cdot\texttt{rew}_1(n+1) \\
&> \mathcal{N}(n_0-1)\cdot\texttt{rew}_2(n_0) + \sum_{n=0}^{n_0-2} \mathcal{N}(n)\cdot\texttt{rew}_2(n+1) \\
&= \mathop{\mathbb{E}}_{n\sim\mathcal{N}}[\texttt{rew}_2(n+1)] = \texttt{PMR}_2(n_0)
\end{aligned} \tag{9}$$

So, agent n_0 chooses principal 1. By Lemma 5 (noting that DynamicGreedy is monotone), all subsequent agents choose principal 1, too. ◀

4.2 HardMax with biased tie-breaking

The HardMax model is very sensitive to the tie-breaking rule. For starters, if ties are broken deterministically in favor of principal 1, then principal 1 can get all agents no matter what the other principal does, simply by using StaticGreedy.

▶ **Theorem 6.** *Assume* HardMax *response function with* $f_{\texttt{resp}}(0) = 1$ *(ties are always broken in favor of principal* 1*). If* $\texttt{alg}_1$ *is* StaticGreedy*, then all agents choose principal* 1*.*

Proof. Agent 1 chooses principal 1 because of the tie-breaking rule. Since StaticGreedy is trivially monotone, all the subsequent agents choose principal 1 by an induction argument similar to the one in the proof of Lemma 5. ◀

A more challenging scenario is when the tie-breaking is biased in favor of principal 1, but not deterministically so: $f_{\texttt{resp}}(0) > \frac{1}{2}$. Then this principal also has a "winning strategy" no matter what the other principal does. Specifically, principal 1 can get all but the first few agents, under a mild technical assumption that DynamicGreedy deviates from StaticGreedy. Principal 1 can use DynamicGreedy, or any other monotone MAB algorithm that coincides with DynamicGreedy in the first few steps.

▶ **Theorem 7.** *Assume* HardMax *response function with* $f_{\texttt{resp}}(0) > \frac{1}{2}$ *(*i.e.*, tie-breaking is biased in favor of principal* 1*). Assume the prior* $\mathcal{P}$ *is such that* DynamicGreedy *deviates from* StaticGreedy *starting from some step* n_0*. Suppose that principal* 1 *runs a monotone MAB algorithm that coincides with* DynamicGreedy *in the first* n_0 *steps. Then all agents* $t \geq n_0$ *choose principal* 1*.*

Proof. The proof re-uses Lemmas 4 and 5, which do not rely on fair tie-breaking.

Because of the biased tie-breaking, for each global round t we have:

$$\text{if } \texttt{PMR}_1(t) \geq \texttt{PMR}_2(t) \text{ then } \Pr[i_t = 1] > \tfrac{1}{2}. \tag{10}$$

Recall that i_t is the principal chosen in global round t.

Let m_0 be the first step when $\texttt{alg}_2$ deviates from DynamicGreedy, or DynamicGreedy deviates from StaticGreedy, whichever comes sooner. Then $\texttt{alg}_2$, DynamicGreedy and StaticGreedy coincide on the first $m_0 - 1$ steps. Moreover, $m_0 \leq n_0$ (since DynamicGreedy deviates from StaticGreedy at step n_0), so $\texttt{alg}_1$ coincides with DynamicGreedy on the first m_0 steps.

So, $\texttt{rew}_1(n) = \texttt{rew}_2(n)$ for each step $n < m_0$, because $\texttt{alg}_1$ and $\texttt{alg}_2$ coincide on the first $m_0 - 1$ steps. Moreover, if $\texttt{alg}_2$ deviates from DynamicGreedy at step m_0 then $\texttt{rew}_1(m_0) > \texttt{rew}_2(m_0)$ by Lemma 4; else, we trivially have $\texttt{rew}_1(m_0) = \texttt{rew}_2(m_0)$. To summarize:

$$\texttt{rew}_1(n) \geq \texttt{rew}_2(n) \quad \text{for all steps } n \leq m_0. \tag{11}$$

We claim that $\Pr[i_t = 1] > \frac{1}{2}$ for all global rounds $t \leq m_0$. We prove this claim using induction on t. The base case $t = 1$ holds by (10) and the fact that in step 1, DynamicGreedy chooses the arm with the highest prior mean reward. For the induction step, we assume that

$\Pr[i_t = 1] > \frac{1}{2}$ for all global rounds $t < t_0$, for some $t_0 \leq m_0$. It follows that distribution $\mathcal{N}_{1,t_0}$ stochastically dominates distribution $\mathcal{N}_{2,t_0}$.[6] Observe that

$$\texttt{PMR}_1(t_0) = \mathop{\mathbb{E}}_{n\sim\mathcal{N}_{1,t_0}}[\texttt{rew}_1(n+1)] \geq \mathop{\mathbb{E}}_{n\sim\mathcal{N}_{2,t_0}}[\texttt{rew}_2(n+1)] = \texttt{PMR}_2(t_0). \tag{12}$$

So the induction step follows by (10). Claim proved.

Now let us focus on global round m_0, and denote $\mathcal{N}_i = \mathcal{N}_{i,m_0}$. By the above claim,

$$\mathcal{N}_1 \text{ stochastically dominates } \mathcal{N}_2, \text{ and moreover } \mathcal{N}_i(m_0-1) > \mathcal{N}_i(m_0-1). \tag{13}$$

By definition of m_0, either (i) $\texttt{alg}_2$ deviates from `DynamicGreedy` starting from local step m_0, which implies $\texttt{rew}_1(m_0) > \texttt{rew}_2(m_0)$ by Lemma 4, or (ii) `DynamicGreedy` deviates from `StaticGreedy` starting from local step m_0, which implies $\texttt{rew}_1(m_0) > \texttt{rew}_1(m_0-1)$ by Lemma 19. In both cases, using (11) and (13), it follows that the inequality in (12) is strict for $t_0 = m_0$.

Therefore, agent m_0 chooses principal 1, and by Lemma 5 so do all subsequent agents. ◀

5 Relaxed rationality: HardMax & Random

This section is dedicated to the `HardMax&Random` response model, where each principal is always chosen with some positive baseline probability. The main technical result for this model states that a principal with asymptotically better `BIR` wins by a large margin: after a "learning phase" of constant duration, all agents choose this principal with maximal possible probability $f_{\texttt{resp}}(1)$. For example, a principal with $\texttt{BIR}(n) \leq \tilde{O}(n^{-1/2})$ wins over a principal with $\texttt{BIR}(n) \geq \Omega(n^{-1/3})$. However, this positive result comes with a significant caveat detailed in Section 5.1.

We formulate and prove a cleaner version of the result, followed by a more general formulation developed in a subsequent Remark 5. We need to express a property that $\texttt{alg}_1$ eventually catches up and surpasses $\texttt{alg}_2$, even if initially it receives only a fraction of traffic. For the cleaner version, we assume that both algorithms are well-defined for an infinite time horizon, so that their `BIR` does not depend on the time horizon T of the game. Then this property can be formalized as:

$$(\forall \epsilon > 0) \qquad \texttt{BIR}_1(\epsilon n)/\texttt{BIR}_2(n) \to 0. \tag{14}$$

In fact, a weaker version of (14) suffices: denoting $\epsilon_0 = f_{\texttt{resp}}(-1)$, for some constant n_0 we have

$$(\forall n \geq n_0) \qquad \texttt{BIR}_1(\epsilon_0 n/2)/\texttt{BIR}_2(n) < \tfrac{1}{2}. \tag{15}$$

We also need a very mild technical assumption on the "bad" algorithm:

$$(\forall n \geq n_0) \qquad \texttt{BIR}_2(n) > 4\, e^{-\epsilon_0 n/12}. \tag{16}$$

▶ **Theorem 8.** *Assume* `HardMax&Random` *response function. Suppose both algorithms are monotone and well-defined for an infinite time horizon, and satisfy* (15) *and* (16)*. Then each agent $t \geq n_0$ chooses principal* 1 *with maximal possible probability $f_{\texttt{resp}}(1) = 1 - \epsilon_0$.*

[6] For random variables X, Y on $\mathbb{R}$, we say that X *stochastically dominates* Y if $\Pr[X \geq x] \geq \Pr[Y \geq x]$ for any $x \in \mathbb{R}$.

Proof. Consider global round $t \geq n_0$. Recall that each agent chooses principal 1 with probability at least $f_{\texttt{resp}}(-1) > 0$.

Then $\mathbb{E}[n_1(t+1)] \geq 2\epsilon_0\, t$. By Chernoff Bounds (Theorem 1), we have that $n_1(t+1) \geq \epsilon_0 t$ holds with probability at least $1-q$, where $q = \exp(-\epsilon_0 t/12)$.

We need to prove that $\texttt{PMR}_1(t) - \texttt{PMR}_2(t) > 0$. For any m_1 and m_2, consider the quantity

$$\Delta(m_1, m_2) := \texttt{BIR}_2(m_2+1) - \texttt{BIR}_1(m_1+1).$$

Whenever $m_1 \geq \epsilon_0 t/2 - 1$ and $m_2 < t$, it holds that

$$\Delta(m_1, m_2) \geq \Delta(\epsilon_0 t/2, t) \geq \texttt{BIR}_2(t)/2.$$

The above inequalities follow, resp., from algorithms' monotonicity and (15). Now,

$$\begin{aligned}
\texttt{PMR}_1(t) - \texttt{PMR}_2(t) &= \mathop{\mathbb{E}}_{m_1\sim\mathcal{N}_{1,t},\, m_2\sim\mathcal{N}_{2,t}} [\Delta(m_1, m_2)] \\
&\geq -q + \mathop{\mathbb{E}}_{m_1\sim\mathcal{N}_{1,t},\, m_2\sim\mathcal{N}_{2,t}} [\Delta(m_1, m_2) \mid m_1 \geq \epsilon_0 t/2 - 1] \\
&\geq \texttt{BIR}_2(t)/2 - q \\
&> \texttt{BIR}_2(t)/4 > 0 \qquad \text{(by (16))}.
\end{aligned}$$

◀

▶ Remark. Many standard MAB algorithms in the literature are parameterized by the time horizon T. Regret bounds for such algorithms usually include a polylogarithmic dependence on T. In particular, a typical upper bound for BIR has the following form:

$$\texttt{BIR}(n \mid T) \leq \text{polylog}(T) \cdot n^{-\gamma} \quad \text{for some } \gamma \in (0, \tfrac{1}{2}]. \tag{17}$$

Here we write $\texttt{BIR}(n \mid T)$ to emphasize the dependence on T.

We generalize (15) to handle the dependence on T: there exists a number T_0 and a function $n_0(T) \in \text{polylog}(T)$ such that

$$(\forall T \geq T_0,\ n \geq n_0(T)) \quad \frac{\texttt{BIR}_1(\epsilon_0 n/2 \mid T)}{\texttt{BIR}_2(n \mid T)} < \frac{1}{2}. \tag{18}$$

If this holds, we say that $\texttt{alg}_1$ *BIR-dominates* $\texttt{alg}_2$.

We provide a version of Theorem 8 in which algorithms are parameterized with time horizon T and condition (15) is replaced with (18); its proof is very similar and is omitted.

To state a game-theoretic corollary of Theorem 8, we consider a version of the competition game between the two principals in which they can only choose from a finite set $\mathcal{A}$ of monotone MAB algorithms. One of these algorithms is "better" than all others; we call it the *special* algorithm. Unless specified otherwise, it BIR-dominates all other allowed algorithms. The other algorithms satisfy (16). We call this game the *restricted competition game*.

▶ **Corollary 9.** *Assume* HardMax&Random *response function. Consider the restricted competition game with special algorithm* alg. *Then, for any sufficiently large time horizon T, this game has a unique Nash equilibrium: both principals choose* alg.

5.1 A little greedy goes a long way

Given any monotone MAB algorithm other than DynamicGreedy, we design a modified algorithm which learns at a slower rate, yet "wins the game" in the sense of Theorem 8. As a corollary, the competition game with unrestricted choice of algorithms typically does not have a Nash equilibrium.

Given an algorithm $\texttt{alg}_1$ that deviates from DynamicGreedy starting from step n_0 and a "mixing" parameter p, we will construct a modified algorithm as follows.

1. The modified algorithm coincides with $\texttt{alg}_1$ (and `DynamicGreedy`) for the first $n_0 - 1$ steps;
2. In each step $n \geq n_0$, $\texttt{alg}_1$ is invoked with probability $1 - p$, and with the remaining probability p does the "greedy choice": chooses an action with the largest posterior mean reward given the current information collected by $\texttt{alg}_1$.

For a cleaner comparison between the two algorithms, the modified algorithm does not record rewards received in steps with the "greedy choice". Parameter $p > 0$ is the same for all steps.

▶ **Theorem 10.** *Assume symmetric* `HardMax&Random` *response function. Let* $\epsilon_0 = f_{\texttt{resp}}(-1)$ *be the baseline probability. Suppose* $\texttt{alg}_1$ *deviates from* `DynamicGreedy` *starting from some step* n_0*. Let* $\texttt{alg}_2$ *be the modified algorithm, as described above, with mixing parameter* p *such that* $(1 - \epsilon_0)(1 - p) > \epsilon_0$*. Then each agent* $t \geq n_0$ *chooses principal* 2 *with maximal possible probability* $1 - \epsilon_0$*.*

▶ **Corollary 11.** *Suppose that both principals can choose any monotone MAB algorithm, and assume the symmetric* `HardMax&Random` *response function. Then for any time horizon* T*, the only possible* pure *Nash equilibrium is one where both principals choose* `DynamicGreedy`*. Moreover, no pure Nash equilibrium exists when some algorithm "dominates"* `DynamicGreedy` *in the sense of* (18) *and the time horizon* T *is sufficiently large.*

▶ Remark. The modified algorithm performs exploration at a slower rate. Let us argue how this may translate into a larger `BIR` compared to the original algorithm. Let $\texttt{BIR}'_1(n)$ be the `BIR` of the "greedy choice" fter after $n - 1$ steps of $\texttt{alg}_1$. Then

$$\texttt{BIR}_2(n) = \mathop{\mathbb{E}}_{m \sim (n_0-1)+\texttt{Binomial}(n-n_0+1,1-p)} \left[(1-p) \cdot \texttt{BIR}_1(m) + p \cdot \texttt{BIR}'_1(m) \right]. \tag{19}$$

In this expression, m is the number of times $\texttt{alg}_1$ is invoked in the first n steps of the modified algorithm. Note that $\mathbb{E}[m] = n_0 - 1 + (n - n_0 + 1)(1 - p) \geq (1 - p)n$.

Suppose $\texttt{BIR}_1(n) = \beta n^{-\gamma}$ for some constants $\beta, \gamma > 0$. Further, assume $\texttt{BIR}'_1(n) \geq c\, \texttt{BIR}_1(n)$, for some $c > 1 - \gamma$. Then for all $n \geq n_0$ and small enough $p > 0$ it holds that:

$$\begin{aligned}
\texttt{BIR}_2(n) &\geq (1 - p + pc)\ \mathbb{E}[\ \texttt{BIR}_1(m)\ ] \\
\mathbb{E}[\ \texttt{BIR}_1(m)\ ] &\geq \texttt{BIR}_1(\ \mathbb{E}[m]\ ) && \text{(By Jensen's inequality)} \\
&\geq \texttt{BIR}_1(\ (1-p)n\ ) && \text{(since } \mathbb{E}[m] \geq n(1-p)\text{)} \\
&\geq \beta \cdot n^{-\gamma} \cdot (1-p)^{-\gamma} && \text{(plugging in } \texttt{BIR}_1(n) = \beta n^{-\gamma}\text{)} \\
&> \texttt{BIR}_1(n)\ \ (1 - p\gamma)^{-1} && \text{(since } (1-p)^{\gamma} < 1 - p\gamma\text{).} \\
\texttt{BIR}_2(n) &> \alpha \cdot \texttt{BIR}_1(n), && \text{where} \quad \alpha = \tfrac{1-p+pc}{1-p\gamma} > 1.
\end{aligned}$$

(In the above equations, all expectations are over m distributed as in (19).)

Proof of Theorem 10. Let $\texttt{rew}'_1(n)$ denote the Bayesian-expected reward of the "greedy choice" after after $n - 1$ steps of $\texttt{alg}_1$. Note that $\texttt{rew}_1(\cdot)$ and $\texttt{rew}'_1(\cdot)$ are non-decreasing: the former because $\texttt{alg}_1$ is monotone and the latter because the "greedy choice" is only improved with an increasing set of observations. Therefore, the modified algorithm $\texttt{alg}_2$ is monotone by (19).

By definition of the "greedy choice," $\texttt{rew}_1(n) \leq \texttt{rew}'_1(n)$ for all steps n. Moreover, by Lemma 4, $\texttt{alg}_1$ has a strictly smaller $\texttt{rew}(n_0)$ compared to `DynamicGreedy`; so, $\texttt{rew}_1(n_0) < \texttt{rew}_2(n_0)$.

Let `alg` denote a copy of $\texttt{alg}_1$ that is running "inside" the modified algorithm $\texttt{alg}_2$. Let $m_2(t)$ be the number of global rounds before t in which the agent chooses principal 2 *and*

alg is invoked; in other words, it is the number of agents seen by alg before global round t. Let $\mathcal{M}_{2,t}$ be the agents' posterior distribution for $m_2(t)$.

We claim that in each global round $t \geq n_0$, distribution $\mathcal{M}_{2,t}$ stochastically dominates distribution $\mathcal{N}_{1,t}$, and $\texttt{PMR}_1(t) < \texttt{PMR}_2(t)$. We use induction on t. The base case $t = n_0$ holds because $\mathcal{M}_{2,t} = \mathcal{N}_{1,t}$ (because the two algorithms coincide on the first $n_0 - 1$ steps), and $\texttt{PMR}_1(n_0) < \texttt{PMR}_2(n_0)$ is proved as in (9), using the fact that $\texttt{rew}_1(n_0) < \texttt{rew}_2(n_0)$.

The induction step is proved as follows. The induction hypothesis for global round $t-1$ implies that agent $t-1$ is seen by alg with probability $(1-\epsilon_0)(1-p)$, which is strictly larger than ϵ_0, the probability with which this agent is seen by $\texttt{alg}_2$. Therefore, $\mathcal{M}_{2,t}$ stochastically dominates $\mathcal{N}_{1,t}$.

$$\begin{aligned}
\texttt{PMR}_1(t) &= \mathop{\mathbb{E}}_{n\sim\mathcal{N}_{1,t}} [\texttt{rew}_1(n+1)] \\
&\leq \mathop{\mathbb{E}}_{m\sim\mathcal{M}_{2,t}} [\texttt{rew}_1(m+1)] && (20)\\
&< \mathop{\mathbb{E}}_{m\sim\mathcal{M}_{2,t}} [(1-p)\cdot\texttt{rew}_1(m+1) + p\cdot\texttt{rew}'_1(m+1)] && (21)\\
&= \texttt{PMR}_2(t).
\end{aligned}$$

Here inequality (20) holds because $\texttt{rew}_1(\cdot)$ is monotone and $\mathcal{M}_{2,t}$ stochastically dominates $\mathcal{N}_{1,t}$, and inequality (21) holds because $\texttt{rew}_1(n_0) < \texttt{rew}_2(n_0)$ and $\mathcal{M}_{2,t}(n_0) > 0$.[7] ◀

6 SoftMax response function

This section is devoted to the SoftMax model. We recover a positive result under the assumptions from Theorem 8 (albeit with a weaker conclusion), and then proceed to a much more challenging result under weaker assumptions. We start with a formal definition:

▶ **Definition 12.** A response function $f_{\texttt{resp}}$ is SoftMax if the following conditions hold:

- $f_{\texttt{resp}}(\cdot)$ is bounded away from 0 and 1: $f_{\texttt{resp}}(\cdot) \in [\epsilon, 1-\epsilon]$ for some $\epsilon \in (0, \frac{1}{2})$,
- the response function $f_{\texttt{resp}}(\cdot)$ is "smooth" around 0:

$$\exists \text{constants } \delta_0, c_0, c'_0 > 0 \qquad \forall x \in [-\delta_0, \delta_0] \qquad c_0 \leq f'_{\texttt{resp}}(x) \leq c'_0. \tag{22}$$

- fair tie-breaking: $f_{\texttt{resp}}(0) = \frac{1}{2}$.

▶ Remark. This definition is fruitful when parameters c_0 and c'_0 are close to $\frac{1}{2}$. Throughout, we assume that $\texttt{alg}_1$ is better than $\texttt{alg}_2$, and obtain results parameterized by c_0. By symmetry, one could assume that $\texttt{alg}_2$ is better than $\texttt{alg}_1$, and obtain similar results parameterized by c'_0.

Our first result is a version of Theorem 8, with the same assumptions about the algorithms and essentially the same proof. The conclusion is much weaker: we can only guarantee that each agent $t \geq n_0$ chooses principal 1 with probability slightly larger than $\frac{1}{2}$. This is essentially unavoidable in a typical case when both algorithms satisfy $\texttt{BIR}(n) \to 0$, by Definition 12.

▶ **Theorem 13.** *Assume* SoftMax *response function. Suppose* $\texttt{alg}_1$ *has better* BIR *in the sense of* (15)*, and* $\texttt{alg}_2$ *satisfies the condition* (16)*. Then each agent* $t \geq n_0$ *chooses principal* 1 *with probability*

$$\Pr[i_t = 1] \geq \tfrac{1}{2} + \tfrac{c_0}{4}\,\texttt{BIR}_2(t). \tag{23}$$

[7] If $\texttt{rew}_1(\cdot)$ is strictly increasing, then inequality (20) is strict, too; this is because $\mathcal{M}_{2,t}(t-1) > \mathcal{N}_{1,t}(t-1)$.

Proof Sketch. We follow the steps in the proof of Theorem 8 to derive

$$\texttt{PMR}_1(t) - \texttt{PMR}_2(t) \geq \texttt{BIR}_2(t)/2 - q, \quad \text{where } q = \exp(-\epsilon_0 t/12).$$

This is at least $\texttt{BIR}_2(t)/4$ by (16). Then (23) follows by the smoothness condition (22). ◀

We recover a version of Corollary 9, if each principal's utility is the number of users (rather than the more general model in (5)). We also need a mild technical assumption that cumulative Bayesian regret (BReg) tends to infinity. BReg is a standard notion from the literature (along with BIR):

$$\texttt{BReg}(n) := n \cdot \mathop{\mathbb{E}}_{\mu\sim\mathcal{P}_{\texttt{mean}}}\left[\max_{a\in A} \mu_a\right] - \sum_{n=1}^{n} \texttt{rew}(n') = \sum_{n'=1}^{n} \texttt{BIR}(n'). \tag{24}$$

▶ **Corollary 14.** *Assume that the response function is* SoftMax*, and each principal's utility is the number of users. Consider the restricted competition game with special algorithm* alg*, and assume that all other allowed algorithms satisfy* $\texttt{BReg}(n) \to \infty$*. Then, for any sufficiently large time horizon T, this game has a unique Nash equilibrium: both principals choose* alg*.*

Further, we prove a much more challenging result in which the condition (15) is replaced with a much weaker "BIR-dominance" condition. For clarity, we will again assume that both algorithms are well-defined for an infinite time horizon. The *weak BIR dominance* condition says there exist constants $\beta_0, \alpha_0 \in (0, 1/2)$ and n_0 such that

$$(\forall n \geq n_0) \quad \frac{\texttt{BIR}_1((1-\beta_0)\, n)}{\texttt{BIR}_2(n)} < 1 - \alpha_0. \tag{25}$$

If this holds, we say that $\texttt{alg}_1$ *weakly BIR-dominates* $\texttt{alg}_2$. Note that the condition (18) involves sufficiently small multiplicative factors (resp., $\epsilon_0/2$ and $\frac{1}{2}$), the new condition replaces them with factors that can be arbitrarily close to 1.

We make a mild assumption on $\texttt{alg}_1$ that its $\texttt{BIR}_1(n)$ tends to 0. Formally, for any $\epsilon > 0$, there exists some $n(\epsilon)$ such that

$$(\forall n \geq n(\epsilon)) \qquad \texttt{BIR}_1(n) \leq \epsilon. \tag{26}$$

We also require a slightly stronger version of the technical assumption (16): for some n_0,

$$(\forall n \geq n_0) \qquad \texttt{BIR}_2(n) \geq \frac{4}{\alpha_0} \exp\left(\frac{-\min\{\epsilon_0, 1/8\} n}{12}\right) \tag{27}$$

▶ **Theorem 15.** *Assume the* SoftMax *response function. Suppose* $\texttt{alg}_1$ *weakly-BIR-dominates* $\texttt{alg}_2$*,* $\texttt{alg}_1$ *satisfies* (26)*, and* $\texttt{alg}_2$ *satisfies* (27)*. Then there exists some t_0 such that each agent $t \geq t_0$ chooses principal* 1 *with probability*

$$\Pr[i_t = 1] \geq \tfrac{1}{2} + \tfrac{c_0 \alpha_0}{4}\, \texttt{BIR}_2(t). \tag{28}$$

The main idea behind our proof is that even though $\texttt{alg}_1$ may have a slower rate of learning in the beginning, it will gradually catch up and surpass $\texttt{alg}_2$. We will describe this process in two phases. In the first phase, $\texttt{alg}_1$ receives a random agent with probability at least $f_{\texttt{resp}}(-1) = \epsilon_0$ in each round. Since $\texttt{BIR}_1$ tends to 0, the difference in BIRs between the two algorithms is also diminishing. Due to the SoftMax response function, $\texttt{alg}_1$ attracts each agent with probability at least $1/2 - O(\beta_0)$ after a sufficient number of rounds. Then the game enters the second phase: both algorithms receive agents at a rate close to $\frac{1}{2}$, and

the fractions of agents received by both algorithms — $n_1(t)/t$ and $n_2(t)/t$ — also converge to $\frac{1}{2}$. At the end of the second phase and in each global round afterwards, the counts $n_1(t)$ and $n_2(t)$ satisfy the weak BIR-dominance condition, in the sense that they both are larger than n_0 and $n_1(t) \geq (1-\beta_0)\, n_2(t)$. At this point, $\texttt{alg}_1$ actually has smaller $\texttt{BIR}$, which reflected in the $\texttt{PMR}$s eventually. Accordingly, from then on $\texttt{alg}_1$ attracts agents at a rate slightly larger than $\frac{1}{2}$. We prove that the "bump" over $\frac{1}{2}$ is at least on the order of $\texttt{BIR}_2(t)$.

Proof of Theorem 15. Let $\beta_1 = \min\{c_0'\delta_0, \beta_0/20\}$ with δ_0 defined in (22). Recall each agent chooses $\texttt{alg}_1$ with probability at least $f_{\texttt{resp}}(-1) = \epsilon_0$. By condition (26) and (27), there exists some sufficiently large T_1 such that for any $t \geq T_1$, $\texttt{BIR}_1(\epsilon_0 T_1/2) \leq \beta_1/c_0'$ and $\texttt{BIR}_2(t) > e^{-\epsilon_0 t/12}$. Moreover, for any $t \geq T_1$, we know $\mathbb{E}[n_1(t+1)] \geq \epsilon_0\, t$, and by the Chernoff Bounds (Theorem 1), we have $n_1(t+1) \geq \epsilon_0 t/2$ holds with probability at least $1 - q_1(t)$ with $q_1(t) = \exp(-\epsilon_0 t/12) < \texttt{BIR}_2(t)$. It follows that for any $t \geq T_1$,

$$\begin{aligned}
\texttt{PMR}_2(t) - \texttt{PMR}_1(t) &= \mathop{\mathbb{E}}_{m_1\sim\mathcal{N}_{1,t},\, m_2\sim\mathcal{N}_{2,t}} [\texttt{BIR}_1(m_1+1) - \texttt{BIR}_2(m_2+1)] \\
&\leq q_1(t) + \mathop{\mathbb{E}}_{m_1\sim\mathcal{N}_{1,t}} [\texttt{BIR}_1(m_1+1) \mid m_1 \geq \epsilon_0 t/2 - 1] - \texttt{BIR}_2(t) \\
&\leq \texttt{BIR}_1(\epsilon_0 T_1/2) \leq \beta_1/c_0'
\end{aligned}$$

Since the response function $f_{\texttt{resp}}$ is c_0'-Lipschitz in the neighborhood of $[-\delta_0, \delta_0]$, each agent after round T_1 will choose $\texttt{alg}_1$ with probability at least

$$p_t \geq \frac{1}{2} - c_0' \left(\texttt{PMR}_2(t) - \texttt{PMR}_1(t)\right) \geq \frac{1}{2} - \beta_1.$$

Next, we will show that there exists a sufficiently large T_2 such that for any $t \geq T_1 + T_2$, with high probability $n_1(t) > \max\{n_0, (1-\beta_0)n_2(t)\}$, where n_0 is defined in (25). Fix any $t \geq T_1 + T_2$. Since each agent chooses $\texttt{alg}_1$ with probability at least $1/2 - \beta_1$, by Chernoff Bounds (Theorem 1) we have with probability at least $1 - q_2(t)$ that the number of agents that choose $\texttt{alg}_1$ is at least $\beta_0(1/2 - \beta_1)t/5$, where the function

$$q_2(x) = \exp\left(\frac{-(1/2-\beta_1)(1-\beta_0/5)^2 x}{3}\right).$$

Note that the number of agents received by $\texttt{alg}_2$ is at most $T_1 + (1/2+\beta_1)t + (1/2-\beta_1)(1-\beta_0/5)t$.

Then as long as $T_2 \geq \frac{5T_1}{\beta_0}$, we can guarantee that $n_1(t) > n_2(t)(1-\beta_0)$ and $n_1(t) > n_0$ with probability at least $1 - q_2(t)$ for any $t \geq T_1 + T_2$. Note that the weak BIR-dominance condition in (25) implies that for any $t \geq T_1 + T_2$ with probability at least $1 - q_2(t)$,

$$\texttt{BIR}_1(n_1(t)) < (1-\alpha_0)\texttt{BIR}_2(n_2(t)).$$

It follows that for any $t \geq T_1 + T_2$,

$$\begin{aligned}
\texttt{PMR}_1(t) - \texttt{PMR}_2(t) &= \mathop{\mathbb{E}}_{m_1\sim\mathcal{N}_{1,t},\, m_2\sim\mathcal{N}_{2,t}} [\texttt{BIR}_2(m_2+1) - \texttt{BIR}_1(m_1+1)] \\
&\geq (1 - q_2(t))\alpha_0 \texttt{BIR}_2(t) - q_2(t) \\
&\geq \alpha_0 \texttt{BIR}_2(t)/4
\end{aligned}$$

where the last inequality holds as long as $q_2(t) \leq \alpha_0 \texttt{BIR}_2(t)/4$, and is implied by the condition in (27) as long as T_2 is sufficiently large. Hence, by the definition of our $\texttt{SoftMax}$ response function and assumption in (22), we have

$$\Pr[i_t = 1] \geq \frac{1}{2} + \frac{c_0 \alpha_0 \texttt{BIR}_2(t)}{4}.$$ ◀

Similar to the condition (15), we can also generalize the weak BIR-dominance condition (25) to handle the dependence on T: there exist some T_0, a function $n_0(T) \in \text{polylog}(T)$, and constants $\beta_0, \alpha_0 \in (0, 1/2)$, such that

$$(\forall T \geq T_0, n \geq n_0(T)) \quad \frac{\texttt{BIR}_1((1-\beta_0)\, n \mid T)}{\texttt{BIR}_2(n \mid T)} < 1 - \alpha_0. \tag{29}$$

We also provide a version of Theorem 13 under this more general weak BIR-dominance condition; its proof is very similar and is omitted. The following is just a direct consequence of Theorem 13 with this general condition.

▶ **Corollary 16.** *Assume that the response function is* `SoftMax`*, and each principal's utility is the number of users. Consider the restricted competition game in which the special algorithm* `alg` *weakly-BIR-dominates the other allowed algorithms, and the latter satisfy* $\texttt{BReg}(n) \to \infty$*. Then, for any sufficiently large time horizon T, there is a unique Nash equilibrium: both principals choose* `alg`*.*

7 Economic implications

aseditWe frame our contributions in terms of the relationship between *competitiveness* and *rationality* on one side, and adoption of better algorithms on the other. Recall that both *competitiveness* (of the game between the two principals) and *rationality* (of the agents) are controlled by the response function $f_{\texttt{resp}}$.

Main story. Our main story concerns the restricted competition game between the two principals where one allowed algorithm `alg` is "better" than the others. We track whether and when `alg` is chosen in an equilibrium. We vary *competitiveness/rationality* by changing the response function from `HardMax` (full rationality, very competitive environment) to `HardMax`&`Random` to `SoftMax` (less rationality and competition). Our conclusions are as follows:

- Under `HardMax`, no innovation: `DynamicGreedy` is chosen over `alg`.
- Under `HardMax`&`Random`, some innovation: `alg` is chosen as long as it BIR-dominates.
- Under `SoftMax`, more innovation: `alg` is chosen as long as it weakly-BIR-dominates.[8]

These conclusions follow, respectively, from Corollaries 3, 9 and 14. Further, we consider the uniform choice between the principals. It corresponds to the least amount of rationality and competition, and (when principals' utility is the number of agents) uniform choice provides no incentives to innovate.[9] Thus, we have an inverted-U relationship, see Figure 3.

Secondary story. Let us zoom in on the symmetric `HardMax`&`Random` model. Competitiveness and rationality within this model are controlled by the baseline probability $\epsilon_0 = f_{\texttt{resp}}(-1)$, which goes smoothly between the two extremes of `HardMax` ($\epsilon_0 = 0$) and the uniform choice ($\epsilon_0 = \frac{1}{2}$). Smaller ϵ_0 corresponds to increased rationality and increased competitiveness. For clarity, we assume that principal's utility is the number of agents.

[8] This is a weaker condition, the better algorithm is chosen in a broader range of scenarios.

[9] On the other hand, if principals' utility is somewhat aligned with agents' welfare, as in (5), then a monopolist principal is incentivized to choose the best possible MAB algorithm (namely, to minimize cumulative Bayesian regret $\texttt{BReg}(T)$). Accordingly, monopoly would result in better social welfare than competition, as the latter is likely to split the market and cause each principal to learn more slowly. This is a very generic and well-known effect regarding economies of scale.

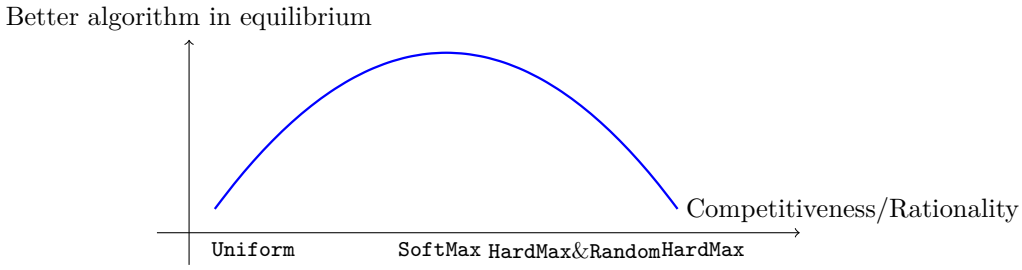


Figure 3 The stylized inverted-U relationship in the "main story".

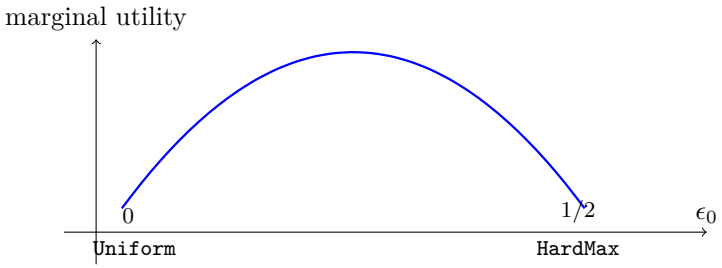


Figure 4 The stylized inverted-U relationship from the "secondary story"

We consider the marginal utility of switching to a better algorithm. Suppose initially both principals use some algorithm `alg`, and principal 1 ponders switching to another algorithm `alg'` which BIR-dominates `alg`. We are interested in the marginal utility of this switch. Then:

- $\epsilon_0 = 0$ (`HardMax`): the marginal utility can be negative if `alg` is `DynamicGreedy`.
- ϵ_0 near 0: only a small marginal utility can be guaranteed, as it may take a long time for `alg'` to "catch up" with `alg`, and hence less time to reap the benefits.
- "medium-range" ϵ_0: large marginal utility, as `alg'` learns fast and gets most agents.
- ϵ_0 near $\frac{1}{2}$: small marginal utility, as principal 1 gets most agents for free no matter what.

The familiar inverted-U shape is depicted in Figure 4.

Acknowledgements. The authors would like to thank Glen Weyl for discussions of related work in economics.

References

1 Alekh Agarwal, Sarah Bird, Markus Cozowicz, Miro Dudik, John Langford, Lihong Li, Luong Hoang, Dan Melamed, Siddhartha Sen, Robert Schapire, and Alex Slivkins. Multiworld testing: A system for experimentation, learning, and decision-making, 2016. A white paper, available at `https://github.com/Microsoft/mwt-ds/raw/master/images/MWT-WhitePaper.pdf`.

2 Philippe Aghion, Nicholas Bloom, Richard Blundell, Rachel Griffith, and Peter Howitt. Competition and innovation: An inverted u relationship. *Quaterly J. of Economics*, 120(2):701–728, 2005.

3 Susan Athey and Ilya Segal. An efficient dynamic mechanism. *Econometrica*, 81(6):2463–2485, 2013. A preliminary version has been available as a working paper since 2007.

4 Peter Auer, Nicolò Cesa-Bianchi, and Paul Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine Learning*, 47(2-3):235–256, 2002.

5 Peter Auer, Nicolò Cesa-Bianchi, Yoav Freund, and Robert E. Schapire. The nonstochastic multiarmed bandit problem. *SIAM J. Comput.*, 32(1):48–77, 2002. Preliminary version in *36th IEEE FOCS*, 1995.

6 Eduardo Azevedo and Daniel Gottlieb. Perfect competition in markets with adverse selection. *Econometrica*, 85(1):67–105, 2017.

7 Moshe Babaioff, Robert Kleinberg, and Aleksandrs Slivkins. Truthful mechanisms with implicit payment computation. *J. of the ACM*, 62(2):10, 2015. Subsumes the conference papers in *ACM EC 2010* and *ACM EC 2013*.

8 Moshe Babaioff, Yogeshwer Sharma, and Aleksandrs Slivkins. Characterizing truthful multi-armed bandit mechanisms. *SIAM J. on Computing (SICOMP)*, 43(1):194–230, 2014. Preliminary version in *10th ACM EC*, 2009.

9 Gal Bahar, Rann Smorodinsky, and Moshe Tennenholtz. Economic recommendation systems. In *16th ACM Conf. on Electronic Commerce (EC)*, 2016.

10 Dirk Bergemann and Juuso Välimäki. The dynamic pivot mechanism. *Econometrica*, 78(2):771–789, 2010. Preliminary versions have been available since 2006, as *Cowles Foundation Discussion Papers* #1584 (2006), #1616 (2007) and #1672(2008).

11 Kostas Bimpikis, Yiangos Papanastasiou, and Nicos Savva. Crowdsourcing exploration. *Management Science*, 2017. Forthcoming.

12 Patrick Bolton and Christopher Harris. Strategic Experimentation. *Econometrica*, 67(2):349–374, 1999.

13 Sébastien Bubeck and Nicolo Cesa-Bianchi. Regret Analysis of Stochastic and Nonstochastic Multi-armed Bandit Problems. *Foundations and Trends in Machine Learning*, 5(1), 2012.

14 Yeon-Koo Che and Johannes Hörner. Optimal design for social learning. Preprint, 2015. First draft: 2013.

15 Nikhil Devanur and Sham M. Kakade. The price of truthfulness for pay-per-click auctions. In *10th ACM Conf. on Electronic Commerce (EC)*, pages 99–106, 2009.

16 Eyal Even-Dar, Shie Mannor, and Yishay Mansour. Action elimination and stopping conditions for the multi-armed bandit and reinforcement learning problems. *J. of Machine Learning Research (JMLR)*, 7:1079–1105, 2006.

17 Peter Frazier, David Kempe, Jon M. Kleinberg, and Robert Kleinberg. Incentivizing exploration. In *ACM Conf. on Economics and Computation (ACM EC)*, pages 5–22, 2014.

18 Xavier Gabaix, David Laibson, Deyuan Li, Hongyi Li, Sidney Resnick, and Casper G. de Vries. The impact of competition on prices with numerous firms. *J. of Economic Theory*, 165:1–24, 2016.

19 Arpita Ghosh and Patrick Hummel. Learning and incentives in user-generated content: multi-armed bandits with endogenous arms. In *Innovations in Theoretical Computer Science Conf. (ITCS)*, pages 233–246, 2013.

20 John Gittins, Kevin Glazebrook, and Richard Weber. *Multi-Armed Bandit Allocation Indices*. John Wiley & Sons, 2011.

21 Ramakrishna Gummadi, Ramesh Johari, and Jia Yuan Yu. Mean field equilibria of multiarmed bandit games. In *13th ACM Conf. on Electronic Commerce (EC)*, 2012.

22 Chien-Ju Ho, Aleksandrs Slivkins, and Jennifer Wortman Vaughan. Adaptive contract design for crowdsourcing markets: Bandit algorithms for repeated principal-agent problems. *J. of Artificial Intelligence Research*, 55:317–359, 2016. Preliminary version appeared in *ACM EC 2014*.

23 Harold Hotelling. Stability in competition. *The Economic Journal*, 39(153):41–57, 1929.

24 Nicole Immorlica, Adam Tauman Kalai, Brendan Lucier, Ankur Moitra, Andrew Postlewaite, and Moshe Tennenholtz. Dueling algorithms. In *43rd ACM Symp. on Theory of Computing (STOC)*, pages 215–224, 2011.

25 Sham M. Kakade, Ilan Lobel, and Hamid Nazerzadeh. Optimal dynamic mechanism design and the virtual-pivot mechanism. *Operations Research*, 61(4):837–854, 2013.

26 Godfrey Keller, Sven Rady, and Martin Cripps. Strategic Experimentation with Exponential Bandits. *Econometrica*, 73(1):39–68, 2005.

27 Robert D. Kleinberg, Bo Waggoner, and E. Glen Weyl. Descending price optimally coordinates search. Working paper, 2016. Preliminary version in *ACM EC 2016*. Under submission to *Econometrica*.

28 Ilan Kremer, Yishay Mansour, and Motty Perry. Implementing the "wisdom of the crowd". *J. of Political Economy*, 122:988–1012, 2014. Preliminary version in *ACM EC 2014*.

29 Tze Leung Lai and Herbert Robbins. Asymptotically efficient Adaptive Allocation Rules. *Advances in Applied Mathematics*, 6:4–22, 1985.

30 Yishay Mansour, Aleksandrs Slivkins, and Vasilis Syrgkanis. Bayesian incentive-compatible bandit exploration. In *15th ACM Conf. on Electronic Commerce (EC)*, 2015.

31 Yishay Mansour, Aleksandrs Slivkins, Vasilis Syrgkanis, and Steven Wu. Bayesian exploration: Incentivizing exploration in bayesian games. Working paper, 2016. available at `https://arxiv.org/abs/1602.07570`. Preliminary version in *ACM EC 2016*.

32 Paul Milgrom and Nancy Stokey. Information, trade and common knowledge. *J. of Economic Theory*, 26(1):17–27, 1982.

33 Michael Mitzenmacher and Eli Upfal. *Probability and Computing: Randomized Algorithms and Probabilistic Analysis*. Cambridge University Press, 2005.

34 Jeffrey M. Perloff and Steven C. Salop. Equilibrium with product differentiation. *Review of Economic Studies*, LII:107–120, 1985.

35 Michael Rothschild and Joseph Stiglitz. Equilibrium in competitive insurance markets: An essay on the economics of imperfect information. *Quaterly J. of Economics*, 90(4):629–649, 1976.

36 Marc Rysman. The economics of two-sided markets. *J. of Economic Perspectives*, 23(3):125–144, 2009.

37 Joseph Schumpeter. *Capitalism, Socialism and Democracy*. Harper & Brothers, 1942.

38 Adish Singla and Andreas Krause. Truthful incentives in crowdsourcing tasks using regret minimization mechanisms. In *22nd Intl. World Wide Web Conf. (WWW)*, pages 1167–1178, 2013.

39 Aleksandrs Slivkins. Introduction to multi-armed bandits, 2017. A book draft, available at http://research.microsoft.com/en-us/people/slivkins.

40 Andre Veiga and Glen Weyl. Product design in selection markets. *Quarterly J. of Economics*, 131(2):1007–1056, 2016.

41 Xavier Vives. Innovation and competitive pressure. *J. of Industrial Economics*, 56(3), 2008.

42 Glen Weyl and Alexander White. Let the right 'one' win: Policy lessons from the new economics of platforms. *Competition Policy International*, 12(2):29–51, 2014.

43 Yisong Yue, Josef Broder, Robert Kleinberg, and Thorsten Joachims. The k-armed dueling bandits problem. *J. Comput. Syst. Sci.*, 78(5):1538–1556, 2012. Preliminary version in COLT 2009.

44 Yisong Yue and Thorsten Joachims. Interactively optimizing information retrieval systems as a dueling bandits problem. In *26th Intl. Conf. on Machine Learning (ICML)*, pages 1201–1208, 2009.

A Background on multi-armed bandits

This appendix provides some pertinent background on multi-armed bandits (*MAB*). We discuss BIR and monotonicity of several MAB algorithms, touching upon: DynamicGreedy and StaticGreedy (Section A.1), "naive" MAB algorithms that separate exploration and exploitation (Section A.2), and "smart" MAB algorithms that combine exploration and exploitation (Section A.3).

As we do throughout the paper, we focus on MAB with i.i.d. rewards and a Bayesian prior; we call it *Bayesian MAB* for brevity.

A.1 DynamicGreedy and StaticGreedy

We provide an example when DynamicGreedy and StaticGreedy have constant BIR, and prove monotonicity of DynamicGreedy. For the example, it suffices to consider *deterministic rewards* (for each action a, the realized reward is always equal to the mean μ_a) and *independent priors* (according to the prior $\mathcal{P}_{\texttt{mean}}$, random variables $\mu_1, \ldots, \mu_K$ are mutually independent) each of *full support*.

The following claim is immediate from the definition of the CDF function

▶ **Claim 17.** *Assume independent priors. Let F_i be the CDF of the mean reward μ_i of action $a_i \in A$. Then, for any numbers $z_2 > z_1 > \mathbb{E}[\mu_2]$ we have $\Pr[\mu_1 \leq z_1 \text{ and } \mu_2 \geq z_2] = F_1(z_1)(1 - F_2(z_2))$.*

We can now draw an immediate corollary of the above claim

▶ **Corollary 18.** *Consider any problem instance of Bayesian MAB with two actions and independent priors which are full support. Then:*

(a) *With constant probability,* StaticGreedy *has a constant* BIR *for all steps.*

(b) *Assuming deterministic rewards, with constant probability* DynamicGreedy *has a constant* BIR *for all steps.*

▶ Remark. A similar result holds for rewards which are distributed as Bernoulli random variables. In this case we consider accumulative reward of an action as a random walk, and use a high probability variation of the law of iterated logarithms. (Details omitted.)

Next, we show that DynamicGreedy is monotone.

▶ **Lemma 19.** DynamicGreedy *is monotone, in the sense that* $\texttt{rew}(n)$ *is non-decreasing. Further,* $\texttt{rew}(n)$ *is strictly increasing for every time step n with $\Pr[a_n \neq a_{n+1}] > 0$.*

Proof. We prove by induction on n that $\texttt{rew}(n) \leq \texttt{rew}(n+1)$ for DynamicGreedy. Let a_n be the random variable recommended at time t, then $\mathbb{E}[\mu_{a_n} | \mathcal{I}_n] = \texttt{rew}(n)$. We can rewrite this as:

$$\texttt{rew}(n) = \mathop{\mathbb{E}}_{\mathcal{I}_n}[\mathop{\mathbb{E}}_{r_n}[\mu_{a_n} | r_n, \mathcal{I}_n]] = \mathop{\mathbb{E}}_{\mathcal{I}_{n+1}}[\mu_{a_n} | \mathcal{I}_{n+1}]$$

since $\mathcal{I}_{n+1} = (\mathcal{I}_n, r_n)$. At time $n+1$ DynamicGreedy will select an action a_{n+1} such that:

$$\texttt{rew}(n+1) = \mathbb{E}[\mu_{a_{n+1}} | \mathcal{I}_{n+1}] \geq \mathbb{E}[\mu_{a_n} | \mathcal{I}_n] = \texttt{rew}(n)$$

which proves the monotonicity. In cases that $\Pr[a_n \neq a_{n+1}] > 0]$ we have a strict inequality, since with some probability we select a better action then the realization of a_n. ◀

A.2 "Naive" MAB algorithms that separate exploration and exploitation

MAB algorithm `ExplorExploit` (m) initially explores each action with m agents and for the remaining $T - |A|m$ agents recommends the action with the highest observed average. In the explore phase it assigns a random permutation of the mK recommendations.

▶ **Lemma 20.** *The* `ExplorExploit` $(T^{2/3} \log |A|/\delta)$ *algorithm has, with probability* $1 - \delta$*, for any* $n \geq |A|T^{2/3}$ *we have* `BIR` $(n) = O(T^{-1/3})$*. In addition,* `ExplorExploit` (m) *is monotone.*

Proof. In the explore phase we we approximate for each action $a \in A$, the value of μ_a by $\hat{\mu}_a$. Using the standard Chernoff bounds we have that with probability $1 - \delta$, for every action $a \in A$ we have $|\mu_a - \hat{\mu}_a| \leq T^{-1/3}$.

Let $a^* = \arg\max_a \mu_a$ and a^{ee} the action that `ExplorExploit` selects in the explore phase after the first $|A|T^{2/3}$ agents. Since $\hat{\mu}_{a^*} \leq \hat{\mu}_{a^{ee}}$, this implies that $\mu_{a^*} - \mu_{a^{ee}} = O(T^{-1/3})$.

To show that `ExplorExploit` (m) is monotone, we need to show only that $\texttt{rew}(mK) \leq \texttt{rew}(mK + 1)$. This follows since for any $t < mK$ we have $\texttt{rew}(t) = \texttt{rew}(t + 1)$, since the recommended action is uniformly distributed for each time t. Also, for any $t \geq mK + 1$ we have $\texttt{rew}(t) = \texttt{rew}(t + 1)$ since we are recommending the same exploration action. The proof that $\texttt{rew}(mK) \leq \texttt{rew}(mK + 1)$ is the same as for `DynamicGreedy` in Lemma 19. ◀

We can also have a a phased version which we call `PhasedExplorExploit` (m_t), where time is partition in to phases. In phase t we have m_t agents and a random subset of K explore the actions (each action explored by a single agent) and the other agents exploit. (This implies that we need that $m_t \geq K$ for all t. We also assume that m_t is monotone in t.)

▶ **Lemma 21.** *Consider the case that* $K = 2$ *and the rewards of the actions are Bernoulli r.v. with parameter* μ_i *and* $\Delta = \mu_1 - \mu_2$*. Algorithm* `PhasedExplorExploit` (m_t) *is monotone and for* $m_t = \sqrt{t}$ *it has* $\texttt{BIR}(n) = O(n^{-1/3} + e^{-O(\Delta^2 n^{2/3})})$*.*

Proof. We first show that it is monotone. Recall that $\mu_1 > \mu_2$. Let $S_i = \sum_{j=1}^{t} r_{i,j}$ be the sum of the rewards of action i up to phase t. We need to show that $\Pr[S_1 > S_2] + (1/2)\Pr[S_1 = S_2]$ is monotonically increasing in t. Consider the random variable $Z = S_1 - S_2$. At each phase it increases by $+1$ with probability $\mu_1(1 - \mu_2)$, decreases by -1 with probability $(1 - \mu_1)\mu_2$ and otherwise does not change.

Consider the values of Z up to phase t. We really care only about the probability that is shifted from positive to negative and vice versa.

First, consider the probability that $Z = 0$. We can partition it to $S_1 = S_2 = r$ events, and let $p(r, r)$ be the probability of this event. For each such event, we have $p(r, r)\mu_1$ moved to $Z = +1$ and $p(r, r)\mu_2$ moved to $Z = -1$. Since $\mu_1 > \mu_2$ we have that $p(r, r)\mu_1 \geq p(r, r)\mu_2$ (note that $p(r, r)$ might be zero, so we do not have a strict inequality).

Second, consider the probability that $Z = +1$ or $Z = -1$. We can partition it to $S_1 = r + 1; S_2 = r$ and $S_1 = r; S_2 = r + 1$ events, and let $p(r + 1, r)$ and $p(r, r + 1)$ be the probabilities of those events. It is not hard to see that $p(r + 1, r)\mu_2 = p(r, r + 1)\mu_1$. This implies that the probability mass moved from $Z = +1$ to $Z = 0$ is identical to that moved from $Z = -1$ to $Z = 0$.

We have showed that $\Pr[S_1 > S_2] + (1/2)\Pr[S_1 = S_2]$ and therefore the expected valued of the exploit action is non-decreasing. Since we have that the size of the phases are increasing, the `BIR` is strictly increasing between phases and identical within each phase.

We now analyze the BIR regret. Note that agent n is in phase $O(n^{2/3})$ and the length of his phase is $O(n^{1/3})$. The BIR has two parts. The first is due to the exploration, which is at most $O(n^{-1/3})$. The second is due to the probability that we exploit the wrong action. This happens with probability $\Pr[S_1 < S_2] + (1/2)\Pr[S_1 = S_2]$ which we can bound using a Chernoff bound by $e^{-O(\Delta^2 n^{2/3})}$, since we explored each action $O(n^{2/3})$ times. ◀

▶ Remark. Actually we have a tradeoff depending on the parameter m_t between the regret due to exploration and exploitation. (Note that the monotonicity is always guarantee assuming m_t is monotone.) If we can set that $m_t = 2^t$ then at time n we have $2/n$ probability of an exploit action. For the explore action we are in phase $\log n$ so the probability of a sub-optimal explore action is $n^{-O(\Delta^{-2})}$. This should give us $\texttt{BIR}(n) = O(n^{-O(\Delta^{-2})})$.

A.3 "Smart" MAB algorithms that combine exploration and exploitation

MAB algorithm SuccesiveEliminationReset works as follows. It keeps a set of surviving actions $A_s \subseteq A$, where initially $A_s = A$. The agents are partition into phases, where each phase is a random permutation of the non-eliminated actions. Let $\hat{\mu}_{i,t}$ be the average of the rewards of action i up to phase t and $\hat{\mu}^* = \max_i \hat{\mu}_{i,t}$. We eliminate action i at the end of phase t, i.e., delete it from A_s, if $\hat{\mu}^*_t - \hat{\mu}_{i,t} > \log(T/\delta)/\sqrt{t}$. In SuccesiveEliminationReset we simply reset the algorithm with $A = A_s - A_{e,t}$, where $A_{e,t}$ is the set of eliminated actions after phase t. Namely, we restart $\hat{\mu}_{i,t}$ and ignore the old rewards before the elimination.

▶ **Lemma 22.** *The algorithm* SuccesiveEliminationReset*, has, with probability* $1-\delta$*,* BIR $(n) = O(\log(T/\delta)/\sqrt{n/K})$.

Proof. Let the best action be $a^* = \arg\max_a \mu_a$. With probability $1-\delta$ at any time n we have that for any action $i \in A_s$ that $|\hat{\mu}_i - \mu_i| \leq \log(T/\delta)/\sqrt{n/K}$, and $a^* \in A_s$. This implies that any action a such $\mu_{a^*} - \mu_a > 3\log(T/\delta)/\sqrt{n/K}$ is eliminated. Therefore, any action in A_s has BIR (n) of at most $6\log(T/\delta)/\sqrt{n/K}$. ◀

▶ **Lemma 23.** *Assume that if* $\mu_i \geq \mu_j$ *then the rewards* r_i *stochastically dominates the rewards* r_j*. Then,* SuccesiveEliminationReset *is monotone*

Proof. Consider the first time T an action is eliminated, and let $T = \tau$ be a realized value of T. Then, clearly for $n < \tau$ we have $\texttt{rew}(n) = \texttt{rew}(1)$.

Consider two actions $a_1, a_2 \in A$, such that $\mu_{a_1} \geq \mu_{a_2}$. At time $T = \tau$, the probability that a_1 is eliminated is smaller than the probability that a_2 is eliminated. This follows since $\hat{\mu}_{a_1}$ stochastically dominates $\hat{\mu}_{a_2}$, which implies that for any threshold θ we have $\Pr[\hat{\mu}_{a_1} \geq \theta] \geq \Pr[\hat{\mu}_{a_2} \geq \theta]$.

After the elimination we consider the expected reward of the eliminated action $\sum_{i \in A} \mu_i q_i$, where q_i is the probability that action i was eliminated in time $T = \tau$. We have that $q_i \leq q_{i+1}$, from the probabilities of elimination.

The sum $\sum_{i \in A} \mu_i q_i$ with $q_i \leq q_{i+1}$ and $\sum_i q_i = 1$ is maximized by setting $q_i = 1/|A|$. (We can see that if there are $q_i \neq 1/|A|$, then there are two $q_i < q_{i+1}$, and one can see that setting both to $(q_i + q_{i+1})/2$ increases the value.) Therefore we have that the $\texttt{rew}(\tau) \geq \texttt{rew}(\tau - 1)$.

Now we can continue by induction. For the induction, we can show the property for *any* remaining set of at most $k-1$ actions. The main issue is that SuccesiveEliminationReset restarts from scratch, so we can use induction. ◀

B Non-degeneracy via a random perturbation

We show that Assumption (4) holds almost surely under a small random perturbation of the prior. We focus on problem instances with 0-1 rewards, and assume that the prior $\mathcal{P}_{\texttt{mean}}$ is independent across arms and has a finite support.[10] Consider the probability vector in the prior for arm a:

$$\vec{p}_a = (\ \Pr[\mu_a = \nu] :\ \nu \in \texttt{support}(\mu_a)\ ).$$

We apply a small random perturbation independently to each such vector:

$$\vec{p}_a \leftarrow \vec{p}_a + \vec{q}_a, \quad \text{where} \quad \vec{q}_a \sim \mathcal{N}_a. \tag{30}$$

Here $\mathcal{N}_a$ is the noise distribution for arm a: a distribution over real-valued, zero-sum vectors of dimension $d_a = |\texttt{support}(\mu_a)|$. We need the noise distribution to satisfy the following property:

$$\forall x \in [-1,1]^{d_a} \setminus \{0\} \qquad \Pr_{q\sim\mathcal{N}_a}[x\cdot(\vec{p}_a+q)\neq 0] = 1. \tag{31}$$

▶ **Theorem 24.** *Consider an instance of MAB with 0-1 rewards. Assume that the prior $\mathcal{P}_{\texttt{mean}}$ is independent across arms, and each mean reward μ_a has a finite support that does not include 0 or 1. Assume that noise distributions $\mathcal{N}_a$ satisfy property (31). If random perturbation (30) is applied independently to each arm a, then Eq. 4 holds almost surely for each history h.*

▶ Remark. As a generic example of a noise distribution which satisfies Property (31), consider the uniform distribution $\mathcal{N}$ over the bounded convex set

$$Q = \left\{q \in \mathbb{R}^{d_a} \mid q\cdot\vec{1} = 0 \text{ and } \|q\|_2 \le \epsilon\right\},$$

where $\vec{1}$ denotes the all-1 vector. If $x = a\vec{1}$ for some non-zero value of a, then (31) holds because

$$x\cdot(p+q) = x\cdot p = a \neq 0.$$

Otherwise, denote $p = \vec{p}_a$ and observe that $x\cdot(p+q) = 0$ only if $x\cdot q = c \triangleq x\cdot(-p)$. Since $x \neq \vec{1}$, the intersection $Q \cap \{x\cdot q = c\}$ either is empty or has measure 0 in Q, which implies $\Pr_q[x\cdot(p+q)\neq 0] = 1$.

To prove Theorem 24, it suffices to focus on two arms, and perturb one of them. Since realized rewards have finite support, there are only finitely many possible histories. Therefore, it suffices to focus on a fixed history h.

▶ **Lemma 25.** *Consider an instance of MAB with 0-1 rewards. Assume that the prior $\mathcal{P}_{\texttt{mean}}$ is independent across arms, and that $\texttt{support}(\mu_1)$ is finite and does not include 0 or 1. Fix history h. Suppose random perturbation (30) is applied to arm 1, with noise distribution $\mathcal{N}_1$ that satisfies (31). Then $\mathbb{E}[\mu_1 \mid h] \neq \mathbb{E}[\mu_2 \mid h]$ almost surely.*

[10] The assumption of 0-1 rewards is for clarity. Our results hold under a more general assumption that for each arm a, rewards can only take finitely many values, and each of these values is possible (with positive probability) for every possible value of the mean reward μ_a.

Proof. Note that $\mathbb{E}[\mu_a \mid h]$ does not depend on the algorithm which produced this history. Therefore, for the sake of the analysis, we can assume w.l.o.g. that this history has been generated by a particular algorithm, as long as this algorithm can can produce this history with non-zero probability. Let us consider the algorithm that deterministically chooses same actions as h.

Let $S = \texttt{support}(\mu_1)$. Then:

$$\mathbb{E}[\mu_1 \mid h] = \sum_{\nu \in S} \nu \cdot \Pr[\mu_1 = \nu \mid h] = \sum_{\nu \in S} \nu \cdot \Pr[h \mid \mu_1 = \nu] \cdot \Pr[\mu_1 = \nu] \;/\; \Pr[h],$$
$$\Pr[h] = \sum_{\nu \in S} \Pr[h \mid \mu_1 = \nu] \cdot \Pr[\mu_1 = \nu].$$

Therefore, $\mathbb{E}[\mu_1 \mid h] = \mathbb{E}[\mu_2 \mid h]$ if and only if

$$\sum_{\nu \in S} (\nu - C) \cdot \Pr[h \mid \mu_1 = \nu] \cdot \Pr[\mu_1 = \nu] = 0, \qquad \text{where} \qquad C = \mathbb{E}[\mu_2 \mid h].$$

Since $\mathbb{E}[\mu_2 \mid h]$ and $\Pr[h \mid \mu_1 = \nu]$ do not depend on the probability vector $\vec{p}_1$, we conclude that

$$\mathbb{E}[\mu_1 \mid h] = \mathbb{E}[\mu_2 \mid h] \quad \Leftrightarrow \quad x \cdot \vec{p}_1 = 0,$$

where vector

$$x := (\ (\nu - C) \cdot \Pr[h \mid \mu_1 = \nu] :\ \nu \in S\ ) \in [-1, 1]^{d_1}$$

does not depend on $\vec{p}_1$.

Thus, it suffices to prove that $x \cdot \vec{p}_1 \neq 0$ almost surely under the perturbation. In a formula:

$$\Pr_{q \sim \mathcal{N}_1} [x \cdot (\vec{p}_1 + q) \neq 0] = 1 \tag{32}$$

Note that $\Pr[h \mid \mu_1 = \nu] > 0$ for all $\nu \in S$, because $0, 1 \notin S$. It follows that at most one coordinate of x can be zero. So (32) follows from property (31). ◀

Limits for Rumor Spreading in Stochastic Populations*†

Lucas Boczkowski[1], Ofer Feinerman[2], Amos Korman[3], and Emanuele Natale[4]

1 **CNRS, IRIF, Université Paris Diderot, 75013 Paris, France**
lukas.boczkowski@irif.fr

2 **Max-Planck-Institut für Informatik, 66123 Saarbrücken, Germany**
enatale@mpi-inf.mpg.de

3 **CNRS, IRIF, Université Paris Diderot, 75013 Paris, France**
amos.korman@irif.fr

4 **Weizmann Institute of Science, 76100 Rehovot, Israel**
feinermanofer@gmail.com

Abstract

Biological systems can share and collectively process information to yield emergent effects, despite inherent noise in communication. While man-made systems often employ intricate structural solutions to overcome noise, the structure of many biological systems is more amorphous. It is not well understood how communication noise may affect the computational repertoire of such groups. To approach this question we consider the basic collective task of rumor spreading, in which information from few knowledgeable sources must reliably flow into the rest of the population.

In order to study the effect of communication noise on the ability of groups that lack stable structures to efficiently solve this task, we consider a noisy version of the uniform $\mathcal{PULL}$ model. We prove a lower bound which implies that, in the presence of even moderate levels of noise that affect all facets of the communication, no scheme can significantly outperform the trivial one in which agents have to wait until directly interacting with the sources. Our results thus show an exponential separation between the uniform $\mathcal{PUSH}$ and $\mathcal{PULL}$ communication models in the presence of noise. Such separation may be interpreted as suggesting that, in order to achieve efficient rumor spreading, a system must exhibit either some degree of structural stability or, alternatively, some facet of the communication which is immune to noise.

We corroborate our theoretical findings with a new analysis of experimental data regarding recruitment in *Cataglyphis niger* desert ants.

1998 ACM Subject Classification F.1.2 Modes of Computation, F.2.m Miscellaneous

Keywords and phrases Noisy communication, Passive communication, Ant recruitment, Hypothesis testing

Digital Object Identifier 10.4230/LIPIcs.ITCS.2018.49

* This work has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No 648032).
† A full version of the paper is available at https://arxiv.org/abs/1712.08507.

1 Introduction

1.1 Background and motivation

Systems composed of tiny mobile components must function under conditions of unreliability. In particular, any sharing of information is inevitability subject to communication noise. The effects of communication noise in distributed living systems appears to be highly variable. While some systems disseminate information efficiently and reliably despite communication noise [2, 21, 11, 31, 37], others generally refrain from acquiring social information, consequently losing all its potential benefits [25, 35, 38]. It is not well understood which characteristics of a distributed system are crucial in facilitating noise reduction strategies and, conversely, in which systems such strategies are bound to fail. Progress in this direction may be valuable towards better understanding the constraints that govern the evolution of cooperative biological systems.

Computation under noise has been extensively studied in the computer science community. These studies suggest that different forms of error correction (*e.g.,* redundancy) are highly useful in maintaining reliability despite noise [3, 1, 40, 39]. All these, however, require the ability to transfer significant amount of information over stable communication channels. Similar redundancy methods may seem biologically plausible in systems that enjoy stable structures, such as brain tissues.

The impact of noise in stochastic systems with ephemeral connectivity patterns is far less understood. To study these, we focus on *rumor spreading* - a fundamental information dissemination task that is a prerequisite to almost any distributed system [10, 12, 16, 28]. A successful and efficient rumor spreading process is one in which a large group manages to quickly learn information initially held by one or a few informed individuals. Fast information flow to the whole group dictates that messages be relayed between individuals. Similar to the game of Chinese Whispers, this may potentially result in runaway buildup of noise and loss of any initial information [9]. It currently remains unclear what are the precise conditions that enable fast rumor spreading. On the one hand, recent works indicate that in some models of random noisy interactions, a collective coordinated process can in fact achieve fast information spreading [22, 23]. These models, however, are based on *push* operations that inherently include a certain reliable component (see more details in Section 1.3.2). On the other hand, other works consider computation through noisy operations, and show that several distributed tasks require significant running time [26]. The tasks considered in these works (including the problem of learning the input bits of all processors, or computing the parity of all the inputs) were motivated by computer applications, and may be less relevant for biological contexts. Moreover, they appear to be more demanding than basic tasks, such as rumor spreading, and hence it is unclear how to relate bounds on the former problems to the latter ones.

In this paper we take a general stance to identify limitations under which reliable and fast rumor spreading cannot be achieved. Modeling a well-mixed population, we consider a passive communication scheme in which information flow occurs as one agent observes the cues displayed by another. If these interactions are perfectly reliable, the population could achieve extremely fast rumor spreading [28]. In contrast, here we focus on the situation in which messages are noisy. Informally, our main theoretical result states that when all components of communication are noisy then fast rumor spreading through large populations is not feasible. In other words, our results imply that fast rumor spreading can only be achieved if either 1) the system exhibits some degree of structural stability or 2) some facet of the pairwise communication is immune to noise. In fact, our lower bounds hold even when individuals are granted unlimited computational power and even when the system can take advantage of complete synchronization.

Finally, we corroborate our theoretical findings with new analyses regarding the efficiency of information dissemination during recruitment by desert ants. More specifically, we analyze data from an experiment conducted at the Weizmann Institute of Science, concerning recruitment in *Cataglyphis niger* desert ants [34]. These analyses suggest that this biological system lacks reliability in all its communication components, and its deficient performances qualitatively validate our predictions. We stress that this part of the paper is highly uncommon. Indeed, using empirical biological data to validate predictions from theoretical distributed computing is extremely rare. We believe, however, that this interdisciplinary methodology may carry significant potential, and hope that this paper could be useful for future works that will follow this framework.

1.2 The problem

An intuitive description of the model follows. For more precise definitions, see Section 2.

Consider a population of n *agents*. Thought of as computing entities, assume that each agent has a discrete internal *state*, and can execute randomized algorithms - by internally flipping coins. In addition, each agent has an *opinion*, which we assume for simplicity to be binary, *i.e.,* either 0 or 1. A small number, s, of agents play the role of *sources*. Source agents are aware of their role and share the same opinion, referred to as the *correct opinion*. The goal of all agents is to have their opinion coincide with the correct opinion.

To achieve this goal, each agent continuously displays one of several *messages* taken from some finite alphabet Σ. Agents interact according to a random pattern, termed as the *parallel-$\mathcal{PULL}$* model: In each round $t \in \mathbb{N}^+$, each agent u observes the message currently displayed by another agent v, chosen uniformly at random (u.a.r) from all agents. Importantly, communication is noisy, hence the message observed by u may differ from that displayed by v. The noise is characterized by a *noise parameter* $\delta > 0$. Our model encapsulates a large family of noise distributions, making our bounds highly general. Specifically, the noise distribution can take *any* form, as long as it satisfies the following criterion.

▶ **Definition 1** (The δ-uniform noise criterion)**.** Any time some agent u observes an agent v holding some message $m \in \Sigma$, the probability that u actually receives a message m' is at least δ, for any $m' \in \Sigma$. All noisy samples are independent.

When messages are noiseless, it is easy to see that the number of rounds that are required to guarantee that all agents hold the correct opinion with high probability is $\mathcal{O}(\log n)$ [28]. In what follows, we aim to show that when the δ-uniform noise criterion is satisfied, the number of rounds required until even one non-source agent can be moderately certain about the value of the correct opinion is very large. Specifically, thinking of δ and s as constants independent of the population size n, this time is at least $\Omega(n)$.

To prove the lower bound, we will bestow the agents with capabilities that far surpass those that are reasonable for biological entities. These include:

- Unique identities: Agents have unique identities in the range $\{1, 2, \ldots n\}$. When observing agent v, its identity is received without noise.
- Complete knowledge of the system: Agents have access to all parameters of the system (including n, s, and δ) as well as to the full knowledge of the initial configuration except, of course, the correct opinion and the identity of the sources. In addition, agents have access to the results of random coin flips used internally by all other agents.
- Full synchronization: Agents know when the execution starts, and can count rounds.

We show that even given this extra computational power, fast convergence cannot be achieved.

1.3 Our contributions

1.3.1 Theoretical results

In all the statements that follow we consider the parallel-$\mathcal{PULL}$ model satisfying the δ-uniform noise criterion, where $cs/n < \delta \leq 1/2$ for some sufficiently large constant c. Note that our criterion given in Definition 1 implies that $\delta \leq 1/|\Sigma|$. Hence, the previous lower bound on δ implies a restriction on the alphabet size, specifically, $|\Sigma| \leq n/(cs)$.

▶ **Theorem 2.** *Any rumor spreading protocol cannot converge in less than* $\Omega(\frac{n\delta}{s^2(1-2\delta)^2})$ *rounds.*

Recall that a source is aware that it is a source, but if it wishes to identify itself as such to agents that observe it, it must encode this information in a message, which is, in turn, subject to noise. We also consider the case in which an agent can reliably identify a source when it observes one (i.e.,, this information is not noisy). For this case, the following bound, which is weaker than the previous one but still polynomial, apply (a formal proof appears in the full version of the paper):

▶ **Corollary 3.** *Assume that sources are reliably detectable. There is no rumor spreading protocol that converges in less than* $\Omega((\frac{n\delta}{s^2(1-2\delta)^2})^{1/3})$ *rounds.*

Our results suggest that, in contrast to systems that enjoy stable connectivity, structureless systems are highly sensitive to communication noise. More concretely, the two crucial assumptions that make our lower bounds work are: 1) stochastic interactions, and 2) δ-uniform noise (see the right column of Figure 1). When agents can stabilize their interactions the first assumption is violated. In such cases, agents can overcome noise by employing simple error-correction techniques, *e.g.,* using redundant messaging or waiting for acknowledgment before proceeding. As demonstrated in Figure 1 (left column), when the noise is not uniform, it might be possible to overcome it with simple techniques based on using default neutral messages, and employing exceptional distinguishable signals only when necessary.

1.3.2 Exponential separation between $\mathcal{PUSH}$ and $\mathcal{PULL}$

Our lower bounds on the parallel-$\mathcal{PULL}$ model (where agents observe other agents) should be contrasted with known results in the parallel-$\mathcal{PUSH}$ model, which is the push equivalent to parallel-$\mathcal{PULL}$ model, where in each round each agent may or may not actively push a message to another agent chosen u.a.r. (see also Section 2.3). Although never proved, and although their combination is known to achieve more power than each of them separately [28], researchers often view the parallel-$\mathcal{PULL}$ and parallel-$\mathcal{PUSH}$ models as very similar on complete communication topologies. Our lower bound result, however, undermines this belief, proving that in the context of noisy communication, there is an exponential separation between the two models. Indeed, when the noise level is constant for instance, convergence (and in fact, a much stronger convergence than we consider here) can be achieved in the parallel-$\mathcal{PUSH}$ using only logarithmic number of rounds [22, 23], by a simple strategy composed of two stages. The first stage consists of providing all agents with a guess about the source's opinion, in such a way that ensures a non-negligible bias toward the correct guess. The second stage then boosts this bias by progressively amplifying it. A crucial aspect in the first stage is that agents remain silent until a certain point in time that they start sending messages continuously, which happens after being contacted for the first time. This prevents agents from starting to spread information before they have sufficiently reliable knowledge. It further allows to control the dynamics of the information spread in a balanced

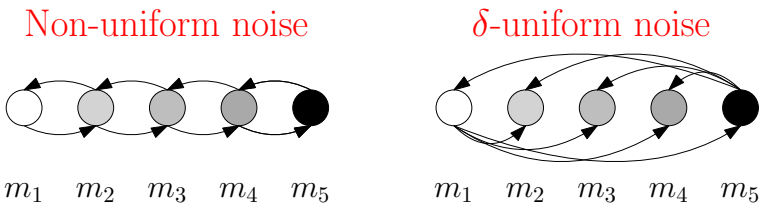


Figure 1 Non-uniform noise vs. uniform noise. On the left, we consider an example with non-uniform noise. Assume that the message vocabulary consists of 5 symbols, that is, $\Sigma = \{m_1, m_2, m_3, m_4, m_5\}$, where $m_1 = 0$ and $m_5 = 1$, represent the opinions. Assume that noise can occur only between consecutive messages. For example, m_2 can be observed as either m_2, m_3 or m_1, all with positive constant probability, but can never be viewed as m_4 or m_5. In this scenario, the population can quickly converge on the correct opinion by executing the following. The sources always display the correct opinion, *i.e.,* either m_1 or m_5, and each other agent displays m_3 unless it has seen either m_1 or m_5 in which case it adopts the opinion it saw and displays it. In other words, m_3 serves as a default message for non-source agents, and m_1 and m_5 serve as attracting sinks. It is easy to see that the correct opinion will propagate quickly through the system without disturbance, and within $\mathcal{O}(\log n)$ number of rounds, where n is the size of the population, all agents will hold it with high probability. In contrast, as depicted on the right picture, if every message can be observed as any other message with some constant positive probability (for clarity, some of the arrows have been omitted from the sketch), then convergence cannot be achieved in less than $\Omega(n)$ rounds, as Theorem 2 dictates.

manner. More specifically, marking an edge corresponding to a message received for the first time by a node, the set of marked edges forms a spanning tree of low depth, rooted at the source. The depth of such tree can be interpreted as the deterioration of the message's reliability.

On the other hand, as shown here, in the parallel-$\mathcal{PULL}$ model, even with the synchonization assumption, rumor spreading cannot be achieved in less than a linear number of rounds. Perhaps the main reason why these two models are often considered similar is that with an extra bit in the message, a $\mathcal{PUSH}$ protocol can be *approximated* in the $\mathcal{PULL}$ model, by letting this bit indicate whether the agent in the $\mathcal{PUSH}$ model was aiming to push its message. However, for such a strategy to work, this extra bit has to be reliable. Yet, in the noisy $\mathcal{PULL}$ model, no bit is safe from noise, and hence, as we show, such an approximation cannot work. In this sense, the extra power that the noisy $\mathcal{PUSH}$ model gains over the noisy $\mathcal{PULL}$ model, is that the very fact that one node attempts to communicate with another is reliable. This, seemingly minor, difference carries significant consequences.

1.3.3 Generalizations

Several of the assumptions discussed earlier for the parallel-$\mathcal{PULL}$ model were made for the sake of simplicity of presentation. In fact, our results can be shown to hold under more general conditions, that include: 1) different rate for sampling a source, and 2) a more relaxed noise criterion.

In addition, our theorems were stated with respect to the parallel-$\mathcal{PULL}$ model. In this model, at every round, each agent samples a single agent u.a.r. In fact, for any integer k, our analysis can be applied to the model in which, at every round, each agent observes k agents chosen u.a.r. In this case, the lower bound would simply reduce by a factor of k. Our analysis can also apply to a sequential variant, in which in each time step, two agents u and

v are chosen u.a.r from the population and u observes v. In this case, our lower bounds would multiply by a factor of n, yielding, for example, a lower bound of $\Omega(n^2)$ in the case where δ and s are constants[1].

1.3.4 Recruitment in desert ants

Our theoretical results assert that efficient rumor spreading in large groups could not be achieved without some degree of communication reliability. An example of a biological system whose communication reliability appears to be deficient in all of its components is recruitment in *Cataglyphis niger* desert ants. In this species, when a forager locates an oversized food item, she returns to the nest to recruit other ants to help in its retrieval [4, 34].

We complement our theoretical findings by providing new analyses from an experiment on this system conducted at the Weizmann Institute of Science [34]. In such experimental setting, we interpret our theoretical findings as an abstraction of the interaction modes between ants. While such high-level approximation may be considered very crude, we retain that it constitutes a good trade-off between analytical tractability and experimental data.

In our experimental setup recruitment happens in the small area of the nest's entrance chamber (Figure 2a). We find that within this confined area, the interactions between ants are nearly uniform [32], such that an ant cannot control which of her nest mates she meets next (see Figure 2b). This random meeting pattern coincides with the first main assumption of our model. Additionally, it has been shown that recruitment in *Cataglyphis niger* ants relies on rudimentary alerting interactions [18, 27] which are subject to high levels of noise [34]. Furthermore, the responses to a recruiting ant and to an ant that is randomly moving in the nest are extremely similar [34]. Although this may resemble a noisy push interaction scheme, ants cannot reliably distinguish an ant that attempts to transmit information from any other non-communicating individual. In our theoretical framework, the latter fact means that the structure of communication is captured by a noisy-pull scheme (see more details about $\mathcal{PUSH}$ vs. $\mathcal{PULL}$ in Section 1.3.2).

It has previously been shown that the information an ant passes in an interaction can be attributed solely to her speed before the interaction [34]. Binning ant speeds into four arbitrary discrete messages and measuring the responses of stationary ants to these messages, we can estimate the probabilities of one message to be mistakenly perceived as another one (see Materials and Methods). Indeed, we find that this communication is extremely noisy and complies with the uniform-noise assumption with a δ of approximately 0.2 (Figure 2c).

Given the coincidence between the communication patterns in this ant system and the requirements of our lower bound we expect long delays before any uninformed ant can be relatively certain that a recruitment process is occurring. We therefore measured the time it takes an ant, that has been at the food source, to recruit the help of two nest-mates. We find that this time increases with group size ($p < 0.05$ Kolmogorov-Smirnov test over $N = 24$ experiments, Figure 2d). Thus, in this system, inherently noisy interactions on the microscopic level have direct implications on group level performance. While group sizes in these experiments are small, we nevertheless find these recruitment times in accordance with our asymptotic theoretical results. More details on the experimental methodology can be found in the full version of the paper.

[1] This increase in not surprising as each round in the parallel-$\mathcal{PULL}$ model consists of n observations, while the sequential model consists of only one observation in each time step.

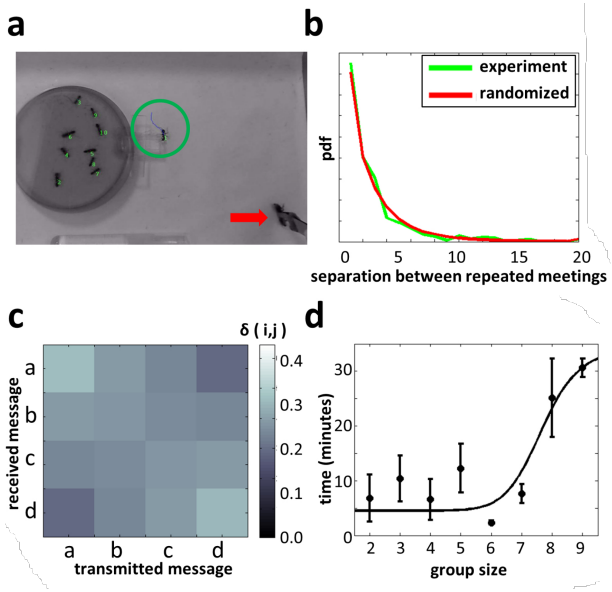


Figure 2 Unreliable communication and slow recruitment by desert ant (*Cataglyphis niger*). a. The experimental setup. The recruiter ant (circled) returns to the nest's entrance chamber (dark, 9cm diameter, disc) after finding the immobilized food item (arrow). Group size is ten. **b.** A *pdf* of the number of interactions that an ant experiences before meeting the same ant twice. The *pdf* is compared to uniform randomized interaction pattern. Data summarizes $N = 671$ interactions from seven experiments with a group size of 6 ants. **c.** Interactions with moving ants where classified into four different messages ('a' to 'd') depending on the ants' speed. The noise at which messages were confused with each other was estimated according to the response recipient, initially stationary, ants (see Materials and Methods). Gray scale indicates the estimated overlap between every two messages $\delta(i, j)$. Note that $\delta = \min(\delta(i, j)) \approx 0.2$. Data collected over $N = 64$ interactions. **d.** The mean time it takes an ant that is informed about the food to recruit two nest-mates to exit the nest is presented for two group size ranges.

1.4 Related work

Lower bound approaches in biological contexts are still extremely rare [8, 20]. Our approach can be framed within the general endeavour of addressing problems in theoretical biology through the algorithmic perspective of theoretical computer science [14, 13].

The computational study of abstract systems composed of simple individuals that interact using highly restricted and stochastic interactions has recently been gaining considerable attention in the community of theoretical computer science. Popular models include *population protocols* [7], which typically consider constant size individuals that interact in pairs (using constant size messages) in random communication patterns, and the *beeping* model [41], which assumes a fixed network with extremely restricted communication. Our model also falls in this framework as we consider the $\mathcal{PULL}$ model [16, 28, 29] with constant size messages. So far, despite interesting works that consider different fault-tolerant contexts [5, 6], most of the progress in this framework considered noiseless scenarios.

In *Rumor Spreading* problems (also referred to as *Broadcast*) a piece of information typically held by a single designated agent is to be disseminated to the rest of the population. It is the subject of a vast literature in theoretical computer science, and more specifically in the distributed computing community, see, *e.g.,* [10, 12, 16, 17, 22, 26, 28, 33]. While some works assume a fixed topology, the canonical setting does not assume a network. Instead

agents communicate through uniform $\mathcal{PUSH}/\mathcal{PULL}$ based interactions (including the *phone call* model), in which agents interact in pairs with other agents independently chosen at each time step uniformly at random from all agents in the population. The success of such protocols is largely due to their inherent simplicity and fault-tolerant resilience [19, 28]. In particular, it has been shown that under the $\mathcal{PUSH}$ model, there exist efficient rumor spreading protocol that uses a single bit per message and can overcome flips in messages (noise) [22].

The line of research initiated by El-Gamal [15], also studies a broadcast problem with noisy interactions. The regime however is rather different from ours: all n agents hold a bit they wish to transmit to a single receiver. This line of research culminated in the $\Omega(n \log \log n)$ lower bound on the number of messages shown in [26], matching the upper bound shown many years sooner in [24].

2 Formal description of the models

We consider a population of n agents that interact stochastically and aim to converge on a particular opinion held by few knowledgable individuals. For simplicity, we assume that the set of opinions contain two opinions only, namely, 0 and 1.

As detailed in this section, we shall assume that agents have access to significant amount of resources, often exceeding reasonable more realistic assumptions. Since we are concerned with lower bounds, we do not loose generality from such permissive assumptions. These liberal assumptions will actually simplify our proofs. One of these assumptions is the assumption that each agent is equipped with a unique identity $id(v)$ in the range $\{1, 2, \ldots, n\}$ (see more details in Section 2.4).

2.1 Initial configuration

The initial configuration is described in several layers. First, the *neutral initial configuration* corresponds to the initial states of the agents, before the sources and the desired opinion to converge to are set. Then, a random initialization is applied to the given neutral initial configuration, which determines the set of sources and the opinion that agents need to converge to. This will result in what we call the *charged initial configuration*. It can represent, for example, an external event that was identified by few agents which now need to deliver their knowledge to the rest of the population.

Neutral Initial Configuration $\mathbf{x^{(0)}}$. Each agent v starts the execution with an *input* that contains, in addition to its identity, an initial *state* taken from some discrete set of states, and[2] a binary *opinion* variable $\lambda_v \in \{0, 1\}$. The *neutral initial configuration* $\mathbf{x}^{(0)}$ is the vector whose i'th index, $\mathbf{x}_i^{(0)}$ for $i \in [n]$, is the input of the agent with identity i.

Charged Initial Configuration and Correct Opinion. The charged initial configuration is determined in three stages. The first corresponds to the random selection of sources, the second to the selection of the correct opinion, and the third to a possible update of states of sources, as a result of being selected as sources with a particular opinion.

[2] The opinion of an agent could have been considered as part of the state of the agent. We separate these two notions merely for the presentation purposes.

- **1st stage - Random selection of sources.** Given an integer $s \leq n$, a set S of size s is chosen uniformly at random (u.a.r) among the agents. The agents in S are called *sources*. Note that any agent has equal probability of being a source. We assume that each source knows it is a source, and conversely, each non-source knows it is not a source.
- **2nd stage - Random selection of correct opinion.** In the main model we consider, after sources have been determined in the first stage, the sources are randomly initialized with an opinion, called the *correct opinion*. That is, a fair coin is flipped to determine an opinion in $\{0, 1\}$ and all sources are assigned with this opinion.
- **3rd stage - Update of initial states of sources.** To capture a change in behavior as a result of being selected as a source with a particular opinion, we assume that once the opinion of a source u has been determined, the initial state of u may change according to some distribution $f_{source-state}$ that depends on (1) its identity, (2) its opinion, and (3) the neutral configuration. Each source samples its new state independently.

2.2 Alphabet and noisy messages

Agents communicate by observing each other according to some random pattern (for details see Section 2.3). To improve communication agents may choose which content, called *message*, they wish to reveal to other agents that observe them. Importantly, however, such messages are subject to noise. More specifically, at any given time, each agent v (including sources) displays a message $m \in \Sigma$, where Σ is some finite alphabet. The alphabet Σ agents use to communicate may be richer than the actual information content they seek to disseminate, namely, their opinions. This, for instance, gives them the possibility to express several levels of certainty [30]. We can safely assume that the size of Σ is at least two, and that Σ includes both symbols 0 and 1. We are mostly concerned with the case where Σ is of constant size (*i.e.,* independent of the number of agents), but note that our results hold for any size of the alphabet Σ, as long as the noise criterion is satisfied (see below).

δ-uniform noise. When an agent u *observes* some agent v, it receives a sample of the message currently held by v. More precisely, for any $m, m' \in \Sigma$, let $P_{m,m'}$ be the probability that, any time some agent u observes an agent v holding some message $m \in \Sigma$, u actually receives message m'. The probabilities $P_{m,m'}$ define the entries of the noise-matrix P [23], which does not depend on time. We hereby also emphasize that the agents' samples are independent.

The noise in the sample is characterized by a *noise parameter* $0 < \delta \leq 1/2$. One of the important aspects in our theorems is that they are general enough to hold assuming *any* distribution governing the noise, as long as it satisfies the following noise criterion.

▶ **Definition 4** (The noise ellipticity parameter δ)**.** We say that the noise has ellipticity δ if $P_{m,m'} \geq \delta$ for any $m, m' \in \Sigma$.

Observe that the aforementioned criterion implies that $\delta \leq 1/|\Sigma|$, and that the case $\delta = 1/|\Sigma|$ corresponds to messages being completely random, and the rumor spreading problem is thus unsolvable. We next define a weaker criterion, that is particularly meaningful in cases in which sources are more restricted in their message repertoire than general agents. This may be the case, for example, if sources always choose to display their opinion as their message (possibly together with some extra symbol indicating that they are sources). Formally, we define $\Sigma' \subseteq \Sigma$ as the set of possible messages that a source can hold together with the set of messages that can be observed when viewing a source (*i.e.,* after noise is applied). Our

theorems actually apply to the following criterion, that requires that only messages in Σ' are attained due to noise with some sufficient probability.

▶ **Definition 5** (The relaxed noise ellipticity parameter δ)**.** We say that the noise has Σ'-relaxed ellipticity δ if $P_{m,m'} \geq \delta$ for any $m \in \Sigma$ and $m' \in \Sigma'$.

2.3 Random interaction patterns

We consider several basic interaction patterns. Our main model is the *parallel-*$\mathcal{PULL}$ model. In this model, time is divided into *rounds*, where at each round $i \in \mathbb{N}^+$, each agent u independently selects an agent v (possibly $u = v$) u.a.r from the population and then u observes the message held by v. The *parallel-*$\mathcal{PULL}$ model should be contrasted with the *parallel-*$\mathcal{PUSH}$ model, in which u can choose between *sending* a message to the selected node v or doing nothing. We shall also consider the following variants of $\mathcal{PULL}$ model.

- *parallel-*$\mathcal{PULL}$(k). Generalizing *parallel-*$\mathcal{PULL}$ for an integer $1 \leq k \leq n$, the *parallel-*$\mathcal{PULL}$(k) model allows agents to observe k other agents in each round. That is, at each round $i \in \mathbb{N}^+$, each agent independently selects a set of k agents (possibly including itself) u.a.r from the population and observes each of them.
- *sequential-*$\mathcal{PULL}$. In each time step $t \in \mathbb{N}^+$, two agents u and v are selected uniformly at random (u.a.r) among the population, and agent u observes v.
- *broadcast-*$\mathcal{PULL}$. It each time step $t \in \mathbb{N}^+$ one agent is chosen u.a.r. from the population and all agents observe it, receiving the same noisy sample of its message[3].

Regarding the difference in time units between the models, since interactions occur in parallel in the *parallel-*$\mathcal{PULL}$ model, one round in that model should informally be thought of as roughly n time steps in the *sequential-*$\mathcal{PULL}$ or *broadcast-*$\mathcal{PULL}$ model.

2.4 Liberal assumptions

As mentioned, we shall assume that agents have abilities that surpass their realistic ones. These assumption not only increases the generality of our lower bounds, but also simplifies their proofs. Specifically, the following liberal assumptions are considered.

- **Unique identities.** Each agent is equipped with a unique identity $id(v) \in \{1, 2, \ldots, n\}$, that is, for every two agents u and v, we have $id(u) \neq id(v)$. Moreover, whenever an agent u observes some agent v, we assume that u can infer the identity of v. In other words, we provide agents with the ability to reliably distinguish between different agents at no cost.
- **Unlimited internal computational power.** We allow agents to have unlimited computational abilities including infinite memory capacity. Therefore, agents can potentially perform arbitrarily complex computations based on their knowledge (and their id).
- **Complete knowledge of the system.** Informally, we assume that agents have access to the complete description of the system except for who are the sources and what is their opinion. More formally, we assume that each agent has access to:
  - the neutral initial configuration $\mathbf{x}^{(0)}$,

[3] The *broadcast-*$\mathcal{PULL}$ model is mainly used for technical considerations. We use it in our proofs as it simplifies our arguments while not harming their generality. Nevertheless, this broadcast model can also capture some situations in which agents can be seen simultaneously by many other agents, where the fact that all agents observe the same sample can be viewed as noise being originated by the observed agent.

- all the systems parameters, including the number of agents n, the noise parameter δ, the number of sources s, and the distribution $f_{source-state}$ governing the update the states of sources in the third stage of the charged initial configuration.

- **Full synchronization.** We assume that all agents are equipped with clocks that can count time steps (in *sequential-*$\mathcal{PULL}$ or *broadcast-*$\mathcal{PULL}$) or rounds (in *parallel-*$\mathcal{PULL}$(k)). The clocks are synchronized, ticking at the same pace, and initialized to 0 at the beginning of the execution. This means, in particular, that if they wish, the agents can actually share a notion of time that is incremented at each time step.
- **Shared randomness.** We assume that algorithms can be randomized. That is, to determine the next action, agents can internally toss coins and base their decision on the outcome of these coin tosses. Being liberal, we shall assume that randomness is shared in the following sense. At the outset, an arbitrarily long sequence r of random bits is generated and the very same sequence r is written in each agent's memory before the protocol execution starts. Each agent can then deterministically choose (depending on its state) which random bits in r to use as the outcome of its own random bits. This implies that, for example, two agents can possibly make use of the very same random bits or merely observe the outcome of the random bits used by the other agents. Note that the above implies that, conditioning on an agent u being a non-source agent, all the random bits used by u during the execution are accessible to all other agents.
- **Coordinated sources.** Even though non-source agents do not know who the sources are, we assume that sources do know who are the other sources. This means, in particular, that the sources can coordinate their actions.

2.5 Considered algorithms and solution concept

Upon observation, each agent can alter its internal state (and in particular, its message to be seen by others) as well as its opinion. The strategy in which agents update these variables is called "algorithm". As mentioned, algorithms can be randomized, that is, to determine the next action, agents can use the outcome of coin tosses in the sequence r (see *Shared randomness* in Section 2.4). Overall, the action of an agent u at time t depends on:

1. the initial state of u in the charged initial configuration (including the identity of u and whether or not it is a source),
2. the initial knowledge of u (including the system's parameters and neutral configuration),
3. the time step t, and the list of its observations (history) up to time $t-1$, denoted $x_u^{(<t)}$,
4. the sequence of random bits r.

2.6 Convergence and time complexity

At any time, the opinion of an agent can be viewed as a binary *guess* function that is used to express its most knowledgeable guess of the correct opinion. The agents aim to minimize the probability that they fail to guess this opinion. In this context, it can be shown that the optimal guessing function is deterministic.

▶ **Definition 6.** We say that *convergence* has been achieved if one can specify a particular non-source agent v, for which is it guaranteed that its opinion is the correct opinion with probability at least 2/3. The *time complexity* is the number of time steps (respectively, rounds) required to achieve convergence.

We remark that the latter definition encompasses all three models considered.

▶ Remark (Different sampling rates of sources). We consider sources as agents in the population but remark that they can also be thought of as representing the environment. In this case, one may consider a different rate for sampling a source (environment) vs. sampling a typical agent. For example, the probability to observe any given source (or environment) may be x times more than the probability to observe any given non-source agent. This scenario can also be captured by a slight adaptation of our analysis. When x is an integer, we can alternatively obtain such a generalization by considering additional *artificial* sources in the system. Specifically, we replace each source u_i with a set of sources U_i consisting of x sources that coordinate their actions and behave identically, simulating the original behavior of u_i. (Recall that we assume that sources know who are the other sources and can coordinate their actions.) Since the number of sources increases by a multiplicative factor of x, our lower bounds (see Theorem 7 and Corollary 3) decrease by a multiplicative factor of x^2.

3 The lower bounds

Throughout this section we consider $\delta < 1/2$, such that $\frac{(1-2\delta)}{\delta sn} \leq \frac{1}{10}$. Our goal in this section is to prove the following result.

▶ **Theorem 7.** *Assume that the relaxed δ-uniform noise criterion is satisfied.*

- *Let k be an integer. Any rumor spreading protocol on the* parallel-$\mathcal{PULL}(k)$ *model cannot converge in fewer rounds than $\Omega\left(\frac{n\delta}{ks^2(1-2\delta)^2}\right)$.*
- *Consider either the* sequential-$\mathcal{PULL}$ *or the* broadcast-$\mathcal{PULL}$ *model. Any rumor spreading protocol cannot converges in fewer rounds than $\Omega\left(\frac{n^2\delta}{s^2(1-2\delta)^2}\right)$.*

To prove the theorem, we first prove (in Section 3.1) that an efficient rumor spreading algorithm in either the noisy *sequential-*$\mathcal{PULL}$ model or the *parallel-*$\mathcal{PULL}$(k) model can be used to construct an efficient algorithm in the *broadcast-*$\mathcal{PULL}$ model. The resulted algorithm has the same time complexity as the original one in the context of *sequential-*$\mathcal{PULL}$ and adds a multiplicative factor of kn in the context of *parallel-*$\mathcal{PULL}$(k).

We then show how to relate the rumor spreading problem in *broadcast-*$\mathcal{PULL}$ to a statistical inference test (Section 3.2). A lower bound on the latter setting is then achieved by adapting techniques from mathematical statistics (Section 3.3).

3.1 Reducing to the *broadcast*-$\mathcal{PULL}$ Model

The following lemma establishes a formal relation between the convergence times of the models we consider. We assume all models are subject to the same noise distribution.

▶ **Lemma 8.** *Any protocol operating in* sequential-$\mathcal{PULL}$ *can be simulated by a protocol operating in* broadcast-$\mathcal{PULL}$ *with the same time complexity. Moreover, for any integer $1 \leq k \leq n$, any protocol $\mathcal{P}$ operating in* parallel-$\mathcal{PULL}(k)$ *can be simulated by a protocol operating in* broadcast-$\mathcal{PULL}$ *with a time complexity that is kn times that of $\mathcal{P}$ in* parallel-$\mathcal{PULL}(k)$.

Proof. Let us first show how to simulate a time step of *sequential-*$\mathcal{PULL}$ in the *broadcast-*$\mathcal{PULL}$ model. Recall that in *broadcast-*$\mathcal{PULL}$, in each time step, all agents receive the same observation sampled u.a.r from the population. Upon drawing such an observation, all agents use their shared randomness to generate a (shared) uniform random integer X between 1 and n. Then, the agent whose unique identity corresponds to X is the one processing the observation, while all other agents ignore it. This reduces the situation to a scenario in *sequential-*$\mathcal{PULL}$, and the agents can safely execute the original algorithm designed for that model.

As for simulating a time step of *parallel-$\mathcal{PULL}$*(k) in *broadcast-$\mathcal{PULL}$*, agents divide time steps in the latter model into *rounds*, each composing of precisely kn time steps. Recall that the model assumes that agents share clocks that start when the execution starts and tick at each time step. This implies that the agents can agree on the division of time into rounds, and can further agree on the round number. For $1 \leq i \leq kn$, during the i-th step of each round, only the agent whose identity is $(i \bmod n)+1$ receives[4] the observation, while all other agents ignore it. This ensures that when a round is completed in the *broadcast-$\mathcal{PULL}$* model, each agent receives precisely k independent uniform samples as it would in a round of *parallel-$\mathcal{PULL}$*(k). Therefore, at the end of each round $j \in \mathbb{N}^+$ in the *broadcast-$\mathcal{PULL}$* model, all agents can safely execute their actions in the j'th round of the original protocol designed for *parallel-$\mathcal{PULL}$*(k). This draws a precise bijection from rounds in *parallel-$\mathcal{PULL}$*(k) and rounds in *broadcast-$\mathcal{PULL}$*. The multiplicative overhead of kn simply follows from the fact that each round in *broadcast-$\mathcal{PULL}$* consists of kn time steps. ◀

Thanks to Lemma 8, Theorem 7 directly follows from the next theorem.

▶ **Theorem 9.** *Consider the* broadcast-$\mathcal{PULL}$ *model and assume that the relaxed δ-uniform noise criterion is satisfied. Any rumor spreading protocol cannot converges in fewer time steps than* $\Omega\left(\frac{n^2\delta}{s^2(1-2\delta)^2}\right)$.

The remaining of the section is dedicated to proving Theorem 9. Towards achieving this, we view the task of guessing the correct opinion in the *broadcast-$\mathcal{PULL}$* model, given access to noisy samples, within the more general framework of distinguishing between two types of stochastic processes which obey some specific assumptions.

3.2 Rumor Spreading and hypothesis testing

To establish the desired lower bound, we next show how the rumor spreading problem in the broadcast-$\mathcal{PULL}$ model relates to a statistical inference test. That is, from the perspective of a given agent, the rumor spreading problem can be understood as the following: Based on a sequence of noisy observations, the agent should be able to tell whether the correct opinion is 0 or 1. We formulate this problem as a specific task of distinguishing between two random processes, one originated by running the protocol assuming the correct opinion is 0 and the other assuming it is 1.

One of the main difficulties lies in the stochastic dependencies affecting these processes. In general, at different time steps, they do not consist of independent draws of a given random variable. In other words, the law of an observation not only depends on the correct opinion, on the initial configuration and on the underlying randomness used by agents, but also on the previous noisy observation samples and (consequently) on the messages agents themselves choose to display on that round. An intuitive version of this problem is the task of distinguishing between two (multi-valued) biased coins, whose bias changes according to the previous outcomes of tossing them (*e.g.,* due to wear). Following such intuition, we define the following general class of *Adaptive Coin Distinguishing Tasks*, for short `ACDT`.

▶ **Definition 10** (`ACDT`)**.** A *distinguisher* is presented with a sequence of observations taken from a coin of type η where $\eta \in \{0, 1\}$. The type η is initially set to 0 or 1 with probability 1/2 (independently of everything else). The goal of the distinguisher is to determine the type

[4] Receiving the observation doesn't imply that the agent processes this observation. In fact, it will store it in its memory until the round is completed, and process it only then.

η, based on the observations. More specifically, for a given time step t, denote the sequence of previous observations (up to, and including, time $t-1$) by $x^{(<t)} = (x^{(1)}, \ldots, x^{(t-1)})$. At each time t, given the type $\eta \in \{0,1\}$ and the history of previous observations $x^{(<t)}$, the distinguisher receives an observation $X_\eta^{(t)} \in \Sigma$, which has law[5] $P(X_\eta^{(t)} = m \mid x^{(<t)})$.

We next introduce, for each $m \in \Sigma$, the parameter $\varepsilon(m, x^{(<t)}) = P(X_1^{(t)} = m \mid x^{(<t)}) - P(X_0^{(t)} = m \mid x^{(<t)})$. Since, at all times t, it holds that $\sum_{m\in\Sigma} P(X_0^{(t)} = m \mid x^{(<t)}) = \sum_{m\in\Sigma} P(X_1^{(t)} = m \mid x^{(<t)}) = 1$, then $\sum_{m\in\Sigma} \varepsilon(m, x^{(<t)}) = 0$. We shall be interested in the quantity $d_\varepsilon(x^{(<t)}) := \sum_{m\in\Sigma} |\varepsilon(m, x^{(<t)})|$, which corresponds to the ℓ_1 distance between the distributions $P(X_0^{(t)} = m \mid x^{(<t)})$ and $P(X_1^{(t)} = m \mid x^{(<t)})$ given the sequence of previous observations.

▶ **Definition 11** (The bounded family $\texttt{ACDT}(\varepsilon, \delta)$)**.** We consider a family of instances of ACDT, called $\texttt{ACDT}(\varepsilon, \delta)$, governed by parameters ε and δ. Specifically, this family contains all instances of ACDT such that for every t, and every history $x^{(<t)}$, we have:

- $d_\varepsilon(x^{(<t)}) \leq \varepsilon$, and
- $\forall m \in \Sigma$ such that $\varepsilon(m, x^{(<t)}) \neq 0$, we have $\delta \leq P(X_\eta^{(t)} = m \mid x^{(<t)})$ for $\eta \in \{0,1\}$.

In the rest of the section, we show how Theorem 9, that deals with the *broadcast-$\mathcal{PULL}$* model, follows directly from the next theorem that concerns the adaptive coin distinguishing task, by setting $\varepsilon = \frac{2s(1-2\delta)}{n}$. The actual proof of Theorem 12 appears in Section 3.3.

▶ **Theorem 12.** *Consider any protocol for any instance of* $\texttt{ACDT}(\varepsilon, \delta)$*, The number of samples required to distinguish between a process of type* 0 *and a process of type* 1 *with probability of error less than* $\frac{1}{3}$ *is at least* $\frac{\ln 2}{9}\left(\frac{6(\delta-\varepsilon)^3}{\delta^3-\delta^2\varepsilon+3\delta\varepsilon^2-\varepsilon^3}\right)\frac{\delta}{\varepsilon^2}$*. In particular, if* $\frac{\varepsilon}{\delta} < 10$*, then the number of necessary samples is* $\Omega\left(\frac{\delta}{\varepsilon^2}\right)$*.*

3.2.1 Proof of Theorem 9 assuming Theorem 12

Consider a rumor spreading protocol $\mathcal{P}$ in the *broadcast-$\mathcal{PULL}$* model. Fix a node u. We first show that running $\mathcal{P}$ by all agents, the perspective of node u corresponds to a specific instance of $\texttt{ACDT}(\frac{2\mathtt{s}(1-2\delta)}{\mathtt{n}}, \delta)$ called $\Pi(\mathcal{P}, u)$. We break down the proof of such correspondence into two claims.

3.2.1.1 The ACDT instance $\Pi(\mathcal{P}, u)$.

Recall that we assume that each agent knows the complete neutral initial configuration, the number of sources s, and the shared of random bits sequence r. We avoid writing such parameters as explicit arguments to $\Pi(\mathcal{P}, u)$ in order to simplify notation, however, we stress that what follows assumes that these parameters are fixed. The bounds we show hold for any fixed value of r and hence also when r is randomized.

Each agent is interested in discriminating between two families of charged initial configurations: Those in which the correct opinion is 0 and those in which it is 1 (each of these possibilities occurs with probability $\frac{1}{2}$). Recall that the correct opinion is determined in the 2nd stage of the charged initial configuration, and is independent form the choice of sources (1st stage).

[5] We follow the common practice to use uppercase letters to denote random variables and lowercase letter to denote a particular realisation, e.g., $\mathbf{X}^{(\leq t)}$ for the sequence of observations up to time t, and $\mathbf{x}^{(\leq t)}$ for a corresponding realization.

We next consider the perspective of a generic non-source agent u, and define the instance $\Pi(\mathcal{P}, u)$ as follows. Given the history $x^{(<t)}$, we set $P(X_\eta^{(t)} = m \mid x^{(<t)})$, for $\eta \in \{0,1\}$, to be equal to the probability that u observes message $m \in \Sigma$ at time step t of the execution $\mathcal{P}$. For clarity's sake, we remark that the latter probability is conditional on: the history of observations being $x^{(<t)}$, ,the sequence of random bits r, ,the correct opinion being $\eta \in \{0,1\}$, ,the neutral initial configuration, ,the identity of u, ,the algorithm $\mathcal{P}$, and the system's parameters (including the distribution $f_{source-state}$ and the number of sources s).

▶ **Claim 13.** *Let $\mathcal{P}$ be a correct protocol for the rumor spreading problem in* broadcast-$\mathcal{PULL}$ *and let u be an agent for which the protocol is guaranteed to produce the correct opinion with probability at least p by some time T (if one exists), for any fixed constant $p \in (0,1)$. Then $\Pi(\mathcal{P}, u)$ can be solved in time T with correctness being guaranteed with probability at least p.*

Proof. Conditioning on $\eta \in \{0,1\}$ and on the random seed r, the distribution of observations in the $\Pi(\mathcal{P}, u)$ instance follows precisely the distribution of observations as perceived from the perspective of u in *broadcast-$\mathcal{PULL}$*. Hence, if the protocol $\mathcal{P}$ at u terminates with output $j \in \{0,1\}$ at round T, after the T-th observation in $\Pi(\mathcal{P}, u)$ we can set $\Pi(\mathcal{P}, u)$'s output to j as well. Given that the two stochastic processes have the same law, the correctness guarantees are the same. ◀

▶ **Lemma 14.** $\Pi(\mathcal{P}, u) \in \texttt{ACDT}\left(\frac{2(1-2\delta)\mathtt{s}}{\mathtt{n}}, \delta\right)$.

Proof. Since the noise in *broadcast-$\mathcal{PULL}$* flips each message $m \in \Sigma$ into any $m' \in \Sigma'$ with probability at least δ, regardless of the previous history and of $\eta \in \{0,1\}$, at all times t, if $m \in \Sigma'$ then $P(X_\eta^{(t)} = m \mid x^{(<t)}) \geq \delta$. Consider a message $m \in \Sigma \setminus \Sigma'$ (if such a message exists). By definition, such a message could only be received by observing a non-source agent. But given the same history $x^{(<t)}$, the same sequence of random bits r, and the same initial knowledge, the behavior of a non-source agent is the same, no matter what is the correct opinion η. Hence, for $m \in \Sigma \setminus \Sigma'$ we have $P(X_0^{(t)} = m \mid x^{(<t)}) = P(X_1^{(t)} = m \mid x^{(<t)})$, or in other words, $m \in \Sigma \setminus \Sigma' \implies \varepsilon(m, x^{(<t)}) = 0$.

It remains to show that $d_\varepsilon(x^{(<t)}) \leq \frac{2(1-2\delta)s}{n}$. Let us consider two executions of the rumor spreading protocol, with the same neutral initial configuration, same shared sequence of random bits r, same set of sources, except that in the first the correct opinion is 0 while in the other it is 1. Let us condition on the history of observations $x^{(<t)}$ being the same in both processes. As mentioned, given the same history $x^{(<t)}$, the behavior of a non-source agent is the same, regardless of the correct opinion η. It follows that the difference in the probability of observing any given message is only due to the event that a source is observed. Recall that the number of sources is s. Therefore, the probability of observing a source is s/n, and we may write as a first approximation $\varepsilon(m, x^{(<t)}) \leq s/n$. However, we can be more precise. In fact, $\varepsilon(m, x^{(<t)})$ is slightly smaller than s/n, because the noise can still affect the message of a source. We may interpret $\varepsilon(m, x^{(<t)})$ as the following difference. For a source $v \in S$, let m_η^v be the message of u assuming the given history $x^{(<t)}$ and that v is of type $\eta \in \{0,1\}$ (the message m_η^v is deterministically determined given the sequence r of random bits, the neutral initial configuration, the parameters of the system, and the identity of v). Let $\alpha_{m',m}$ be the probability that the noise transforms a message m' into a message m. Then $\varepsilon(m, x^{(<t)}) = \frac{1}{n}\sum_{v \in S}(\alpha_{m_1^v,m} - \alpha_{m_0^v,m})$, and

$$d_\varepsilon(x^{(<t)}) = \sum_{m \in \Sigma} |\varepsilon(m, x^{(<t)})| \leq \frac{1}{n} \sum_{m \in \Sigma} \sum_{v \in S} |\alpha_{m_1^v,m} - \alpha_{m_0^v,m}|. \tag{1}$$

By the definition of $\texttt{ACDT}(\varepsilon,\delta)$, it follows that either $\alpha_{m_1^v,m} = \alpha_{m_0^v,m}$ (if $\varepsilon(m, x^{(<t)}) = 0$) or $\delta \leq \alpha_{m_1^v,m}, \alpha_{m_0^v,m} \leq 1-\delta$ (if $\varepsilon(m, x^{(<t)}) \neq 0$). Thus, to bound the right hand side in (1), we can use the following claim (proven in Appendix 3.2.1.1)

▶ **Claim 15.** *Let P and Q be two distributions over a universe Σ such that for any element $m \in \Sigma$, $\delta \leq P(m), Q(m) \leq 1-\delta$. Then $\sum_{m\in\Sigma}|P(m)-Q(m)| \leq 2(1-2\delta)$.*

Proof of Claim 15. Let $\Sigma_+ := \{m : P(m) > Q(m)\}$. We may write

$$\begin{aligned}\sum_{m\in\Sigma}|P(m)-Q(m)| &= \sum_{m\in\Sigma_+}(P(m)-Q(m)) + \sum_{m\in setminus\Sigma_+}(Q(m)-P(m))\\ &= P(\Sigma_+)-Q(\Sigma_+)+Q(\Sigma\setminus\Sigma_+)-P(\Sigma\setminus\Sigma_+)\\ &= 2(P(\Sigma_+)-Q(\Sigma_+)),\end{aligned}$$

where in the last line we used the fact that $Q(\Sigma\setminus\Sigma_+)-P(\Sigma\setminus\Sigma_+) = 1-Q(\Sigma_+)-1+P(\Sigma_+) = P(\Sigma_+)-Q(\Sigma_+)$. We now distinguish two cases. **Case 1.**If Σ_+ is a singleton, $\Sigma_+ = \{m^*\}$, then $P(\Sigma_+)-Q(\Sigma_+) = P(m^*)-Q(m^*) \leq 1-2\delta$, by assumption. **Case 2.** Otherwise, $|\Sigma_+| \geq 2$ and $2\sum_{m\in\Sigma_+}(P(m)-Q(m)) \leq 2-2\sum_{m\in\Sigma_+}Q(m) \leq 2(1-\delta|\Sigma_+|) \leq 2(1-2\delta)$, using the fact that for any m, $Q(m) \geq \delta$, and the fact that P is a probability measure. This completes the proof of Claim 15. ◀

Applying Claim 15 for a fixed $v \in S$ to distributions $(\alpha_{m_0^v,m})_m$ and $(\alpha_{m_1^v,m})_m$, we obtain

$$\frac{1}{n}\sum_{m\in\Sigma}\sum_{v\in S}|\alpha_{m_1^v,m}-\alpha_{m_0^v,m}| \leq \frac{1}{n}2\sum_{v\in S}(1-2\delta) \leq \frac{2(1-2\delta)s}{n}.$$

Hence, we have $\Pi(\mathcal{P}) \in \texttt{ACDT}\left(\frac{2(1-2\delta)\mathtt{s}}{\mathtt{n}},\delta\right)$, establishing Lemma 14. ◀

Thanks to Claims 13 and Lemma 14, Theorem 9 regarding the *broadcast-$\mathcal{PULL}$* model becomes a direct consequence of Theorem 12 on the adaptive coin distinguishing task, taking $\varepsilon = \frac{2(1-2\delta)s}{n}$. More precisely, the assumption $\frac{(1-2\delta)}{\delta sn} \leq c$ for some small constant c, ensures that $\frac{\varepsilon}{\delta} \leq c$ as required by Theorem 12. The lower bound $\Omega\left(\frac{\varepsilon^2}{\delta}\right)$ corresponds to $\Omega\left(\frac{n^2\delta}{(1-2\delta)^2s^2}\right)$. This concludes the proof of Theorem 9. ◀

To establish our results it remains to prove Theorem 12.

3.3 Proof of Theorem 12

We start by recalling some facts from Hypothesis Testing. First let us recall two standard notions of (pseudo) distances between probability distributions. Given two discrete distributions P_0, P_1 over a probability space Ω with the same support[6], the *total variation distance* is defined as $TV(P_0,P_1) := \frac{1}{2}\sum_{x\in\Omega}|P_0(x)-P_1(x)|$, and the Kullback-Leibler divergence $KL(P_0,P_1)$ is defined[7] as $KL(P_0,P_1) := \sum_{x\in\Omega}P_0(x)\log\frac{P_1(x)}{P_0(x)}$.

The following lemma shows that, when trying to discriminate between distributions P_0, P_1, the total variation relates to the smallest error probability we can hope for.

[6] The assumption that the support is the same is not necessary but it is sufficient for our purposes, and is thus made for simplicity's sake.

[7] We use the notation $\log(\cdot)$ to denote the base 2 logarithms, i.e., $\log_2(\cdot)$ and for a probability distribution P, use the notation $P(x)$ as a short for $P(X = x)$.

▶ **Lemma 16** (Neyman-Pearson [36, Lemma 5.3 and Proposition 5.4]). *Let P_0, P_1 be two distributions. Let X be a random variable of law either P_0 or P_1. Consider a (possibly probabilistic) mapping $f : \Omega \to \{0,1\}$ that attempts to "guess" whether the observation X was drawn from P_0 (in which case it outputs 0) or from P_1 (in which case it outputs 1). Then, we have the following lower bound,*

$$P_0\left(f(X)=1\right) + P_1\left(f(X)=0\right) \geq 1 - TV(P_0, P_1).$$

The total variation is related to the KL divergence by the following inequality.

▶ **Lemma 17** (Pinsker [36, Lemma 5.8]). *For any two distributions P_0, P_1,*

$$TV(P_0, P_1) \leq \sqrt{KL\left(P_0, P_1\right)}.$$

We are now ready to prove the theorem.

Proof of Theorem 12. Let us define $P_\eta(\,\cdot\,) = P(\,\cdot \mid \text{"correct distribution is } \eta\text{"})$ for $\eta \in \{0,1\}$. We denote $P_\eta^{(\leq t)}$, $\eta \in \{0,1\}$, the two possible distributions of $\mathbf{X}^{(\leq t)}$. We refer to $P_0^{(\leq t)}$ as the distribution of *type* 0 and to $P_1^{(\leq t)}$ as the distribution of *type* 1. Furthermore, we define the *correct type* of a sequence of observations $\mathbf{X}^{(\leq t)}$ to be 0 if the observations are sampled from $P_0^{(\leq t)}$, and to be 1 if they are sampled from $P_1^{(\leq t)}$.

After t observations $\mathbf{x}^{(\leq t)} = (x^{(1)}, \ldots, x^{(t)})$ we have to decide whether the distribution is of type 0 or 1. Our goal is to maximize the probability of guessing the type of the distribution, observing $\mathbf{X}^{(\leq t)}$, which means that we want to minimize

$$f = \sum_{\eta \in \{0,1\}} P_\eta\left(f(\mathbf{X}^{(\leq t)}) = 1-\eta\right) P\left(\text{"correct type is } \eta\text{"}\right). \tag{2}$$

Recall that the correct type is either 0 or 1 with probability $\frac{1}{2}$. Thus, the error probability described in (2) becomes

$$\frac{1}{2} P_0\left(f(\mathbf{X}^{(\leq t)}) = 1\right) + \frac{1}{2} P_1\left(f(\mathbf{X}^{(\leq t)}) = 0\right). \tag{3}$$

By combining Lemmas 16 and 17 with $X = \mathbf{X}^{(\leq t)}$ and $P_\eta = P_\eta^{(\leq t)}$ for $\eta = 0, 1$, we get the following Theorem. Although for convenience we think of f as a deterministic function, it could in principle be randomized.

▶ **Theorem 18.** *Let f be any guess function. Then*

$$P_0\left(ft(\mathbf{X}^{(\leq t)}) = 1\right) + P_1\left(f(\mathbf{X}^{(\leq t)}) = 0\right) \geq 1 - \sqrt{KL\left(P_0^{(\leq t)}, P_1^{(\leq t)}\right)}.$$

Theorem 18 implies that for the probability of error to be small, it must be the case that the term $KL\left(P_0^{(\leq t)}, P_1^{(\leq t)}\right)$ is large. Our next goal is therefore to show that in order to make this term large, t must be large.

Note that $P_\eta^{(\leq T)}$ for $\eta \in \{0,1\}$ cannot be written as the mere product of the marginal distributions of the $X^{(t)}$s, since the observations at different times may not necessarily be independent. Nevertheless, we can still express the term $KL(P_0^{(\leq T)}, P_1^{(\leq T)})$ as a sum, using

the Chain Rule for KL divergence[8]. It yields

$$KL(P_0^{(\leq T)}, P_1^{(\leq T)}) = \sum_{t \leq T} KL(P_0(x^{(t)} \mid x^{(<t)}), P_1(x^{(t)} \mid x^{(<t)})) \tag{4}$$

$$:= \sum_{x^{(<t)} \in \Sigma^{t-1}} P_0(x^{(<t)}) \sum_{x^{(t)} \in \Sigma} P_0(x^{(t)} \mid x^{(<t)}) \log \frac{P_0(x^{(t)} \mid x^{(<t)})}{P_1(x^{(t)} \mid x^{(<t)})}.$$

$$= \sum_{x^{(<t)} \in \Sigma^{t-1}} P_0(x^{(<t)}) \sum_{m \in \Sigma} P_0(X_0^{(t)} = m \mid x^{(<t)}) \log \frac{P(X_0^{(t)} = m \mid x^{(<t)})}{P(X_1^{(t)} = m \mid x^{(<t)})}. \tag{5}$$

Since we are considering an instance of $\texttt{ACDT}(\varepsilon, \delta)$, we have

- $d_\varepsilon(x^{(<t)}) = \sum_{m \in \Sigma} |\varepsilon(m, x^{(<t)})| \leq \varepsilon$, and
- for every $m \in \Sigma$ such that $\varepsilon(m, x^{(<t)}) \neq 0$, it holds that $\delta \leq P_\eta(X_0^{(t)} = m \mid x^{(<t)})$ for $\eta \in \{0, 1\}$.

We make use of the previous facts to upper bound the KL divergence terms in the right hand side of (5), as follows.

$$\begin{aligned} &KL(P_0(x^{(t)} \mid x^{(<t)}), P_1(x^{(t)} \mid x^{(<t)})) \\ &= \sum_{x^{(<t)} \in \Sigma^{t-1}} P_0(x^{(<t)}) \sum_{m \in \Sigma} \left(P(X_0^{(t)} = m \mid x^{(<t)}) \log \frac{P(X_0^{(t)} = m \mid x^{(<t)})}{P(X_0^{(t)} = m \mid x^{(<t)}) + \varepsilon(m, x^{(<t)})} \right) \\ &= -\sum_{x^{(<t)}} P_0(x^{(<t)}) \sum_{m \in \Sigma} \left(P(X_0^{(t)} = m \mid x^{(<t)}) \log \left(1 + \frac{\varepsilon(m, x^{(<t)})}{P(X_0^{(t)} = m \mid x^{(<t)})} \right) \right). \end{aligned} \tag{6}$$

Recall that we assume $\frac{\varepsilon(m, x^{(<t)})}{P(X_0^{(t)} = m \mid x^{(<t)})} \leq \frac{\varepsilon(m, x^{(<t)})}{\delta} \leq \frac{\varepsilon}{\delta}$. We make use of the following claim, which follows from the Taylor expansion of $\log(1+u)$ around 0.

▶ **Claim 19.** *Let $x \in [-a, a]$ for some $a \in (0, 1)$. Then $|\log(1+x) - x + x^2/2| \leq \frac{x^3}{3(1-a)^3}$.*

Using Claim 19 with $a = \frac{\varepsilon}{\delta}$, we can bound the inner sum appearing in (6) from above and below with

$$\frac{1}{\ln 2} \sum_{m \in \Sigma} \left(\varepsilon(m, x^{(<t)}) - \frac{1}{2} \frac{(\varepsilon(m, x^{(<t)}))^2}{P(X_0^{(t)} = m \mid x^{(<t)})} \pm \frac{\delta^3}{3(\delta - \varepsilon)^3} \left(\frac{(\varepsilon(m, x^{(<t)}))^3}{P(X_0^{(t)} = m \mid x^{(<t)})^2} \right) \right). \tag{7}$$

Since $\sum_m |\varepsilon(m, x^{(<t)})| \leq \varepsilon$, we also have that $\sum_m (\varepsilon(m, x^{(<t)}))^2 \leq \varepsilon^2$. The latter bound, together with the fact that $P(X_0^{(t)} = \tilde{m} \mid x^{(<t)}) \geq \delta$ for any $\tilde{m} \in \Sigma$ such that $\varepsilon(\tilde{m}, x^{(<t)}) \neq 0$, implies

$$\sum_m \frac{(\varepsilon(m, x^{(<t)}))^2}{P(X_0^{(t)} = m \mid x^{(<t)})} \leq \frac{\varepsilon^2}{\delta}. \tag{8}$$

Finally, we can similarly bound the term $\sum_{m \in \Sigma} \left((\varepsilon(m, x^{(<t)}))^3 / P(X_0^{(t)} = m \mid x^{(<t)})^2 \right)$ with

$$\sum_{m \in \Sigma} \left((\varepsilon(m, x^{(<t)}))^3 / P(X_0^{(t)} = m \mid x^{(<t)})^2 \right) \leq \frac{\varepsilon^3}{\delta^2}. \tag{9}$$

[8] See Lemma 3 in `http://homes.cs.washington.edu/anuprao/pubs/CSE533Autumn2010/lecture3.pdf`.

Recall that $\sum_m \varepsilon(m, x^{(<t)}) = 0$, thus the first term in (7) disappears. Hence, substituting the bounds (8) and (9) in (7), we have

$$\left|\log\left(1+\frac{\varepsilon(m,x^{(<t)})}{P(X_0^{(t)}=m \mid x^{(<t)})}\right)\right| \leq \frac{1}{\ln 2}\left(\frac{1}{2}\frac{\varepsilon^2}{\delta}+\frac{\delta\varepsilon^3}{3(\delta-\varepsilon)^3}\right) \leq \frac{1}{\ln 2}\left(\frac{1}{2}+\frac{\delta^2\varepsilon}{3(\delta-\varepsilon)^3}\right)\frac{\varepsilon^2}{\delta}. \tag{10}$$

If we define the right hand side (10) to be $W(\varepsilon,\delta)$ and we substitute the previous bound in (6), we get

$$KL(P_0(x^{(t)} \mid x^{(<t)}), P_1(x^{(t)} \mid x^{(<t)})) \leq W(\varepsilon,\delta),$$

and combining the previous bound with (4), we can finally conclude that for any integer T, we have $KL(P_0^{(\leq T)}, P_1^{(\leq T)}) \leq T \cdot W(\varepsilon,\delta)$. Thus, from Theorem 18 and the latter bound, it follows that the error under a uniform prior of the source type, as defined in (3), is at least

$$\frac{1}{2}P_0\left(ft(\mathbf{X}^{(\leq t)})=1\right)+\frac{1}{2}P_1\left(f(\mathbf{X}^{(\leq t)})=0\right) \geq \frac{1}{2}-\frac{1}{2}\sqrt{KL(P_0^{(\leq T)},P_1^{(\leq T)})} \geq \frac{1}{2}-\frac{1}{2}\sqrt{T\cdot W(\varepsilon,\delta)}.$$

Hence, the number of samples T needs to be greater than $\frac{1}{9}\frac{1}{W(\varepsilon,\delta)} = \frac{\ln 2}{9}\left(\frac{6(\delta-\varepsilon)^3}{\delta^3-\delta^2\varepsilon+3\delta\varepsilon^2-\varepsilon^3}\right)\frac{\delta}{\varepsilon^2}$ to allow the possibility that the error be less than $1/3$.

In particular, if we assume that $10\varepsilon < \delta$, then we can bound $\frac{\delta^2\varepsilon}{3(\delta-\varepsilon)^3} \leq \frac{\delta^3}{10}\cdot\frac{1}{3(9/10)^3\delta^3} \leq \frac{100}{2187}$. It follows that (10) can be bounded with $W(\varepsilon,\delta) \leq \frac{1}{\ln 2}\left(\frac{1}{2}+\frac{100}{2187}\right) \leq 0.79$, and so $\frac{1}{9}\frac{1}{W(\varepsilon,\delta)} \geq 0.14\cdot\frac{\delta}{\varepsilon^2} = \Omega\left(\frac{\delta}{\varepsilon^2}\right)$. This completes the proof of Theorem 12 and hence of Theorem 9. ◀

References

1. A. El Gamal a. Young-Han Kim. *Network Information Theory*. Cambridge University Press, 2011.
2. M. Abeles. *Corticonics: Neural circuits of the cerebral cortex*. Cambridge University Press, 1991.
3. N. Alon, M. Braverman, K. Efremenko, R. Gelles, and B. Haeupler. Reliable communication over highly connected noisy networks. In *PODC*, pages 165–173, 2016.
4. F. Amor, P. Ortega, X. Cerdá, and R. Boulay. Cooperative prey-retrieving in the ant cataglyphis floricola: An unusual short-distance recruitment. *Insectes Sociaux*, 57(1), 2010. URL: `https://link.springer.com/article/10.1007/s00040-009-0053-x`.
5. D. Angluin, J. Aspnes, and D. Eisenstat. A simple population protocol for fast robust approximate majority. *Distributed Computing*, 21(2):87–102, 2008.
6. D. Angluin, J. Aspnes, M. J. Fischer, and H. Jiang. Self-stabilizing population protocols. *TAAS*, 3(4), 2008.
7. J. Aspnes and E. Ruppert. An introduction to population protocols. *Bulletin of the EATCS*, 93:98–117, 2007.
8. W. Bialek. Physical limits to sensation and perception. *Annual review of biophysics and biophysical chemistry*, 16(1):455–478, 1987.
9. S. Bikhchandani, D. Hirshleifer, and I. Welch. Learning from the behavior of others: Conformity, fads, and informational cascades. *J. Economic Perspectives*, 12(3):151–170, 1998.

10 Lucas Boczkowski, Amos Korman, and Emanuele Natale. Minimizing message size in stochastic communication patterns: Fast self-stabilizing protocols with 3 bits. In Philip N. Klein, editor, *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2017, Barcelona, Spain, Hotel Porta Fira, January 16-19*, pages 2540–2559. SIAM, 2017. doi:10.1137/1.9781611974782.168.

11 A. Cavagna, A. Cimarelli, I. Giardina, G. Parisi, R. Santagati, F. Stefanini, and M. Viale. Scale-free correlations in starling flocks. *PNAS*, 107(26):11865–11870, 2010.

12 Keren Censor-Hillel, Bernhard Haeupler, Jonathan A. Kelner, and Petar Maymounkov. Global computation in a poorly connected world: fast rumor spreading with no dependence on conductance. 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 961–970. ACM, 2012. doi:10.1145/2213977.2214064.

13 E. Chastain, A. Livnat, C. Papadimitriou, and U. Vazirani. Algorithms, games, and evolution. *PNAS*, 111(29):10620–10623, 2014. doi:10.1073/pnas.1406556111.

14 Bernard Chazelle. Natural algorithms. In *SODA*, pages 422–431, 2009.

15 Thomas M Cover and B Gopinath. *Open problems in communication and computation*. Springer Science & Business Media, 2012.

16 A. Demers, D. Greene, C. Hauser, W. Irish, J. Larson, S. Shenker, H. Sturgis, D. Swinehart, and D. Terry. Epidemic algorithms for replicated database maintenance. In *PODC*, 1987.

17 Benjamin Doerr, Leslie Ann Goldberg, Lorenz Minder, Thomas Sauerwald, and Christian Scheideler. Stabilizing consensus with the power of two choices. In Rajmohan Rajaraman and Friedhelm Meyer auf der Heide, editors, *SPAA 2011: Proceedings of the 23rd Annual ACM Symposium on Parallelism in Algorithms and Architectures, San Jose, CA, USA, June 4-6, 2011 (Co-located with FCRC 2011)*, pages 149–158. ACM, 2011. doi:10.1145/1989493.1989516.

18 A. Dornhaus and L. Chittka. Food alert in bumblebees (bombus terrestris): Possible mechanisms and evolutionary implications. *Behavioral Ecology and Sociobiology*, 50(6):570–576, 2001.

19 Robert Elsässer and Thomas Sauerwald. On the runtime and robustness of randomized broadcasting. *Theor. Comput. Sci.*, 410(36):3414–3427, 2009. doi:10.1016/j.tcs.2008.04.017.

20 O. Feinerman and A. Korman. Theoretical distributed computing meets biology: A review. In *ICDCIT*, pages 1–18. Springer, 2013.

21 O. Feinerman, A. Rotem, and E. Moses. Reliable neuronal logic devices from patterned hippocampal cultures. *Nature physics*, 4(12):967–973, 2008.

22 Ofer Feinerman, Bernhard Haeupler, and Amos Korman. Breathe before speaking: efficient information dissemination despite noisy, limited and anonymous communication. In Magnús M. Halldórsson and Shlomi Dolev, editors, *ACM Symposium on Principles of Distributed Computing, PODC '14, Paris, France, July 15-18, 2014*, pages 114–123. ACM, 2014. doi:10.1145/2611462.2611469.

23 P. Fraigniaud and E. Natale. Noisy rumor spreading and plurality consensus. In *PODC*, pages 127–136, 2016.

24 R. G. Gallager. Finding parity in a simple broadcast network. *IEEE Trans. Inf. Theor.*, 34(2):176–180, 2006.

25 L. A. Giraldeau, T. J. Valone, and J.J. Templeton. Potential disadvantages of using socially acquired information. *Philosophical Transactions of the Royal Society of London B: Biological Sciences*, 357(1427):1559–1566, 2002.

26 Navin Goyal, Guy Kindler, and Michael E. Saks. Lower bounds for the noisy broadcast problem. *SIAM J. Comput.*, 37(6):1806–1841, 2008. doi:10.1137/060654864.

27 B. Hölldobler. Recruitment behavior in camponotus socius (hym. formicidae). *J. of Comparative Physiology A*, 75(2):123–142, 6 1971.

28 R. M. Karp, C. Schindelhauer, S. Shenker, and B. Vöcking. Randomized rumor spreading. In *FOCS*, pages 565–574, 2000.

29 D. Kempe, A. Dobra, and J. Gehrke. Gossip-based computation of aggregate information. In *FOCS*, pages 482–491. IEEE, 2003.

30 Amos Korman, Efrat Greenwald, and Ofer Feinerman. Confidence sharing: An economic strategy for efficient information flows in animal groups. *PLoS Computational Biology*, 10(10), 2014. doi:10.1371/journal.pcbi.1003862.

31 S. Marras, R. Batty, and P. Domenici. Information transfer and antipredator maneuvers in schooling herring. *Adaptive Behavior*, 20(1):44–56, 2012.

32 Cameron Musco, Hsin-Hao Su, and Nancy A. Lynch. Ant-inspired density estimation via random walks: Extended abstract. In George Giakkoupis, editor, *Proceedings of the 2016 ACM Symposium on Principles of Distributed Computing, PODC 2016, Chicago, IL, USA, July 25-28, 2016*, pages 469–478. ACM, 2016. doi:10.1145/2933057.2933106.

33 B. Pittel. On spreading a rumor. *SIAM J. Appl. Math.*, 47(1):213–223, 1987.

34 N. Razin, J.P. Eckmann, and O. Feinerman. Desert ants achieve reliable recruitment across noisy interactions. *Journal of the Royal Society Interface; 10(20170079).*, 2013.

35 G. Rieucau and L. A. Giraldeau. Persuasive companions can be wrong: the use of misleading social information in nutmeg mannikins. *Behavioral Ecology*, pages 1217–1222, 2009.

36 P. Rigollet. High dimensional statistics. *Lecture notes for course 18S997.*, 2015.

37 S. B. Rosenthal, C. R. Twomey, A. T. Hartnett, H. S. Wu, and I. D. Couzin. Revealing the hidden networks of interaction in mobile animal groups allows prediction of complex behavioral contagion. *PNAS*, 112(15):4690–4695, 2015.

38 J. J. Templeton and Giraldeau L. A. Patch assessment in foraging flocks of european starlings: evidence for the use of public information. *Behavioral Ecology*, 6(1):65–72, 1995.

39 J. von Neumann. Probabilistic logics and the synthesis of reliable organisms from unreliable components. *Automata Studies*, pages 43–98, 1956.

40 A. Xu and M. Raginsky. Information-theoretic lower bounds for distributed function computation. *IEEE Trans. Information Theory*, 63(4):2314–2337, 2017.

41 Y.Afek, N. Alon, O. Barad, E. Hornstein, N. Barkai, and Z. Bar-joseph. A biological solution to a fundamental distributed computing problem. *Science*, 2011.

Making Asynchronous Distributed Computations Robust to Channel Noise*†

Keren Censor-Hillel[1], Ran Gelles[2], and Bernhard Haeupler[3]

1 **Department of Computer Science, Technion, Israel**
ckeren@cs.technion.ac.il
2 **Faculty of Engineering, Bar-Ilan University, Israel**
ran.gelles@biu.ac.il
3 **Department of Computer Science, Carnegie Mellon University, USA**
haeupler@cs.cmu.edu

Abstract

We consider the problem of making distributed computations robust to noise, in particular to worst-case (adversarial) corruptions of messages. We give a general distributed interactive coding scheme which simulates any asynchronous distributed protocol while tolerating a maximal corruption level of $\Theta(1/n)$-fraction of all messages. Our noise tolerance is optimal and is obtained with only a moderate overhead in the number of messages.

Our result is the first *fully distributed* interactive coding scheme in which the topology of the communication network is not known in advance. Prior work required either a coordinating node to be connected to all other nodes in the network or assumed a synchronous network in which all nodes already know the complete topology of the network.

Overcoming this more realistic setting of an unknown topology leads to intriguing distributed problems, in which nodes try to learn sufficient information about the network topology in order to perform efficient coding and routing operations for coping with the noise. What makes these problems hard is that these topology exploration computations themselves must already be robust to noise.

1998 ACM Subject Classification C.2.4 Distributed Systems, E.4 Coding and Information Theory

Keywords and phrases Distributed Computation, Coding for Interactive Communication, Noise-Resilient Computation, Coding Theory

Digital Object Identifier 10.4230/LIPIcs.ITCS.2018.50

1 Introduction

Fault tolerance is one of the central challenges in the design of distributed algorithms. Typically, computation is performed by n nodes, of which some subset may be *faulty* and not behave as expected. This includes *crash* or *Byzantine* failures. Faults can also occur as communication errors, if links suffer from, e.g., *omissions*, *alterations* or *Byzantine* errors (see, e.g., [31, 2]).

We focus on alteration errors, in which the content of sent messages may be corrupted. Previous work in the setting of faulty channels provides fault-tolerant algorithms either for

* Research supported in part by the Israel Science Foundation (grants 1696/14 and 1078/17), the Binational Science Foundation (grant 2015803), and NSF grants CCF-1527110 and CCF-1618280.
† A full version of this paper is available at [11], https://arxiv.org/abs/1702.07403.

specific tasks such as the leader election or the consensus problem (e.g., [36, 39, 24, 40]), or for a specific class of topologies (e.g., [32]).

In this paper, we provide a general technique that takes as an input an asynchronous distributed protocol over an arbitrary topology and outputs a simulation of this protocol that is resilient to noise. Specifically, we develop several tools whose combination allows us to obtain the first *fully distributed* interactive coding scheme.

The Challenge. In order to tolerate channel noise and preserve the correctness of computations, sophisticated coding techniques must be employed. Several works (e.g., [35, 28, 27, 1]) provide such coding schemes, however, they all assume that the network's topology is known in advance. We wish to challenge this assumption, and allow truly distributed coding schemes.

Interestingly, once communication is unreliable, even the simplest distributed tasks, such as flooding information over the network or constructing a BFS tree, become tremendously difficult to execute correctly. For instance, the asynchronous distributed Dijkstra or Bellman-Ford algorithms [33] miserably fail when messages may be corrupted. To see why, recall that in the Bellman-Ford algorithm, each node sends to all of its neighbors its distance from the root. A node then sets its neighbor that is closest to the root as its parent. However, if messages are incorrect, the distance mechanism may fail and nodes may set their parents in an arbitrary way.

Our Contribution. In any attempt to tolerate message corruptions, naturally, some bound on the noise must be given. Indeed, if a majority of the sent messages are corrupted, there is no hope to complete a computation correctly. On the other hand, when the noise falls below a certain threshold, fault tolerant computation can be obtained, for example, by employing various coding techniques.

The field of *coding for interactive communication* (see, e.g., the survey of [19]) considers the case where two or more parties carry some computation by sending messages to one another over noisy channels and strives to devise *coding schemes* with good guarantees. A coding scheme is a method that is given as an input a protocol π that assumes reliable channels, and outputs a noise-resilient protocol Π that simulates the communication of π. The two main measures upon which a coding scheme is evaluated are its *noise resilience* – the fraction of noise that the resilient simulation Π can withstand – and its *overhead* – the amount of redundancy Π adds in order to tolerate faults. For networks with n nodes, it is easy to show that the maximal adversarial noise fraction that any resilient protocol can cope with is $\Theta(1/n)$ [28]. Indeed, if more than $(1/n)$-fraction of the messages are corrupted, then the noise can completely corrupt all the communication of the node that sends the least number of messages. The overhead depends on the network topology, communication model, and noise resilience, as we elaborate upon in the Related Work section below.

Our main result, informally stated as follows, is a deterministic coding scheme that fortifies any asynchronous protocol designed for a noise-free setting over any network topology, such that its resilient simulation withstands the maximal $\Theta(1/n)$-fraction of noise.

▶ **Theorem 1.** *There exists a deterministic coding scheme that takes as an input any asynchronous distributed protocol π designed for reliable channels, and outputs an asynchronous distributed protocol Π that simulates π, is resilient to an optimal fraction of $\Theta(1/n)$ of adversarially corrupted messages, and has a multiplicative communication overhead of $O(n \log^2 n)$.*

Our coding scheme introduces a multiplicative overhead of $O(n \log^2 n)$; no other results are known for this model. This overhead should be compared with the state of the art coding scheme by Hoza and Schulman [27]. Their scheme applies to *synchronous* networks where the

topology is *known by all the parties* in advance, and achieves an overhead of $O((|E| \log n)/n)$, where $|E|$ is the number of communication links in the network. Setting the optimal coding overhead in the *asynchronous* model over an arbitrary topology (*unknown to the parties*) remains as an open question.

1.1 Techniques

1.1.1 A Content-Oblivious BFS Construction

A key ingredient in our coding scheme is a BFS construction which is *content oblivious*. That is, in our BFS construction, the nodes send messages to each other and *ignore their content*, basing their decisions only on the order of received messages. The challenge is to be able to do this despite asynchrony and despite lack of FIFO assumptions. In a sense, our construction can be seen as a variant of the distributed Dijkstra algorithm, with the property that the nodes send "empty messages" that contain no information (alternatively, the nodes ignore the content of received messages).

Recall that the distributed Dijkstra algorithm, see, e.g., [33, Chapter 5], is initiated by some node r,[1] which governs the BFS construction layer by layer, where the construction of each layer is called a *phase*. The invariant is that after the p-th phase, the algorithm has constructed a BFS tree T_p of depth p rooted at r, where all nodes in T_p know their parent and children in T_p. The base case is $T_0 = \{r\}$, and the construction of the first layer is as follows. The node r sends an Explore message to all its neighbors, who in turn set r as their parent. Each Explore message is replied to with an Ack message. Once r receives Ack messages from all of its neighbors, the first phase ends and the construction of the second layer begins. Note that T_1 indeed holds r and all of its neighbors.

For the p-th phase, the root floods a message Phase through T_{p-1}. Once a leaf in T_{p-1} receives a Phase message, it sends Explore to all of its neighbors, who in turn set their parent unless already in T_{p-1}. Each node that receives an Explore replies with an Ack and an indication of its parent node, so that the exploring node learns which of its neighbors is a child and which is a sibling. Upon receiving an Ack from all of its neighbors, the node sends an Ack to its parent, which propagates it all the way to r. Once r has received Ack messages from all of its children, the phase is complete.

Our content-oblivious BFS construction imitates the above behavior while using only a *single* type of message, instead of Phase, Explore and Ack messages. Specifically, the construction begins with r sending a message (Explore[2]) to all of its neighbors, who in turn set r as their parent and reply with a message (Ack). When r receives a message from all of its neighbors, the first phase is complete. Then, r begins the second phase by sending another message (Explore/Phase) to all of its neighbors. This message causes a node that has already set its parent to behave like r – it sends a message to all of its neighbors (Explore) *except for its parent*. After receiving a message (Ack) from all of its neighbors, it sends a message (Ack) to its parent.

One can easily verify that this approach behaves similarly to the Dijkstra algorithm described above, in the sense that every node sets its parent correctly. The only difference is when a node u sends an (Explore) message to its sibling w. In the Dijkstra algorithm

[1] In all the protocols we discuss, the root node does not need to be identified in advance. Rather, the algorithm initiates by waking up an *arbitrary* node who will act as the root. From this point an on, nodes wake up when receiving a message from a neighbor.

[2] To ease the readability, we write in parenthesis the functionality of each sent message, but we emphasize that messages in our construction contain no content at all, and the labels of Explore and Ack are given only for the analysis.

the sibling w replies by telling the exploring node u that they are siblings (by indicating the parent of w, which is not u). However, in our case messages contain no content and u is unable to distinguish whether w is a child or a sibling, since in both cases w should reply to the EXPLORE message in the same way.

Our insight is that serializing each phase provides a solution to the above ambiguity. That is, we let r send a message (EXPLORE/PHASE) to one child at a time, waiting to receive a message (ACK) from that child before sending a message (EXPLORE/PHASE) to the next child. This gives that if a node is expecting a message (EXPLORE) from its parent but instead it receives a message (EXPLORE) from a non-parent neighbor, then it knows that this neighbor must be a sibling. Hence, the node can mark all siblings and distinguish them from its children.

The main advantage of not basing our construction on the content of received messages is that the obtained BFS construction is *inherently tolerant against message corruptions*: the noise has no effect on the construction since the content of the communicated messages is already being ignored. Notice that in our construction, the nodes do not learn their distance from r, in contrast to what can easily be obtained in the noise-free case. However, this will suffice for our usage of the BFS tree in our coding scheme.

1.1.2 Interactive Coding over Sparse Subgraphs

A crucial framework we rely on in our simulation is a multiparty coding scheme for interactive communication by Hoza and Schulman [27], which is in turn based on ideas from [35]. This coding scheme allows simulating protocols over any graph $G = (V, E)$ and withstands an $O(1/|E|)$-fraction of adversarial message corruption, while incurring a *constant* communication overhead. The caveat of using this scheme for our simulation is that it applies only for *synchronous* protocols that communicate over G in a manner which we call *fully-utilized synchronous*: in each round, every node communicates one symbol over to *each* of its neighbors.

In order to obtain our coding scheme for asynchronous protocol with resilience $\Theta(1/n)$, we first convert the asynchronous input protocol π into a fully-utilized synchronous protocol defined over some subgraph $G' = (V, E')$ of G with $|E'| = \Theta(n)$. To this end, we use the BFS tree constructed by our content-oblivious method described above. Once we obtain a BFS tree $\mathcal{T}$, we simulate each message communicated by π via n fully-utilized synchronous rounds over the tree $\mathcal{T}$. During each of such n rounds, a message of π is flooded throughout $\mathcal{T}$ until it reaches all the nodes and, in particular, its destination node. Note that in every round, all nodes send messages over all the edges of $\mathcal{T}$. This implies a communication overhead of $O(n^2 \log n)$: we have n rounds with $O(n)$ messages per round. The $\log n$ term stems from adding the identity of the source node and the destination node to each flooded message.[3]

Using the Hoza and Schulman [27] coding scheme taking as an input the fully-utilized synchronous protocol defined over the topology $\mathcal{T}$ gives a resilient simulation of π which withstands a maximal $\Theta(1/n)$-fraction of corrupted messages. Alas, it is a synchronous simulation, while our environment is asynchronous. Hence, to complete our simulation, we need to use a *synchronizer* [3].

1.1.3 A Root-Triggered Synchronizer

In the original error-free setting, if the input protocol to a synchronizer is guaranteed to be fully-utilized then synchronization is trivial. Each node simply attaches a round number to each of its outgoing messages and produces the outgoing messages for round $i + 1$ only after

[3] Throughout this work, all logarithms are taken to base 2.

receiving messages for round i from all of its neighbors. The key difficulty is then for non-fully utilized synchronous input algorithms, in which a node cannot simply wait to receive a message for round i from all of its neighbors, as it may be the case that some of these do not exist.

In our setting, we guarantee that we produce a fully-utilized synchronous algorithm as an input to our synchronizer. However, we do not assume FIFO channels, which means that we cannot rely on the naive synchronizer, despite the promise of a fully-utilized synchronous protocol for an input. Thus, we need a different solution for synchronizing the messages, and our approach is based on having a single node responsible for triggering messages of each round only after the previous round has been simulated by all nodes. To this end, our synchronizer bears similarity to the classic tree-based synchronizer of Awerbuch [3], with the difference that it does not incur any message overhead because it is given a fully-utilized synchronous input.

1.1.4 A Spanner-Based Coding Scheme

Finally, we show how to further improve the communication overhead of our coding technique. Routing each message over a tree $\mathcal{T}$ requires n rounds in the worst case for a message to reach its destination. A more efficient solution would be to route each message through a spanning subgraph $S = (V, E_S)$ of G in which the distance over S of every $(u, v) \in E$ is not too large. On the other hand, the Hoza-Schulman coding scheme on S has a noise resilience of $\Theta(1/|E_S|)$, and hence we require $|E_S|$ to be $O(n)$ in order to maintain an optimal resilience level of $\Theta(1/n)$. Luckily, for every G there exist sparse spanning subgraphs in which $|E_S| = O(n)$ while every two neighbors in G are at distance at most $O(\log n)$ in S; such subgraphs are known as $O(\log n)$-spanners [33, 34].

Flooding a message of π from u to v can be done within $O(\log n)$ rounds, in each of which $O(|E_S|) = O(n)$ messages are sent by a fully-utilized synchronous simulation of π, leading to our claimed communication overhead of $O(n \log^2 n)$. Here again, the extra $\log n$ term stems from adding identifiers to each flooded message.

However, flooding information over a spanner introduces several other difficulties. For instance, in contrast to the case of a tree, it is not guaranteed anymore that each message arrives only once to its destination – indeed, multiple paths may exist between any two nodes. Furthermore, when multiple nodes send messages, the congestion may cause super-polynomial delays if a simple flooding algorithm is used. Then, due to having multiple paths with arbitrary delays, messages may arrive to their destination out of order. Since the delay is super-polynomial in the worst case, adding a counter to each message increases the overhead by $\omega(\log n)$ and damages the global overhead.

Instead, we provide a contention-resolution flavored technique, which consists of priority-based windows for delivering the messages. In more detail, a message flooding starts only at the beginning of an $O(\log n)$-round window. Multiple messages that are sent during the same window may be dropped during their flooding, yet the source always learns when its message is dropped, so it can retransmit the message in the next window. A similar approach is well-known for constructing a BFS tree when no specific root is given, but our extension of this technique is more involved, since dropped messages *must be resent*.

It remains to explain how to construct the $O(\log n)$-spanner over the noisy network to begin with. For this, we use our previously described tree-based coding scheme to simulate a distributed spanner construction, e.g., the (noiseless) construction of Derbel, Mosbah, and Zemmari [13]. While coding this part incurs a large overhead of $O(n^2 \log n)$, this overhead applies only to the part of constructing the spanner, and the global overhead of our coding scheme is dominated by the overhead of coding the input protocol over the spanner.

1.2 Related Work

Performing computations over noisy channels is the heart of *coding for interactive communication*, initiated by Schulman [37, 38]. A long line of work considers the 2-party case in various settings and noise models [10, 6, 23, 17, 14, 29, 25, 20, 7, 9]. See [19] for a survey.

Interactive coding in the multiparty setting was first considered by Rajagopalan and Schulman [35] for the case of stochastic noise. For any topology G, they show a coding scheme with an overhead of $O(\log(d+1))$, where d is the maximal degree of G. Gelles et al. [22] provide an efficient extension to that scheme. Alon et al. [1] show a coding scheme with an overhead of $O(1)$ for d-regular graphs with degree $d = n^{\Omega(1)}$. Braverman et al. [8] demonstrate a lower bound of $\Omega(\log n)$ on the communication over a star graph. All the above works assume fully-utilized synchronous protocols, in which in every round all nodes communicate on all the channels connected to them. Gelles and Kalai [21] show that if nodes are not required to speak at every round, a lower bound of $\Omega(\log n)$ on the overhead can be proved even for graphs with a small degree, e.g., $d = 2$.

Jain et al. [28] show a multiparty coding scheme resilient to an *adversarial* noise fraction of $\Theta(1/n)$ with constant overhead, assuming a topology that contains a star as a subgraph. Lewko and Vitercik [30] improve the communication balance of that scheme. Hoza and Schulman [27] consider fully-utilized synchronous protocols on arbitrary graphs and show a coding with resilience $\Theta(1/|E|)$ and overhead $O(1)$. If the topology of G is known to all nodes, they can route information through a sparser spanning graph with $O(n)$ edges. In this case, they show a coding scheme with an optimal resilience level of $\Theta(1/n)$ and an overhead of $O((|E|\log n)/n)$.

Previous work in distributed settings that allow edge failures are typically different from our setting in various aspects. Most notable are synchrony assumptions, complete communication graphs or addressing specific distributed tasks [36, 39, 24, 40, 12]. Assumptions regarding the noise include random link corruptions [32, 5, 15], or a given bound on the number of links that may exhibit failures [32, 24, 40]. This is in contrast to our work, which addresses an asynchronous setting with an arbitrary topology, and considers the simulation of any distributed task where there is no bound on the number of faulty links. In particular, *all* links may send corrupted messages, with the bound being the number of corruptions rather than the number of faulty links.

Synchronizers for unreliable settings have been studied in [4], which addresses a dynamic setting, and in [26], which assumes faulty nodes.

2 Preliminaries

Throughout this work we assume a network described by a graph $G = (V, E)$ with $n = |V|$ nodes and $m = |E|$ edges. Each node $u \in V$ is a party that participates in the computation and each edge $(u, v) \in E$ is a bi-directional communication channel between nodes u and v. The task of the nodes is to conduct some distributed computation given by a deterministic[4] protocol π, which consists of the algorithm each node (locally) runs. In particular, the protocol dictates to each node which messages to send to which neighbor as a function of all previous communication (and possibly the node's identity, private randomness and private input, if exists). The *communication complexity* of the protocol, $\mathsf{CC}(\pi)$, is the maximal

[4] While we focus here on deterministic protocols, ours result also apply to randomized Monte-Carlo protocols.

number of bits communicated by all nodes in any instance of π. The *message complexity* of π is the maximal number of message sent by all nodes in any instance of π.

We assume that the topology of G is known only locally, namely, each node v knows only the set $\mathcal{N}_u$ of identities of its own neighbors. However, the size of the network n is known to all nodes.

Communication Models. Our protocols are for the *Asynchronous* communication model defined below. In addition, we describe a different communication model named the *Fully-Utilized Synchronous Model*, which is common in previous interactive coding work [35, 27, 1, 8]. In particular, we use coding schemes defined in the fully-utilized synchronous model (specifically, [27]) as primitives for encoding our asynchronous protocols (see Lemma 2 below).

- *Asynchronous Model.* In this setting, there are no timing assumptions. We assume each node is asleep until receiving a message. Once a message is received, the receiver wakes up, performs some local computation, transmits one or more messages to one or more adjacent nodes and goes back to sleep. Messages can be of any length. A protocol starts by waking up a single node r of its choice.
- *The Fully-Utilized Synchronous Model.* Communication in this model works in synchronous *rounds*, determined by a global clock. At every clock tick, every node sends one symbol (from some fixed alphabet Σ) on each and every one of the communication links connected to it. That is, at every round exactly $2m$ symbols are being communicated.

Adversarial Channel Noise. We assume an all-powerful adversary that knows the network G, the protocol π and the private inputs of the nodes (if there are any). The adversary is able to (a) corrupt messages by changing the content of a transmitted message and (b) rush or delay the delivery of messages by an unbounded but finite amount of time. We restrict the number of messages that the adversary can corrupt, namely, we assume that the adversary can corrupt at most some fixed fraction μ of the communicated messages. We do not restrict how a message can be corrupted and, in particular, the adversary may replace a sent message M with any other message M' of any length and content. However, our coding scheme will have the invariant that each message contains a single symbol (from a given alphabet Σ), thus a message corruption will be equivalent to corrupting a single symbol. Note that the adversary is *not* allowed to inject new messages or completely delete existing messages.[5]

Protocol Simulation, Resilience, and Overhead. A protocol Π is said to *simulate* π, if after the completion of Π, each node outputs the transcript it would have seen when running π assuming noiseless channels. The protocol Π is *resilient* to a μ fraction of noise, if Π succeeds in simulating π even if an all powerful adversary completely corrupts up to a fraction μ of the messages communicated by Π. The *overhead* of Π with respect to π is defined by $\mathit{overhead}(\Pi \mid \pi) = \mathsf{CC}(\Pi)/\mathsf{CC}(\pi)$.

A coding scheme $\mathcal{C} : \pi \to \Pi$ converts any input protocol π into a resilient version $\Pi = \mathcal{C}(\pi)$. The resilience of a coding scheme is the minimal resilience of any simulation generated by the coding scheme. The (asymptotic) overhead of a coding scheme considers the maximal overhead for the worst input protocol π when $\mathsf{CC}(\pi)$ tends to infinity. Namely,

$$\mathit{overhead}(\mathcal{C}) = \limsup_{c\to\infty} \max_{\substack{\pi \text{ s.t.} \\ \mathsf{CC}(\pi)=c}} \mathit{overhead}(\mathcal{C}(\pi) \mid \pi).$$

[5] This type of noise, commonly called, *insertion and deletion* noise is known cause issues of synchronization in the interactive setting [9] and may be destructive for asynchronous protocols [16].

We are mainly interested in how the overhead scales with n and m.

A famous multiparty coding scheme in the fully-utilized synchronous model, shown by Hoza and Schulman [27] (based on a previous scheme [35]), provides a coding scheme that simulates any noiseless fully-utilized synchronous protocol π defined over some topology G with resilience $\Theta(1/m)$ and a constant overhead $O(1)$.

▶ **Lemma 2** ([27]). *In the fully-utilized synchronous model, any T-round protocol π can be simulated by a protocol $\Pi = \mathrm{HS}(\pi)$ with round complexity $O(T)$ and communication complexity $O(\mathsf{CC}(\pi))$ that is resilient to adversarial corruption of up to an $\Theta(1/m)$ fraction of the messages.*

3 A Distributed Content-Oblivious BFS Algorithm

In this section we show a distributed construction of a BFS tree using messages whose content can be arbitrary. We call this a *content-oblivious* construction. Our algorithm can be seen as a variant of a simple distributed layered-BFS algorithm, see, e.g., [18, 33, 41].

3.1 The BFS Algorithm: Description

The BFS construction is initiated by one designated node r we call here the *root*. The construction builds the tree layer by layer. First, the root sends a message to all of its neighbors. This triggers its neighbors to set r as their parent. Each such a neighbor replies a message to r to acknowledge that it has received r's message. Once r has received a message from all of its neighbors, it knows that the first layer is completed, and all nodes with distance 1 have set r as their parent. We call the above an Explore step.

The root then begins a second Explore which causes all nodes at distance 2 to set their parent and connect to the BFS tree. Specifically, the root sends a message to each of its children and waits until all children reply a message to indicate they are done. However, in contrast to previous distributed BFS algorithms, messages are sent *sequentially* – the root sends a message to its next child only after receiving the acknowledgement message from its previous child.

When a node v that has already set its parent $parent_v$ receives a message *from its parent* $parent_v$, it acts as a root and invokes an Explore: it sends a message to all of its neighbors excluding $parent_v$ and waits until they all send a message back. Only then v sends a message to its parent to indicate its Explore process has completed. It is easy to see that when the root completes its k-th Explore, all nodes within distance at most k have set their parent and connected to the BFS tree.

A special treatment is needed when a node u receives a message from a node v who is *not* the parent of u during a time at which u is not in the middle of an Explore step. That is, u is not expecting any messages from its neighbors, except for its parent that may trigger it to initiate another Explore step. Recalling that messages are sent to children in a sequential manner, it is easy to verify that such a message delivery may happen only when v has received a message from its own parent and is now processing its own Explore. That is, such a message indicates that v is *a sibling* of u in the BFS tree (namely, v is not a parent nor a child of u in the BFS tree). Thus, upon receiving such a message, u marks v as a sibling and removes it from its list of children. To simplify the presentation, as we elaborate in Remark 1, in next exploration steps u will keep sending messages to v as if it was one of its children.

One additional property that we require from our BFS construction is that all the nodes complete the algorithm *at the same time*. As explained in the introduction, we use this construction as an initial part for our coding scheme. Furthermore, recall that in order to be

Algorithm 1(a) Content-oblivious BFS construction: Main Algorithm

Initialization: All nodes begin in the INIT state.

For node r designated as root:
Begin
  $parent_r \leftarrow \perp$
  $children_r \leftarrow \mathcal{N}_r$
  $count_r \leftarrow 0$
  $state_r \leftarrow$ IDLE

  while $state_r \neq$ DONE **do** ▷ Perform n instances of Explore
    r invokes Explore
  end while
End

noise-resilient, during the BFS construction the nodes ignore the content of the messages and their entire behavior is based on whether or not a message was received. However, once this construction is complete, the nodes send and receive messages according to the coding scheme and it is crucial that a node is able to distinguish messages that belong to the BFS construction from messages of the coding scheme.

We solve this issue by making sure that each node participates in exactly n steps of Explore. Once the node has sent the n-th acknowledgement to its parent, the node knows that the next message *from the parent* belongs to the coding scheme rather than to the BFS construction.[6] To make sure that each node participates in exactly n Explore steps, regardless of its distance from r, we let every node initiate one additional Explore, which we refer to as a *dummy* Explore. Specifically, when a node completes its $(n-1)$-th Explore, and *before the node sends the acknowledgement back to its parent*, it invokes another Explore step. Now, just by counting the messages received from the parent, every node knows whether the BFS construction has completed or not.

The pseudocode of the BFS construction is given in Algorithm 1(a) and Algorithm 1(b).

3.2 The BFS Algorithm: Analysis

In this section we analyze Algorithm 1 and show that it satisfies the following properties.

▶ **Theorem 3.** *For any input $G = (V, E)$ and node $r \in V$, Algorithm 1 finds a BFS tree $\mathcal{T}$ with root r. Specifically, each node knows its parent in T and all of its adjacent edges that belong to $\mathcal{T}$. The algorithm communicates $O(nm)$ messages, where no payload is needed in any messages.*

Furthermore, we show that all nodes know that the BFS construction is complete, in the following sense.

▶ **Claim 4.** *At the end of Algorithm 1 all nodes are in state* DONE. *Moreover, if r is in state* DONE *then all other nodes are in state* DONE *as well.*

[6] Note that additional messages may arrive from a sibling node for the BFS construction but still, the next message arriving from the *parent* belongs to the coding scheme rather than the BFS construction.

Algorithm 1(b) Content-oblivious BFS construction: Message Handling Procedures

For every node u in state INIT upon receiving a message from node v
procedure SetParent
  $parent_u \leftarrow v$
  $children_u \leftarrow \mathcal{N}_u \setminus \{v\}$
  $count_u \leftarrow 0$
  $state_u \leftarrow$ IDLE
  send a message to v ▷ an "ACK" message
end procedure

For every node u in state IDLE/DONE upon receiving a message from $v \neq parent_u$
procedure MarkSibling
  $children_u \leftarrow children_u \setminus \{v\}$
  send a message to v ▷ an "ACK" message
end procedure

For every node u in state IDLE upon receiving a message from $parent_u$
procedure Explore
  $state_u \leftarrow$ EXPLORE
  $count_u \leftarrow count_u + 1$
  for all $v \in \mathcal{N}_u \setminus \{parent_u\}$ **do** ▷ note: **for** is sequential
    send a message to v ▷ an "Explore" message
    wait until a message is received from v
  end for

  if $count_u = n - 1$ **then** ▷ Extra *dummy* Explore
    for all $v \in children_u$ **do**
      send a message to v
      wait until a message is received from v
    end for
  end if

  send a message to $parent_u$ ▷ an "ACK" message

  if $count_u = n - 1$ **then** ▷ Change state to DONE if completed; otherwise, back to IDLE
    $state_u \leftarrow$ DONE
  else
    $state_u \leftarrow$ IDLE
  end if
end procedure

Proof of Theorem 3. Let $\mathcal{T}$ be a graph on the nodes V defined at the end of Algorithm 1 in the following manner: If $v = parent_u$, then (u, v) is an edge in $\mathcal{T}$. We begin by proving that $\mathcal{T}$ is a spanning tree. This is implied by the following claim.

▶ **Claim 5.** *At the end of the k-th invocation of the root's* Explore *step, all the nodes that are at distance k from r set their parent to a node with distance $k-1$ from r and move to the state* IDLE*, and every node of distance larger than k from r is in state* INIT*.*

Proof. We prove the claim by induction on k. The base case $k = 1$ follows since in the first Explore invocation all of r's children run SetParent, setting r as their parent, and switch to IDLE. They send message only back to r, hence all other nodes remain in INIT.

Assume that the claim holds for the k-th invocation and consider the $(k+1)$-th invocation of Explore by r. Messages propagating along the BFS tree cause all nodes of distance at most

k to invoke Explore (in some order). This triggers a message to every node of distance $k+1$, which causes it to switch its state to IDLE and set its parent to the first node (of distance k) that sent it a message. Note that nodes of distance $k+1$ only communicate back to their parent and do not invoke Explore at this time, so nodes of distance larger than $k+1$ remain in state INIT. At the end of the invocation each Explore, the invoking node switches back to state IDLE. ◀

Next, we prove that each node learns which neighbors are its children and which are not. Assume (u, v) is an edge in G but not in $\mathcal{T}$. We show that at the end of the algorithm $v \notin \mathit{children}_u$ and $u \notin \mathit{children}_v$. Let t be the first time after which both u and v have invoked SetParent. We claim that both u and v invoke Explore after time t. This is because time t is within the execution of an Explore step invoked by r and before Line 19 of that execution, and hence for every node $w \neq r$ there is a time $t_w > t$ during the execution of the loop in Lines 19–23 for r in which w invokes Explore.

Finally, we note that since (u, v) is an edge in G but not in $\mathcal{T}$, then neither u is an ancestor of v in $\mathcal{T}$ nor v is an ancestor of u in $\mathcal{T}$. This implies that when v invokes Explore then u is in state IDLE, which causes it to invoke MarkSibling and hence $v \notin \mathit{children}_u$. The proof for $u \notin \mathit{children}_v$ is exactly the same.

Finally let us analyze the message complexity. In Algorithm 1 each node invokes Explore for n times (see also the proof of Claim 4 below), where during each Explore it sends a message on each edge. Therefore, there are $O(n)$ messages sent on each one of the m edges, which amounts to a total message complexity of $O(nm) = O(|V| \cdot |E|)$. ◀

▶ Remark 1. It is possible to reduce the message complexity by sending Explore messages only to $\mathit{children}_v$ nodes. However, this must be delayed at least one Explore step, beyond the point in time where all the neighbors have completed their first Explore (in order to be able to identify siblings). The new message complexity will be $O(|V|^2 + |E|)$. For simplicity, we avoid this optimization and assume Explore messages are sent to all non-parent nodes all the time, incurring a message complexity of $O(|V| \cdot |E|)$.

We now prove Claim 4. This property is important in particular for the next section, as it suggests that there is a point in time (known by the root), when all nodes have completed their BFS algorithm. In hindsight, this allows to distinguish messages that are part of the BFS construction, whose content is ignored, from messages of the coding scheme, whose content is meaningful and must not be ignored.

Proof of Claim 4. Note that the Explore procedure works in an DFS manner: a node replies an ACK to its parent only after all of its children reply an ACK to it. Similarly, the root completes an Explore step after receiving an ACK from all its children, which means that they have all completed their Explore steps.

Note that each node invokes exactly n Explore steps due to the dummy Explore step initiated in Line 19. To see this, consider the same algorithm without the extra Explore in Lines 19–23 and note that nodes at distance k from the root r invoke exactly $n-k$ Explore steps. Adding this extra Explore step at every node makes all nodes invoke Explore exactly n times. Specifically, during the n-th invocation of Explore by r, every node with distance 1 from r invokes its $(n-1)$-th Explore step, and then, *before sending an ACK to* r in Line 30, it invoke its n-th Explore step. This then continues in an inductive manner all the way to the leaves.

Only once all of its children have sent an ACK and thus terminated the protocol and switched to DONE, a node replies with an ACK to its parent and changes its state to DONE. It follows that when the root receives an ACK for the n-th Explore step from all of its children, all the nodes have terminated the protocol and switched state to DONE. ◀

4 A Distributed Interactive Coding Scheme

In this section we show how to simulate any asynchronous protocol over a noisy network whose topology is unknown in advance. Our main theorem for this part is the following.

▶ **Theorem 6.** *Any asynchronous protocol π over a network G can be simulated by an asynchronous protocol Π resilient to an $\Theta(1/n)$-fraction of adversarial message corruption, and it holds that $\mathsf{CC}(\Pi) = O(nm \log n) + \mathsf{CC}(\pi) \cdot O(n^2 \log n)$.*

4.1 A fully-utilized synchronous protocol from an asynchronous input protocol π

The first ingredient we need is a way to transform an asynchronous protocol (defined over G) into a fully-utilized synchronous protocol defined over a given spanning tree $\mathcal{T}$ of G.[7] This is done in order to be able to use the Hoza-Schulman coding scheme. This transformation does not need to be robust to noise, as it is not going to be executed as is, but we will rather encode the fully-utilized synchronous protocol and execute the robust version. Later, we transform it back into the asynchronous setting using a synchronizer that is robust to noise.

Recall that in a fully-utilized synchronous protocol nodes operate in rounds, where at each round every node communicates one symbol (from some fixed alphabet Σ) on each communication channel connected to it. We will assume the alphabet is large enough to convey all the information that our coding scheme needs. In particular, we assume each symbol contains $O(\log n)$ bits.

▶ Remark 2. In the following, we assume the network G is composed of channels with a fixed alphabet Σ of size $\text{poly}(n)$. That is, each symbol contains $O(\log n)$ bits.

In order to avoid confusion, we will use the term "symbols" for messages sent by the coding scheme, while using "messages" to indicate the information sent by the noiseless protocol π.

The construction of our transformation into a fully-utilized synchronous protocol is given in Algorithm 2. In this construction, each node u maintains a queue of symbols that it needs to relay throughout a locally known spanning tree $\mathcal{T}$. The queue is initialized with the bits of any message that u needs to send according to the input protocol π, where each bit is encapsulated in a symbol that contains the bit value, the identity of the source (i.e., of u), and the identity of the destination node. Every symbol received by u is pushed into its queue, and relayed to u's neighbors in future rounds. In particular, upon receiving the symbol $(src, dest, val)$ from a node w, the node u pushes the vector $(src, dest, val, w)$ to its queue. If u is the destination node, it does not push the symbol into its queue; instead, u collects this bit for decoding the message.

The transformation works by having each node pop a record from its queue in each round and send the obtained triplet to all of its neighbors in $\mathcal{T}$ except for the node w from which the message was received. If the queue is empty then an empty message is sent to all neighbors in $\mathcal{T}$.

Note that all fragments of a message are received in order at the destination, since $\mathcal{T}$ has no cycles. Therefore, we can assume that the protocol sends a predefined symbol that indicates the end of the message, in order to avoid an assumption of knowledge of the message length. This ensures that Line 17 is well-defined. Our transformation guarantees the following.

[7] The spanning tree $\mathcal{T}$ used here will be later constructed using our content-oblivious BFS construction.

Algorithm 2 Simulating an asynchronous protocol π by a fully-utilized synchronous protocol π'.

Initialization: Given is a BFS tree $\mathcal{T}$ rooted at r.

In every round, for every node u:
Begin
  for every node v **do**
    Let $M_1 \cdots M_\ell$ be the bit representation of a message M that u has to send to v in π.
    Push $(u, v, M_1, \perp), \cdots, (u, v, M_\ell, \perp)$ into $queue_u$
  end for
  $(src, dest, val, w) \leftarrow$ pop item out of $queue_u$
  if $(src, dest, val, w)$ is not empty **then**
    send $(src, dest, val)$ to every $v \in \mathcal{N}_u(\mathcal{T}) \setminus \{w\}$ and send $\perp$ to w
  else
    send $\perp$ to every $v \in \mathcal{N}_u(\mathcal{T})$
  end if
  For every message $(src, dest, val)$ received from w:
  if $dest \neq u$ **then**
    push $(src, dest, val, w)$ into $queue_u$
  else
    collect the bits val for decoding M
  end if
End

▶ **Lemma 7.** *Algorithm 2 creates a fully-utilized synchronous protocol π' that simulates π, in the sense that all messages of π are sent and received. The simulation π' has a communication overhead of $O(n^2 \log n)$ with respect to π, and a message complexity of $\mathsf{CC}(\pi) \cdot O(n^2)$.*

Proof. By construction, every node sends a symbol to all of its neighbors in each round and hence Algorithm 2 is a fully-utilized synchronous protocol. In addition, eventually every messages of π reaches its destination and hence the obtained fully-utilized synchronous protocol simulates π. For the communication overhead, note that $O(\log n)$ bits of the identities of source and destination are appended to each bit sent by π; that is, a symbol size of $O(\log n)$ bits suffices. In addition, a delivery of a single message of π may require $O(n)$ rounds of relaying symbols sent along the tree $\mathcal{T}$. In each such round there are $O(n)$ symbols that are sent since the obtained protocol is a fully-utilized synchronous protocol. This implies that $O(n^2)$ symbols are communicated per each bit of π and gives a total communication overhead of $O(n^2 \log n)$.

Note that this is a worst-case analysis that assumes a single bit travels within the network at each time so that another bit is sent only after a previous bit reached its destination. If several bits are sent consecutively or if several nodes send bits simultaneously, the resulting number of messages can only decrease. ◀

4.2 Root-triggered synchronizers

We now describe our root-triggered synchronizer, which we use in order to execute the resilient synchronous protocol (which can be obtained by using the Hoza-Schulman coding scheme) in our asynchronous setting. We constructed a tree-based synchronizer as in Awerbuch [3].

The synchronizer gets as an input a fully-utilized synchronous protocol Π' and outputs an equivalent asynchronous protocol Π that simulates Π' round by round.

We first describe our simulation of a single round of Π' over a *tree*. Our synchronizer works as follows. The protocol begins by waking up an arbitrary node; denote this node as the root. The root initiates the process by sending its messages, determined by Π', to its children. This triggers its children to send their messages to their children, but not yet to their parent, and so forth, so that messages propagate all the way to the leaves. Once a leaf receives a message, it sends its message to its parent, and similarly, any node which receives a message from all of its children sends its message to its parent. This continues inductively all the way back to the root, which eventually receives messages from all of its children and complete the simulation of this round of Π'.

We build upon the above idea in order to simulate a fully-utilized synchronous algorithm Π' over an arbitrary graph S. That is, each node u has a message m_{uv} designated to each one of its neighbors $v \in \mathcal{N}_u(S)$. [8] The pseudocode is given in Algorithm 3. We single out a node r, which we refer to as the *initiator*, which starts by sending a message to all of its neighbors in S. This triggers each neighboring node to send its messages to its neighbors, but not yet to its parent, which is now simply the neighbor from which it receives the *first* message. This continues inductively, and only when a node receives messages from all of its neighbors it sends its message to its parent. Eventually, the initiator receives messages from all of its neighbors and completes the simulation of the round.

We prove the following properties of Algorithm 3.

▶ **Lemma 8.** *By the end of Algorithm 3 each node u receives the messages m_{vu} from every node $v \in \mathcal{N}_u(S)$, and all nodes are in state* DONE.

Proof. Let T denote the tree rooted at r that is induced by the edges of S that connect each node u with $parent_u$. By construction, each node $u \neq r$ sets its parent to be the first node from which it receives a messages and hence u sets exactly one node as its parent in an acyclic manner, inducing the tree T.

We prove by induction on the height of the nodes with respect to T, that each node u receives the messages m_{vu} from every node $v \in \mathcal{N}_u(S)$ and then switches its state to DONE. Note that every node sends its messages to all of its neighbors so that eventually all such messages arrive, and we only need to verify that the message from u to $parent_u$ is eventually sent.

The base case is for the leaves of T, which indeed receive messages from all of their neighbors since the only messages that get delayed are messages from nodes to their parents. Assume this holds for all nodes at height h, and consider a node u at height $h+1$. Node u receives messages from all of its siblings in the tree. By the induction hypothesis, every child v of u in T receives all of its messages and switches to state DONE. This implies that in between, node v sends its message m_{vu} to its parent u. When this happens for all nodes $v \in children_u$ it is the case that u receives the messages m_{vu} from every node $v \in \mathcal{N}_u(S)$ and then switches its state to DONE. ◀

By having the initiator r control the simulation of each round of a simulated fully-utilized synchronous protocol Π', we obtain synchronization for an arbitrary number of rounds.

[8] Later, in Section 5, we apply our root-triggered synchronizer to an input protocol on G which is fully-utilized on a spanning subgraph S of G.

Algorithm 3 A root-triggered synchronizer for a fully-utilized synchronous protocol Π' over a graph S.

Initialization: All nodes begin in the INIT state.

For node r designated as initiator:
Begin
  $state_r \leftarrow$ ACTIVE
  $parent_r \leftarrow \perp$
  $children_r \leftarrow \mathcal{N}_r(S)$

  r sends m_{rv} to each node $v \in children_r$
  r waits to receive a message m_{vr} from every node $v \in children_r$
  $state_r \leftarrow$ DONE
End

For every node u, upon receiving a message from w when in state INIT:
Begin
  $state_u \leftarrow$ ACTIVE
  $parent_u \leftarrow w$
  $children_u \leftarrow \mathcal{N}_r(S) \setminus \{w\}$

  u sends m_{uv} to each node $v \in children_u$
  u waits to receive a message m_{vu} from every node $v \in children_u$
  u sends m_{uw} to w
  $state_u \leftarrow$ DONE
End

▶ **Corollary 9.** *Multiple consecutive invocations of Algorithm 3 simulate any input fully-utilized synchronous protocol Π' round by round, resulting in an asynchronous protocol Π that uses the same number of messages.*

4.3 The Coding Scheme

We can now complete the details of our coding scheme for asynchronous networks with unknown topology. The scheme consists of two parts. In the first part, the scheme uses the BFS construction given in Section 3 in order to obtain a spanning BFS tree $\mathcal{T}$ of G. Note that the nodes ignore the content of messages during this part, therefore an adversary that can only modify messages cannot disturb this part.

In the second part, the scheme translates π into a fully-utilized synchronous protocol π' via $O(n)$ fully-utilized synchronous rounds over $\mathcal{T}$. This is done using Algorithm 2. The protocol π' is still non-resilient to noise and hence is not the protocol that is executed. Instead, we add a coding layer for multiparty interactive communication, namely via the Hoza-Schulman coding scheme, whose properties are given in Lemma 2. This results in a fully-utilized synchronous protocol Π' that is resilient to noise, which we then execute through our root-triggered synchronizer to obtain the asynchronous resilient protocol Π.

The complete construction is given in Algorithm 4. We prove its communication overhead in the following lemma, and then we prove its correctness and resilience.

Algorithm 4 A coding scheme Π for any noiseless asynchronous input protocol π.

Initialization: All nodes begin in the INIT state.

For node r designated as initiator:
Begin
    Execute Algorithm 1 with r designated as root. Let $\mathcal{T}$ be the obtained BFS tree.
    Let π' be a fully-utilized synchronous algorithm induced by π using Algorithm 2.
    Let $\Pi' = \mathrm{HS}(\pi')$ be the Hoza-Schulman coding scheme for π'.
    Simulate Π' using the synchronizer of Algorithm 3 over $\mathcal{T}$ with r as the initiator.
End

▶ **Lemma 10.** *For any asynchronous protocol π the coding Π of Algorithm 4 has a communication complexity of $\mathsf{CC}(\Pi) = O(nm \log n) + \mathsf{CC}(\pi) \cdot O(n^2 \log n)$.*

Proof. Recall that we assume channels with a fixed alphabet size, so that each symbol contains $O(\log n)$ bits (Remark 2). The $O(nm \log n)$ term follows from Theorem 3. The transformation of Algorithm 2 induces a communication overhead factor of $O(n^2 \log n)$ per bit of π, as shown in Lemma 7. By Lemma 2 there exists a resilient fully-utilized synchronous protocol Π' that simulates π' whose message/communication complexity is linear in the message complexity of π'. Finally, Corollary 9 gives that the asynchronous simulation of Π' via Algorithm 2 has the same message and communication complexity as Π'. It follows that the total overhead in communication of Algorithm 4 is $O(n^2 \log n)$, as claimed. ◀

▶ Remark 3. Note that the BFS construction (Algorithm 1) ignores the contents of messages. Hence, if we relax the assumption of Remark 2, the communication complexity can be reduced by sending empty messages (without any payload) during that step. In this case the message complexity of Π remains $O(mn) + \mathsf{CC}(\pi) \cdot O(n^2)$ yet the communication complexity reduces to $\mathsf{CC}(\Pi) = \mathsf{CC}(\pi) \cdot O(n^2 \log n)$.

▶ Remark 4. In the above, each message sent in π is split into single bits and a separate symbols is dedicated to each such bit. However, instead of communicating a single bit M_i in each symbol, nodes can aggregate blocks of $O(\log n)$ bits, so that the payload of each symbol is a single *block* (of π's communication) while keeping the coding's symbol size of the magnitude $O(\log n)$.

For some protocols, namely those which send large messages, this may result in a slight logarithmic decrease in the message complexity. This optimization, however, will not change the asymptotic overhead in the worst case, when the protocol π communicates a single bit at a time.

▶ **Lemma 11.** *For any asynchronous protocol π the coding Π of Algorithm 4 correctly simulates π even if up to $\Theta(1/n)$ of the messages are adversarially corrupted.*

Proof. Correctness and resilience to noise are proved as follows. Theorem 3 proves the correctness of our content-oblivious BFS construction despite noise, since the contents of the sent messages are ignored by the nodes. We emphasize that by Corollary 4, all of the nodes know when to stop ignoring the content of messages for the BFS construction and start executing that synchronizer over Π'.

Lemma 7 proves that indeed π' is a fully-utilized synchronous transformation of π. By Lemma 2, we have that Π' is a fully-utilized synchronous protocol that simulates π' in a manner that is resilient to corrupting up to $\Theta(1/|\tilde{E}|)$ of the messages, where $\tilde{E}$ is the edges

over which the protocol communicates. In our case these are the edges of the BFS tree $\mathcal{T}$, and hence this step is resilient to an $\Theta(1/n)$-fraction of corruptions.

Finally, Corollary 9 gives that Π' is executed correctly in the asynchronous setting despite noise.

We now need to sum up the maximal number of symbols that can be corrupted and the total number of communicated symbols. Recall that the noise resilience is the ratio between these two sums. Since corruption can only take place on symbols of the Hoza-Schulman coding, of which there are $\mathsf{CC}(\pi) \cdot O(n^2)$ many, we get that the scheme is resilient to at most $O(1/n) \cdot CC(\pi) \cdot O(n^2)$ corrupted symbols. The total number of symbols communicated in the scheme includes also the $O(mn)$ symbols required for constructing the BFS tree, implying that our scheme is resilient to a fraction of symbol corruption equal to

$$\frac{O(1/n) \cdot CC(\pi) \cdot O(n^2)}{O(nm) + CC(\pi) \cdot O(n^2)}.$$

This is asymptotically equal to an $O(1/n)$ fraction of noise when $\mathsf{CC}(\pi) > n$, $\mathsf{CC}(\pi) \to \infty$. ◀

Lemmas 10 and 11 directly give our main theorem for this section, Theorem 6.

5 A Spanner-Based Distributed Interactive Coding Scheme

In this section we slightly improve the overhead obtained by the coding scheme of Theorem 6. We demonstrate a family of coding schemes with an interesting tradeoff between their overhead and resilience. The key ingredient is replacing the underlying infrastructure of the BFS tree $\mathcal{T}$ with a sparse spanning graph S, where we can trade off the sparseness of the graph (i.e., the number of edges it contains, and as a consequence, the resilience of the obtained coding scheme) with its distance distortion (i.e., the maximal distance in S for any neighboring nodes in G, and as a consequence, the added overhead for routing messages through S in the coding scheme).

Assume u sends v a message in the input protocol π. The coding scheme of Algorithm 4 routes every such message via the BFS tree $\mathcal{T}$. This incurs a delay in Π', which can be of $O(n)$ rounds: in the worst case, u and v which are neighbors in G may now be two leaves of $\mathcal{T}$ whose distance is n. In fact, even if their distance in $\mathcal{T}$ is smaller, the coding scheme is not aware of this fact and must propagate the message to the entire network. The only guarantee we have in this case is that the message reaches its destination after at most n rounds (of the underlying fully-utilized synchronous protocol).

In this section we suggest a way to reduce the delay factor of n by routing messages over *a spanner* rather than over the tree $\mathcal{T}$.

▶ **Definition 12** (t-Spanner)**.** A subgraph $S = (V, E_S)$ is a t-spanner of $G = (V, E)$ if for every $(u, v) \in E$ it holds that $dist(u, v) \leq t$ in S.

Replacing the BFS tree $\mathcal{T}$ with a t-spanner that has $s = |E_S|$ edges ensures that a message reaches its destination after at most t steps (instead of n). Since the noise resilience is determined by the number of edges used by the underlying fully-utilized synchronous protocol, by Lemma 2, we obtain a resilience of $\Theta(1/s)$. The main result of this section is the following.

▶ **Theorem 13.** *Let $\pi_{spanner}$ be an asynchronous distributed algorithm for constructing a t-spanner S with s edges in a noiseless setting. Any asynchronous protocol π over a network G with $\mathsf{CC}(\pi) \gg \mathsf{CC}(\pi_{spanner})$ can be simulated by a noise-resilient asynchronous protocol Π resilient to an $\Theta(1/s)$-fraction of message corruption and it holds that $\mathsf{CC}(\Pi) = \mathsf{CC}(\pi) \cdot O(st \log n)$.*

Specifically, due to the existence of $O(\log n)$-spanners with $O(n)$ edges [3, 34] (see also [33, Section 16]), we can let $\pi_{spanner}$ be a distributed construction of a spanner with the same parameters [13] and obtain the following corollary.

▶ **Corollary 14.** *Let $\pi_{spanner}$ be an asynchronous distributed algorithm for constructing a $\log n$-spanner with $O(n)$ edges in a noiseless setting. Any asynchronous protocol π over a network G with $\mathsf{CC}(\pi) \gg \mathsf{CC}(\pi_{spanner})$ can be simulated by a noise-resilient asynchronous protocol Π resilient to an $\Theta(1/n)$-fraction of message corruption and it holds that $\mathsf{CC}(\Pi) = \mathsf{CC}(\pi) \cdot O(n \log^2 n)$.*

We defer the detailed construction and proofs to the full version of this paper (see [11]).

Acknowledgement. We are grateful to Merav Parter for bringing [13] to our attention.

References

1. Noga Alon, Mark Braverman, Klim Efremenko, Ran Gelles, and Bernhard Haeupler. Reliable communication over highly connected noisy networks. In George Giakkoupis, editor, *Proceedings of the 2016 ACM Symposium on Principles of Distributed Computing, PODC 2016, Chicago, IL, USA, July 25-28, 2016*, pages 165–173. ACM, 2016. doi:10.1145/2933057.2933085.
2. Hagit Attiya and Jennifer Welch. *Distributed Computing: Fundamentals, Simulations and Advanced Topics*. John Wiley & Sons, 2004.
3. Baruch Awerbuch. Complexity of network synchronization. *J. ACM*, 32(4):804–823, 1985. doi:10.1145/4221.4227.
4. Baruch Awerbuch, Boaz Patt-Shamir, David Peleg, and Michael E. Saks. Adapting to asynchronous dynamic networks (extended abstract). In S. Rao Kosaraju, Mike Fellows, Avi Wigderson, and John A. Ellis, editors, *Proceedings of the 24th Annual ACM Symposium on Theory of Computing, May 4-6, 1992, Victoria, British Columbia, Canada*, pages 557–570. ACM, 1992. doi:10.1145/129712.129767.
5. Piotr Berman, Krzysztof Diks, and Andrzej Pelc. Reliable broadcasting in logarithmic time with byzantine link failures. *J. Algorithms*, 22(2):199–211, 1997. doi:10.1006/jagm.1996.0810.
6. Zvika Brakerski, Yael Tauman Kalai, and Moni Naor. Fast interactive coding against adversarial noise. *J. ACM*, 61(6):35:1–35:30, 2014. doi:10.1145/2661628.
7. Mark Braverman and Klim Efremenko. List and unique coding for interactive communication in the presence of adversarial noise. *SIAM J. Comput.*, 46(1):388–428, 2017. doi:10.1137/141002001.
8. Mark Braverman, Klim Efremenko, Ran Gelles, and Bernhard Haeupler. Constant-rate coding for multiparty interactive communication is impossible. 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 999–1010. ACM, 2016. doi:10.1145/2897518.2897563.
9. Mark Braverman, Ran Gelles, Jieming Mao, and Rafail Ostrovsky. Coding for interactive communication correcting insertions and deletions. *IEEE Trans. Information Theory*, 63(10):6256–6270, 2017. doi:10.1109/TIT.2017.2734881.
10. Mark Braverman and Anup Rao. Toward coding for maximum errors in interactive communication. *IEEE Trans. Information Theory*, 60(11):7248–7255, 2014. doi:10.1109/TIT.2014.2353994.
11. Keren Censor-Hillel, Ran Gelles, and Bernhard Haeupler. Making asynchronous distributed computations robust to noise, 2017. arXiv:1702.07403.

12 Pallab Dasgupta. Agreement under faulty interfaces. *Inf. Process. Lett.*, 65(3):125–129, 1998. doi:10.1016/S0020-0190(97)00202-0.

13 Bilel Derbel, Mohamed Mosbah, and Akka Zemmari. Sublinear fully distributed partition with applications. *Theory Comput. Syst.*, 47(2):368–404, 2010. doi:10.1007/s00224-009-9190-x.

14 Klim Efremenko, Ran Gelles, and Bernhard Haeupler. Maximal noise in interactive communication over erasure channels and channels with feedback. *IEEE Trans. Information Theory*, 62(8):4575–4588, 2016. doi:10.1109/TIT.2016.2582176.

15 Ofer Feinerman, Bernhard Haeupler, and Amos Korman. Breathe before speaking: efficient information dissemination despite noisy, limited and anonymous communication. In Magnús M. Halldórsson and Shlomi Dolev, editors, *ACM Symposium on Principles of Distributed Computing, PODC '14, Paris, France, July 15-18, 2014*, pages 114–123. ACM, 2014. doi:10.1145/2611462.2611469.

16 Michael J. Fischer, Nancy A. Lynch, and Mike Paterson. Impossibility of distributed consensus with one faulty process. *J. ACM*, 32(2):374–382, 1985. doi:10.1145/3149.214121.

17 Matthew K. Franklin, Ran Gelles, Rafail Ostrovsky, and Leonard J. Schulman. Optimal coding for streaming authentication and interactive communication. *IEEE Trans. Information Theory*, 61(1):133–145, 2015. doi:10.1109/TIT.2014.2367094.

18 Robert G. Gallager. Distributed minimum hop algorithms. Technical Report LIDS-P-1175, M.I.T. Laboratory for Information and Decision Systems, 1982.

19 Ran Gelles. Coding for interactive communication: A survey. *Foundations and Trends in Theoretical Computer Science*, 13(1-2):1–157, 2017. doi:10.1561/0400000079.

20 Ran Gelles, Bernhard Haeupler, Gillat Kol, Noga Ron-Zewi, and Avi Wigderson. Towards optimal deterministic coding for interactive communication. 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 1922–1936. SIAM, 2016. doi:10.1137/1.9781611974331.ch135.

21 Ran Gelles and Yael Tauman Kalai. Constant-rate interactive coding is impossible, even in constant-degree networks. 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 21:1–21:13. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2017. doi:10.4230/LIPIcs.ITCS.2017.21.

22 Ran Gelles, Ankur Moitra, and Amit Sahai. Efficient coding for interactive communication. *IEEE Trans. Information Theory*, 60(3):1899–1913, 2014. doi:10.1109/TIT.2013.2294186.

23 Mohsen Ghaffari, Bernhard Haeupler, and Madhu Sudan. Optimal error rates for interactive coding I: adaptivity and other settings. In David B. Shmoys, editor, *Symposium on Theory of Computing, STOC 2014, New York, NY, USA, May 31 - June 03, 2014*, pages 794–803. ACM, 2014. doi:10.1145/2591796.2591872.

24 Li Gong, Patrick Lincoln, and John Rushby. Byzantine agreement with authentication: Observations and applications in tolerating hybrid and link faults. In *Dependable Computing and Fault Tolerant Systems*, volume 10, pages 139–158. IEEE Computer Society, 1995. URL: http://www.csl.sri.com/papers/dcca95/dcca95.pdf.

25 Bernhard Haeupler. Interactive channel capacity revisited. In *55th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2014, Philadelphia, PA, USA, October 18-21, 2014*, pages 226–235. IEEE Computer Society, 2014. doi:10.1109/FOCS.2014.32.

26 Michael Harrington and Arun K. Somani. Synchronizing hypercube networks in the presence of faults. *IEEE Trans. Computers*, 43(10):1175–1183, 1994. doi:10.1109/12.324543.

27 William M. Hoza and Leonard J. Schulman. The adversarial noise threshold for distributed protocols. 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 240–258. SIAM, 2016. doi:10.1137/1.9781611974331.ch18.

28 Abhishek Jain, Yael Tauman Kalai, and Allison Bishop Lewko. Interactive coding for multiparty protocols. In Tim Roughgarden, editor, *Proceedings of the 2015 Conference on Innovations in Theoretical Computer Science, ITCS 2015, Rehovot, Israel, January 11-13, 2015*, pages 1–10. ACM, 2015. doi:10.1145/2688073.2688109.

29 Gillat Kol and Ran Raz. Interactive channel capacity. 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 715–724. ACM, 2013. doi:10.1145/2488608.2488699.

30 Allison Lewko and Ellen Vitercik. Balancing communication for multi-party interactive coding, 2015. arXiv:1503.06381.

31 Nancy A. Lynch. *Distributed Algorithms*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1996.

32 Andrzej Pelc. Reliable communication in networks with byzantine link failures. *Networks*, 22(5):441–459, 1992. doi:10.1002/net.3230220503.

33 David Peleg. *Distributed Computing: A Locality-Sensitive Approach*. Society for Industrial and Applied Mathematics, 2000. doi:10.1137/1.9780898719772.

34 David Peleg and Alejandro A. Schäffer. Graph spanners. *Journal of Graph Theory*, 13(1):99–116, 1989. doi:10.1002/jgt.3190130114.

35 Sridhar Rajagopalan and Leonard J. Schulman. A coding theorem for distributed computation. In Frank Thomson Leighton and Michael T. Goodrich, editors, *Proceedings of the Twenty-Sixth Annual ACM Symposium on Theory of Computing, 23-25 May 1994, Montréal, Québec, Canada*, pages 790–799. ACM, 1994. doi:10.1145/195058.195462.

36 Hasan Md. Sayeed, Marwan Abu-Amara, and Hosame Abu-Amara. Optimal asynchronous agreement and leader election algorithm for complete networks with byzantine faulty links. *Distributed Computing*, 9(3):147–156, 1995. doi:10.1007/s004460050016.

37 Leonard J. Schulman. Communication on noisy channels: A coding theorem for computation. In *33rd Annual Symposium on Foundations of Computer Science, Pittsburgh, Pennsylvania, USA, 24-27 October 1992*, pages 724–733. IEEE Computer Society, 1992. doi:10.1109/SFCS.1992.267778.

38 Leonard J. Schulman. Coding for interactive communication. *IEEE Transactions on Information Theory*, 42(6):1745–1756, 1996.

39 Gurdip Singh. Leader election in the presence of link failures. *IEEE Trans. Parallel Distrib. Syst.*, 7(3):231–236, 1996. doi:10.1109/71.491576.

40 Hin-Sing Siu, Yeh-Hao Chin, and Wei-Pang Yang. Byzantine agreement in the presence of mixed faults on processors and links. *IEEE Trans. Parallel Distrib. Syst.*, 9(4):335–345, 1998. doi:10.1109/71.667895.

41 Gerard Tel. *Introduction to distributed algorithms*. Cambridge university press, 2000. Chapter 12.4. Asynchronous BFS Algorithms, pages 414–420.

Distance-Preserving Graph Contractions*

Aaron Bernstein[1], Karl Däubel[2], Yann Disser[†3], Max Klimm[4], Torsten Mütze[5], and Frieder Smolny[6]

1 **Institut für Mathematik, TU Berlin, Germany**
bernstein@math.tu-berlin.de
2 **Institut für Mathematik, TU Berlin, Germany**
daeubel@math.tu-berlin.de
3 **Department of Mathematics, Graduate School CE, TU Darmstadt, Germany**
disser@mathematik.tu-darmstadt.de
4 **Wirtschaftswissenschaftliche Fakultät, HU Berlin, Germany**
max.klimm@hu-berlin.de
5 **Institut für Mathematik, TU Berlin, Germany**
muetze@math.tu-berlin.de
6 **Institut für Mathematik, TU Berlin, Germany**
smolny@math.tu-berlin.de

Abstract

Compression and sparsification algorithms are frequently applied in a preprocessing step before analyzing or optimizing large networks/graphs. In this paper we propose and study a new framework contracting edges of a graph (merging vertices into super-vertices) with the goal of preserving pairwise distances as accurately as possible. Formally, given an edge-weighted graph, the contraction should guarantee that for any two vertices at distance d, the corresponding super-vertices remain at distance at least $\varphi(d)$ in the contracted graph, where φ is a tolerance function bounding the permitted distance distortion. We present a comprehensive picture of the algorithmic complexity of the contraction problem for affine tolerance functions $\varphi(x) = x/\alpha - \beta$, where $\alpha \geq 1$ and $\beta \geq 0$ are arbitrary real-valued parameters. Specifically, we present polynomial-time algorithms for trees as well as hardness and inapproximability results for different graph classes, precisely separating easy and hard cases. Further we analyze the asymptotic behavior of the size of contractions, and find efficient algorithms to compute (non-optimal) contractions despite our hardness results.

1998 ACM Subject Classification G.2.2 Graph Theory

Keywords and phrases distance oracle, contraction, spanner

Digital Object Identifier 10.4230/LIPIcs.ITCS.2018.51

1 Introduction

When dealing with large networks, it is often beneficial to compress or sparsify the data to manageable size before analyzing or optimizing the network directly. To be useful, a meaningful compression should represent salient features of the original network with good approximation, while being much smaller in size. In this paper, we focus on a compression of

* A preprint version of this paper with full proofs is available at http://arxiv.org/abs/1705.04544
† Supported by the 'Excellence Initiative' of the German Federal and State Governments and the Graduate School CE at TU Darmstadt.

undirected edge-weighted graphs that approximately maintains all distances between vertices in the graph.

In this context, an extensively studied concept are *spanners* (e.g. [19, 3, 6, 1]). Given an undirected graph $G = (V, E)$ and real numbers $\alpha \geq 1$ and $\beta \geq 0$, a subgraph $H = (V, E')$, $E' \subseteq E$, is an (α, β)-*spanner of* G if $\text{dist}_H(u, v) \leq \alpha \cdot \text{dist}_G(u, v) + \beta$ holds for all $u, v \in V$. While the number of edges in a spanner may be much smaller than that of the original graph, the number of vertices is the same for both, leaving further potential for compression untapped. For illustration, consider the road network of Europe with about 50 million vertices [5], any spanner of which must again have about 50 million vertices and edges. However, to approximately represent distances in Europe's road network one may also merge nearby vertices into super-vertices, thus achieving a much better compression of the network. This is akin to the visual process of zooming out of a graphical representation of the map, where neighbored vertices fade into each other and edges between merged vertices vanish. At a large enough zoom level, the entire network merges into a single vertex.

In this paper we propose and study a new framework for contracting networks that formalizes this intuitive idea and makes it applicable to general graphs (even without metric embedding). Specifically, we study a contraction problem on graphs where a subset of edges $C \subseteq E$ is contracted. We denote the resulting simple graph obtained from G by contracting the edges in C and by deleting resulting loops and multiple edges, keeping only the shortest edge between any two vertices, by G/C. For any two vertices in G, we compare their distance in G with the distance of the corresponding super-vertices in G/C.

It is interesting to contrast this concept with graph spanners. When constructing a spanner, the length of the removed edges is implicitly set to ∞, resulting in an overall increase of distances. On the other hand, a contraction implicitly sets the length of the contracted edges to zero, leading to an overall decrease of distances. For both problems, the ultimate goal is to reduce the complexity of the network while maintaining an approximation guarantee on the distances.

The following example shows that contractions may be better suited than spanners to achieve this goal. In a subgraph with small radius, a spanner can at best result in a spanning tree of the same order, while a contraction can reduce the whole subgraph to a single vertex, while entailing a multiplicative distance distortion of similar magnitude. In addition, the contraction may also merge many edges entering the contracted subgraph. Clearly, the objective here is to maximize the total number of contracted and deleted edges, as this minimizes the memory required to represent the resulting network in a computer (using e.g. adjacency lists).

Given the results presented in this paper and the known results for spanners (discussed in detail below), we further believe that the combination of spanners and contractions is very powerful, promising and flexible. As the former only increases and the latter only decreases the distances, the respective distortion guarantees provably also hold for the overall distortion. In fact, both effects may even compensate each other. This is true *regardless* of the order in which both compression operations are applied, even when they are applied repeatedly.

In order to measure the distance distortion of the contraction, we assume a non-decreasing tolerance function $\varphi\colon \mathbb{R} \to \mathbb{R}$, similar to the corresponding function for spanners, see e.g. [6]. We are interested in computing contractions that preserve distances in the following sense: For any two vertices u and v at distance d in G, the distance of the corresponding vertices in the contracted graph G/C must be at least $\varphi(d)$. If this condition is satisfied, we call C a *φ-distance preserving contraction*, or *φ-contraction* for short. Formally, the algorithmic problem Contraction considered in this paper is to compute for a given graph $G = (V, E)$

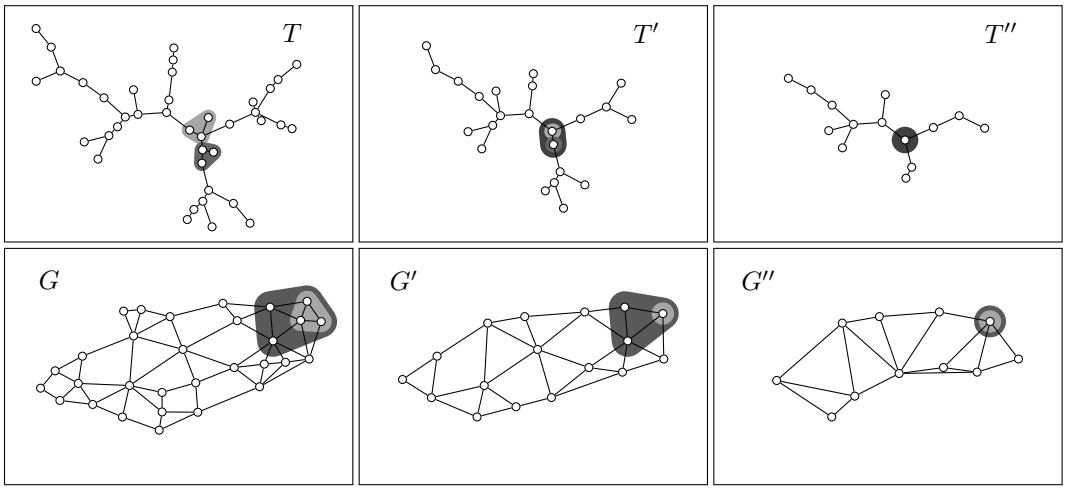


Figure 1 Top: two iterations of Contraction with $\varphi(x) = 4x/5 - 3$ on a tree; bottom: two iterations of Contraction with $\varphi(x) = 3x/4 - 3$ on a planar graph. Distances are geometric and some contracted sets of vertices are highlighted.

with edge lengths $\ell\colon E \to \mathbb{R}_{>0}$ and a given tolerance function φ, a φ-contraction $C \subseteq E$ such that the number of contracted and deleted edges is maximized. We are specifically interested in the case where the tolerance function φ is an affine function $\varphi(x) = x/\alpha - \beta$ for real-valued parameters $\alpha \geq 1$ and $\beta \geq 0$. We then simply write (α, β)-*contraction* instead of φ-contraction. See Figure 1 for some example instances of the problem Contraction.

When considering the case of a purely multiplicative error ($\beta = 0$), a slight subtlety has to be taken into account. Specifically, for a graph with positive edge lengths it is not feasible to contract a single edge. Therefore, we propose a slight modification of our original model: We say that a set $C \subseteq E$ of edges of G is a *weak φ-distance preserving contraction*, or *weak φ-contraction* for short, if it does not contract the entire graph and, for any two vertices u and v at distance d in G, the distance of the corresponding vertices in G/C is either zero or at least $\varphi(d)$. We will refer to the corresponding algorithmic problem as Weak Contraction. Put differently, in a weak contraction, the distances between different super-vertices satisfy the given distortion guarantee, but for vertices belonging to the same super-vertex, no guarantee is given.

1.1 Our results

In this paper, we present a comprehensive picture of the algorithmic complexity of the described contraction problems. Recall that we are given an input graph with edge lengths and tolerance function φ, and our goal is to compute a (weak) contraction that maximizes the total number of contracted and deleted edges. Our main results concern affine tolerance functions $\varphi(x) = x/\alpha - \beta$ with parameters $\alpha \geq 1$ and $\beta \geq 0$. For the reader's convenience, our results are summarized in Tables 1, and 2. Within the tables and throughout this paper, n and m denote the number of vertices and edges, respectively, of the input graph under consideration.

Table 1 Overview of algorithmic and hardness results presented in this paper.

Problem	Graph classes: Path	Tree	Cycle	General
Contraction				
addit. (α=1), unit lg.	$\mathcal{O}(n)$ [Th. 1 1]	$\mathcal{O}(n)$ [Th. 1 3]	$\mathcal{O}(n)$ [Th. 1 2]	$m^{\frac{1}{2}-\varepsilon}$-inapx.[a] [Th. 8]
affine (α, β), unit lg.				
addit. (α=1)	$\mathcal{O}(n^3)$ [Th. 2]		NP-hard [Th. 6]	$n^{1-\varepsilon}$-inapx. [Th. 7]
affine (α, β)				
Weak Contraction				
additive (α=1)	$\mathcal{O}(n^5)$ [Th. 4]		NP-hard[b] [Th. 6]	
affine (α, β)				$n^{1-\varepsilon}$-inapx.[c] [Th. 10]

[a] even for bipartite graphs and $\beta = 1$
[b] also NP-hard for planar graphs with arb. large girth, $(\alpha, \beta) = (2, 0)$, and unit lg. ($\ell = 1$) [Th. 9].
[c] even if $(\alpha, \beta) = (3/2, 0)$.

Algorithmic results

We develop linear time greedy algorithms for Contraction with unit lengths on paths, cycles, and on trees with $\alpha = 1$ (Theorem 1). The first two algorithms are inspired by LP rounding techniques, the latter algorithm relies on a structural characterization of optimal solutions.

We present dynamic programming algorithms solving Contraction and Weak Contraction on trees in time $\mathcal{O}(n^3)$ or $\mathcal{O}(n^5)$, respectively (Theorems 2 and 4). These dynamic programs compute optimal solutions on subtrees, in the latter case combining several Pareto optimal solutions in a two-dimensional parameter space (hence the larger running time).

Note that instead of maximizing the number of contracted and deleted edges, we could optimize for α or β while fixing the other parameters. The resulting problems are polynomially equivalent to our setting, via binary search over one of the parameters.

Hardness results

We complement these algorithms by several hardness results. First we consider the purely additive case where $\alpha = 1$. We show that here both Contraction and Weak Contraction are NP-hard on cycles for any fixed $\beta > 0$, by a reduction of a variant of Partition (Theorem 6). As mentioned before, both problems can be solved efficiently on graphs without cycles, and there is a linear time algorithm for Contraction on cycles with unit lengths. By reductions from Clique we show that both the general as well as the unit lengths case of Contraction with $\alpha = 1$ are hard to approximate within factors of $n^{1-\varepsilon}$ or $m^{1/2-\varepsilon}$, respectively (Theorem 7 and Theorem 8).

Further we consider the purely multiplicative case where $\beta = 0$ (here Contraction is trivial). We show that in this case Weak Contraction is NP-hard on planar graphs with arbitrarily large girth and unit length edges by a reduction from a special case of Planar 3SAT (Theorem 9). Since these graphs are locally tree-like, this result constitutes another rather sharp separation from the polynomially solvable tree case. Furthermore, we show that the problem is hard to approximate within a factor of $n^{1-\varepsilon}$ by a reduction from Independent Set (Theorem 10).

Table 2 Overview of asympotic bounds presented in this paper.

Contraction with unit lg. (ℓ=1)	# of edges in G/C	Time	Reference
$(\alpha, \beta) = (2k-1, 1)$	$n^{1+1/k}$	$\mathcal{O}(m)$	[Th. 11]
$(\alpha, \beta) = (2\log_2 n - 1, 1)$	$2n$	$\mathcal{O}(m)$	[Cor. 12]
$(\alpha, \beta) = (k-1, 1)$	$\Omega(n^{1+1/k})$	—	[Th. 14]
$(\alpha, \beta) = (1, k)$	$m - km/(2n)$	$\mathcal{O}(m)$	[Th. 15 1]
$(\alpha, \beta) = (1, k)$	$\mathcal{O}(n^2/k)$	$\mathcal{O}(m)$	[Th. 15 2]
$(\alpha, \beta) = (1, \mathcal{O}(1))$	$\Omega(n^{4/3-o(1)})$	—	[1]
Contraction with unit lg. (ℓ=1) and min. degree D	**# of vertices in G/C**	**Time**	**Reference**
$(\alpha, \beta) = (5, 1)$	n/D	$\mathcal{O}(m)$	[Th. 16]
$(\alpha, \beta) = (k, 1)$	$\Omega(n/(kD))$	—	[Th. 17]

Asymptotic bounds

We now discuss our asymptotic bounds for contractions. In this setting, we are interested in (non-optimal) contractions for graphs with unit lengths that can be computed efficiently despite the above-mentioned hardness results. We prove that for any $k \geq 1$ any graph G has a $(2k-1, 1)$-contraction C such that G/C has at most $n^{1+1/k}$ edges, and such a contraction can be computed in time $\mathcal{O}(m)$ (Theorem 11) by successively growing clusters around center vertices. Assuming Erdős' girth conjecture, we show a corresponding (not tight) lower bound (Theorem 14).

For a purely additive error, we observe two simple $(1, k)$-contractions that can be computed in $\mathcal{O}(m)$ time (Theorem 15). We show that for any even integer $0 \leq k \leq n$, the edges incident to the $k/2$ vertices of highest degrees form a $(1, k)$-contraction with objective value at least $km/(2n)$, which is asymptotically best possible for paths. Another $(1, k)$-contraction C is implicitly used by Bernstein and Chechik in their faster deterministic algorithm for dynamic shortest paths in dense graphs [8]. For any number $0 < k \leq n$, it consists of the edges incident to two vertices of degree at least n/k, and G/C has $\mathcal{O}(n^2/k)$ edges. Both of these contractions can be computed in $\mathcal{O}(m)$ time. Further we note that the main result in [1] implies that for all $\varepsilon > 0$, any contraction C such that G/C has $\mathcal{O}(n^{4/3-\varepsilon})$ edges does not admit a constant additive error.

One possible advantage of contraction compared to spanners is the potentially significant reduction of *vertices* as well as edges, e.g. reducing the complexity of performing algorithmic tasks in the smaller graph. To ground this intuition, we exhibit a contraction that significantly reduces the number of vertices in any graph with minimum degree D to $\mathcal{O}(n/D)$ (Theorem 16). We also present a lower bound (Theorem 17) showing that we cannot guarantee $o(n/D)$ vertices, even if we allow larger approximation error.

1.2 Comparison with previous results

There are several models aiming to compress graphs while preserving distances. They differ by their choice of compression operation, such as replacing the graph by a subgraph or minor, and by whether the aim is to preserve all or only certain distances.

As discussed before, graph spanners are a concept closely related to contractions, where the length of removed edges is set to ∞ rather than to 0. Our results highlight further intrinsic similarities of the two models. Like contractions, spanners are NP-hard to compute

optimally (see [19, 18]). While the spanner literature considers the problem of minimizing the number of remaining edges, we analyze the objective of maximizing the number of contracted edges, prohibiting a direct comparison of the respective inapproximability results. We note however that approximation algorithms for spanner problems have been studied extensively, even though strong lower bounds are known. For instance, computing (2, 0)-spanners in unweighted graphs is $\Theta(\log n)$-hard to approximate ([16, 15]), for further references see e.g. [11].

Despite these negative results, it is still possible to obtain powerful asymptotic guarantees in both models. In particular, our $(2k-1,1)$-contraction with $\mathcal{O}(n^{1+1/k})$ edges for unweighted graphs has a clear analogy to the classic $(2k-1,0)$-spanner with the same number of edges [3] (note that the additive error of 1 in our result is strictly necessary, as discussed above). There is, however, a major difference between the two results: whereas the $(2k-1,0)$-spanner can trivially be shown to be optimal assuming Erdős' girth conjecture, applying this conjecture to the contraction model only yields a lower bound of $n^{1+1/(2k)}$ edges for a $(2k-1,1)$-contraction. Closing this gap thus remains as an interesting open problem in the contraction model, whose solution would likely yield further insight into the relationship to spanners.

It is interesting to note that the clustering yielding our $(2k-1,1)$-contraction was previously used in [19] to obtain a $(4r+1,0)$-spanner of the same density. On the other hand, no deterministic linear time algorithm computing a $(2k-1,0)$-spanner is known, though [7] achieves randomized linear time. Meanwhile our $(2k-1,1)$-contraction can be constructed deterministically in linear time.

There are also spanner results that significantly sparsify unweighted graphs at the cost of a purely additive error, as a (1,2)-spanner with $\mathcal{O}(n^{3/2})$ edges [2], or a (1,6)-spanner with $\mathcal{O}(n^{4/3})$ edges [6]. We do not know if analogous results are possible in the contraction model. The incompressibility result in [1] mentioned above implies the same lower bound for spanners as for contractions and every other distance oracle with additive error: For every $\varepsilon > 0$ any spanner of size $\mathcal{O}(n^{4/3-\varepsilon})$ does not admit a constant additive error. Finally, for spanners there are results that combine multiplicative and additive error, such as the $(k, k-1)$-spanner of [6].

Gupta [14] considered the problem of approximating a tree metric on a subset of the vertices by another tree, and gave a linear time algorithm computing an 8-approximation. As Chan et al. [9] observed later, on complete binary trees a solution of minimum distortion is always achieved by a minor (with possibly different edge lengths) of the input tree, so this seems to be the first investigation of contractions that approximate graph distances. Krauthgamer et al. [17] considered an extension to general graphs, studying the size of minors preserving all distances between a given terminal set of fixed size. Cheung et al. [10] introduced a multiplicative distortion to this model. As here no two terminals may be merged, these approaches cannot compress a graph at all if every vertex is a terminal.

1.3 Outline of this paper

In Section 2 we introduce important definitions and notations that will be used throughout this paper. In Sections 3–6 we formally state our results, in exactly the same order as they were discussed in Section 1.1 before. Due to the limited space in this extended abstract, we will only mention the main steps and ideas needed to prove a few selected theorems. Full proofs can be found in the preprint [12].

2 Preliminaries

Throughout this paper we consider simple undirected graphs G (without parallel edges or loops). We let $V(G)$ and $E(G)$ denote the vertex and edge set of G, respectively, and we define $n(G) := |V(G)|$ and $m(G) := |E(G)|$. If the context is clear, we simply write V, E, n and m. We also use the notation $[n] := \{1, 2, \dots, n\}$. We assume that G is connected, otherwise the contraction problem can be solved independently for each connected component. Edge lengths are given by a function $\ell\colon E \to \mathbb{R}_{>0}$. The *distance* $\mathrm{dist}_\ell(u, v)$ between two vertices u and v is the length of a shortest path between u and v in G with respect to ℓ.

Given a subset of edges $C \subseteq E$, we denote the resulting simple graph obtained from G by contracting the edges in C, deleting resulting loops and keeping only the shortest edge between any two vertices by G/C. We denote the number of deleted loops and multi-edges by $\Delta(C)$ (thus $m(G/C) = m(G) - |C| - \Delta(C)$). Instead of contracting a set $C \subseteq E$ of edges in G, setting their edge lengths to zero has the same effect on the distances in the resulting graph. This is somewhat cleaner conceptually, so we will often adopt this viewpoint. Specifically, we let ℓ_C be the new length function that assigns 0 to every edge in C, and that is equal to the original edges lengths ℓ on the edges $E \setminus C$.

A *tolerance function* is a non-decreasing function $\varphi\colon \mathbb{R} \to \mathbb{R}$. Roughly speaking, this function describes by how much the distance between two vertices may drop when contracting edges (i.e., setting edge lengths to zero). Formally, given a graph G with edge lengths ℓ and a tolerance function φ, we say that a subset of edges $C \subseteq E$ is a *φ-distance preserving contraction* or *φ-contraction* for short, if

$$\mathrm{dist}_{\ell_C}(u, v) \geq \varphi(\mathrm{dist}_\ell(u, v)) \tag{1}$$

holds for any two vertices u and v in G. Similarly, we say that C is a *weak φ-distance preserving contraction* or *weak φ-contraction* for short, if (1) or $\mathrm{dist}_{\ell_C}(u, v) = 0$ holds for any two vertices u and v, and if the graph (V, C) is disconnected (equivalently, if G/C is not a single vertex). The last condition prevents solutions $C \subseteq E$ for which the graph is contracted to a single vertex. If $\varphi(x) = x/\alpha - \beta$, then we simply write (weak) (α, β)-contraction instead of (weak) φ-contraction.

An *instance* of the problem Contraction or Weak Contraction is a triple (G, ℓ, φ), where G is the underlying graph, ℓ the length function and φ the tolerance function, and the objective is to find a (weak) φ-distance preserving contraction $C \subseteq E$, such that

$$\Phi(C) := |C| + \Delta(C) = m(G) - m(G/C) \tag{2}$$

is maximized. This quantity equals the number of edges we save when going from G to G/C. Note that for instance on trees we have $\Phi(C) = |C|$ for any (weak) contraction C.

In this context we sometimes refer to a set of edges that forms a (weak) contraction as a *feasible* solution, and to a (weak) contraction of maximum size as an *optimal* solution.

Note that our contraction model is well-behaved in the sense that successively solving (Weak) Contraction with general tolerance functions φ and ψ yields a feasible solution with respect to the composition $\psi \circ \varphi$ (see for a proof [12]).

3 Greedy algorithms

In this section we summarize our results on greedy algorithms that allow solving several special cases of the problem Contraction with affine tolerance function $\varphi(x) = x/\alpha - \beta$ in linear time. The proof of the three cases in Theorem 1 can be found in [12].

▶ **Theorem 1.** *We can solve* Contraction *in time* $\mathcal{O}(n)$ *in the following three cases:*
(i) *Paths with* $\ell = 1$ *and* $\varphi(x) = x/\alpha - \beta$, $\alpha, \beta \geq 1$.
(ii) *Cycles with* $\ell = 1$ *and* $\varphi(x) = x/\alpha - \beta$, $\alpha \geq 1, \beta \geq 0$.
(iii) *Trees with* $\ell = 1$ *and* $\varphi(x) = x - \beta$, $\beta \geq 0$.

Generalizing cases 1 and 3, in the next section we will present polynomial time algorithms for the general case on trees (with somewhat larger running times). In contrast to the algorithmic result for unit length cycles in case 2, we will see in Section 5 that Contraction is NP-hard on cycles with general edge lengths, even with $\alpha = 1$.

4 Dynamic programs for general trees

In this section we consider the problem of computing (weak) contractions for trees $T = (V, E)$ with affine tolerance function $\varphi(x) = x/\alpha - \beta$. Recall that on trees we have $f(C) = |C|$ for every (weak) contraction C. In the following we present the main steps of our dynamic programming approach for solving these problems, first for the problem Contraction and then for Weak Contraction. Full proofs are deferred to [12].

▶ **Theorem 2.** *We can solve* Contraction *on trees with* $\varphi(x) = x/\alpha - \beta$, $\alpha \geq 1$ *and* $\beta \geq 0$ *in time* $\mathcal{O}(n^3)$.

The idea is to root the tree T at an arbitrary vertex, and to decompose the problem by splitting T into rooted subtrees at every vertex. Specifically, let v be a vertex of T, and T_1 and T_2 subtrees of T rooted at v that only have the vertex v in common. Now consider an optimal contraction C on T, and let C_1 and C_2 be the subsets of C on T_1 or T_2, respectively. Clearly, C_1 and C_2 are feasible contractions on their subtrees. Furthermore, the set C_2 has maximum size under the condition that its union with C_1 forms a feasible contraction (and vice versa).

We thus identified two quality parameters of solutions on rooted subtrees that we need to consider as possible parts of optimal contractions in T: One is their size, the other is whether they can be combined with other partial solutions in the rest of T, when growing subtrees towards the root. To quantify this seemingly unwieldy second parameter, we observe that a solution $C \subseteq E$ is feasible if and only if for any two vertices u and v of T we have $\mathrm{load}_{C,\alpha}(u, v) \leq \beta$, where the *load between* u *and* v is defined as

$$\mathrm{load}_{C,\alpha}(u, v) := \mathrm{dist}_{\ell}(u, v)/\alpha - \mathrm{dist}_{\ell_C}(u, v).$$

(recall (1)). For any vertex v of T we further define the *load of* T *at* v as

$$\mathrm{load}_{C,\alpha}(T, v) := \max\{\mathrm{load}_{C,\alpha}(u, v) : u \in V\}.$$

Note that $\mathrm{load}_{C,\alpha}(T, v) \geq 0$, as we have $\mathrm{load}_{C,\alpha}(v, v) = 0$. The following lemma justifies that this definition is the correct second quality parameter.

▶ **Lemma 3.** *Consider a partition of* T *into two subtrees* T_1 *and* T_2 *that only have a vertex* $v \in V$ *in common. Then* $C \subseteq E$ *is a feasible solution for the instance* (T, ℓ, φ) *of the problem* Contraction *if and only if the following two conditions hold:* $C \cap E(T_1)$ *and* $C \cap E(T_2)$ *are feasible solutions for the instances* (T_1, ℓ, φ) *and* (T_2, ℓ, φ) *respectively; and we have* $\mathrm{load}_{C,\alpha}(T_1, v) + \mathrm{load}_{C,\alpha}(T_2, v) \leq \beta$.

Our strategy is to recursively find all contractions on subtrees, that for some fixed size between 1 and n minimize the load. To this end, we choose an arbitrary root vertex r of T,

and starts by considering rooted subtrees consisting of single leaves. We then grows these subtrees towards the root r using two operations: Either two subtrees T_1 and T_2 with the same root v as before are joined (keeping the root v), or a subtree T' containing all successors of its root v in T is extended by adding the edge that leads from v to its parent vertex u in T (in this case, u becomes the new root). Let T^* be the resulting joined or extended subtree arising from the respective operation, and let C be any contraction on T^*. In case of a join-operation we have

$$\mathrm{load}_{C,\alpha}(T^*, v) = \max\{\mathrm{load}_{C,\alpha}(T_1, v), \mathrm{load}_{C,\alpha}(T_2, v)\}, \tag{4a}$$

and the size of C is simply the sum of the sizes of the subsets of C on T_1 and T_2. In case of an extend-operation we have

$$\mathrm{load}_{C,\alpha}(T^*, v) = \begin{cases} \max\{\mathrm{load}_{C,\alpha}(T', u) + \ell(v,u)/\alpha, & \text{if } \{u,v\} \in C, \\ \max\{\mathrm{load}_{C,\alpha}(T', u) - (1-1/\alpha)\ell(v,u), 0\}, & \text{otherwise,} \end{cases} \tag{4b}$$

and the size of C is either equal to the size of the subset of C on T' in the second case, or one more in the first case.

These formulas indicate a monotone behavior of our two parameters, which allows us to compute the necessary partial solutions on T^* by combining the previously computed partial solutions of its subtrees. Furthermore they allow us to compute our parameters for the combined sets.

This yields the dynamic programming algorithm for Contraction referred to in Theorem 2. A similar approach also works for the problem Weak Contraction.

▶ **Theorem 4.** *We can solve* Weak Contraction *on trees with* $\varphi(x) = x/\alpha - \beta$, $\alpha \geq 1$ *and* $\beta \geq 0$ *in time* $\mathcal{O}(n^5)$.

Here, our task is complicated by the fact that the combinability of solutions on subtrees cannot be captured by one single parameter. As we need to keep track of pairs of vertices whose distances remain positive when contracting a set of edges $C \subseteq E$, we define the *weak load* of a rooted tree T at one of its vertices v by

$$\mathrm{wload}_{C,\alpha}(T, v) := \max\{\mathrm{load}_{C,\alpha}(u, v) : u \in V \text{ and } \mathrm{dist}_{\ell_C}(u, v) > 0\},$$

allowing us to formulate the following combinability criterion analogous to Lemma 3 from before.

▶ **Lemma 5.** *Let* T, T_1, T_2 *and* v *be as in Lemma 3. Then* $C \subsetneq E$ *is a feasible solution for the instance* (T, ℓ, φ) *of the problem* Weak Contraction *if and only if the following two conditions hold: For* $i = 1, 2$, *either* C *contains every edge of* T_i *or* $C \cap E(T_i)$ *is a feasible solution for the instance* (T_i, ℓ, φ) *of* Weak Contraction*; and we have*

$$\mathrm{load}_{C,\alpha}(T_1, v) + \mathrm{wload}_{C,\alpha}(T_2, v) \leq \beta \quad \textit{and} \quad \mathrm{wload}_{C,\alpha}(T_1, v) + \mathrm{load}_{C,\alpha}(T_2, v) \leq \beta. \tag{5}$$

We now proceed similarly by computing sets of solutions on rooted subtrees of T that are optimal with respect to the three parameters size, load and weak load. In particular, for any fixed size we compute a Pareto front of subsolutions of that size, minimizing both load and weak load. The key step for getting an efficient algorithm is to prove that these Pareto fronts have polynomial, in fact even linear, size (this is not clear a priori, as the number of feasible solutions on subtrees can be exponential). Using that the weak load has similar monotonicity properties and recursive formulas as stated in (4) for the load, we thus arrive

at an efficient dynamic program. As our algorithm computes $\mathcal{O}(n^2)$ Pareto fronts of size $\mathcal{O}(n)$ at every vertex, and we can combine optimal solutions from two such fronts in time $\mathcal{O}(n)$, we get an additional factor of n^2 in the running time compared to our first dynamic program, giving an overall running time of $\mathcal{O}(n^5)$.

5 Hardness and inapproximability

In this section we state our NP-hardness and inapproximability results for the problems Contraction and Weak Contraction. All proofs throughout this section can be found in [12].

We start by considering the purely additive case, where $\alpha = 1$. Recall that we can compute maximum size (weak) contractions in polynomial time on trees with arbitrary edge lengths (Theorem 2), and on cycles with unit length edges (Theorem 12). In contrast to that, our next result asserts that both problems are NP-hard on cycles with arbitrary edge lengths, even with $\alpha = 1$.

▶ **Theorem 6.** *For any fixed $\beta > 0$, the problems* Contraction *and* Weak Contraction *with tolerance function $\varphi(x) = x - \beta$, $\beta \geq 0$, are NP-hard on cycles.*

It proceeds by a reduction from a variant of the Partition problem. Via inapproximability of the Clique problem (see [21]), we extend this result in the following two ways:

▶ **Theorem 7.** *For all $\beta, \varepsilon > 0$ it is NP-hard to approximate the problem* Contraction *with $\varphi = x - \beta$, $\beta \geq 0$ to within a factor of $n^{1-\varepsilon}$.*

▶ **Theorem 8.** *For all $\varepsilon > 0$ it is NP-hard to approximate* Contraction *with $\varphi = x - 1$ on bipartite graphs with unit lengths ($\ell = 1$) to within a factor of $m^{1/2-\varepsilon}$.*

The next two theorems capture our results for the purely multiplicative case, where $\beta = 0$ (recall that Contraction is trivial in this case). To state the first result, recall that the *girth* of a graph is the length of the shortest cycle.

▶ **Theorem 9.** *For any $g \geq 2$, the problem* Weak Contraction *with tolerance function $\varphi(x) = x/2$, is NP-hard for planar graphs with girth at least $3g$ and unit length edges $\ell = 1$.*

The proof of Theorem 9 uses a reduction from a variant of Planar 3SAT.

▶ **Theorem 10.** *For all $\varepsilon > 0$ it is NP-hard to approximate* Weak Contraction *with $\varphi = 2x/3$ to within a factor of $n^{1-\varepsilon}$.*

The proof of Theorem 10 proceeds via a reduction from Independent Set.

6 Asymptotic bounds

In this section we show how to compute contractions for graphs that are not optimal, but can be computed efficiently despite our hardness results from the previous section. In this vein, the main results of this section are Theorem 11 and the corresponding (not tight) lower bound (Theorem 14). Further we consider the factor by which a contraction can reduce the number of vertices (Theorem 16 and Theorem 17). Throughout this section, we assume all graphs to have unit length edges $\ell = 1$.

▶ **Theorem 11.** *Let $k \geq 1$ be a real number. Any graph G with unit length edges has a $(2k-1, 1)$-contraction C such that the contracted graph G/C has at most $n^{1+1/k}$ edges, and such a contraction can be computed in time $\mathcal{O}(m)$.*

Recall that here and throughout, n and m denote the number of vertices and edges of the input graph G, not of the contracted graph G/C. Setting $k := \log_2 n$ in Theorem 11 yields the following corollary.

▶ **Corollary 12.** *Any graph G with unit length edges has a $(2\log_2 n - 1, 1)$-contraction C such that the contracted graph G/C has at most $2n$ edges, and such a contraction can be computed in time $\mathcal{O}(m)$.*

To prove Theorem 11, we use a clustering approach as presented in [4], yielding the next lemma. For any real number $r \geq 1$, we define an *r-partition* of a graph $G = (V, E)$ as a set of *clusters* $P_i \subseteq V$, $i \in [l]$, with corresponding *cluster centers* $p_i \in P_i$, where the P_i are required to form a partition of the vertex set V and where $\text{dist}_\ell(p_i, u) \leq r - 1$ for all $u \in P_i$ and $i \in [l]$. We denote the resulting r-partition by $P := \{(p_i, P_i) : i \in [l]\}$. We write $\rho(P)$ for the number of pairs $1 \leq i < j \leq l$ for which P_i and P_j are connected by at least one edge, and we refer this quantity as the *density of P*.

▶ **Lemma 13.** *Let $r \geq 1$ be a real number. Any graph G with unit length edges has an r-partition P with density $\rho(P) \leq n^{1+1/r}$, and such a partition can be computed in time $\mathcal{O}(m)$.*

For the proof of Lemma 13 we refer the reader to [12]. With Lemma 13 in hand, we are now ready to prove Theorem 11.

Proof of Theorem 11. Given $G = (V, E)$, we first compute a k-partition P into l clusters as described by Lemma 13. We define the set C of contracted edges as the union of all edges within the clusters, $C := \{\{u, v\} \in E : u, v \in P_i \text{ for some } i \in [l]\}$. We thus contract each cluster into a single vertex and remove from every set of resulting parallel edges all but a single edge.

We proceed to show that C is a $(2k - 1, 1)$-contraction, i.e., we show that $\text{dist}_{\ell_C}(u, v) \geq \text{dist}_\ell(u, v)/(2k - 1) - 1$ for all $u, v \in V$. Consider two vertices $u \in P_i$ and $v \in P_j$, where i and j might be equal. Let $Q_{u,v}$ be the shortest path from u to v in G with edge lengths ℓ_C (all edges from C receive length zero). The length d of $Q_{u,v}$ is the number of edges on that path that connect different clusters. Note that $Q_{u,v}$ enters and leaves each of the $d + 1$ visited clusters at most once, using at most $2k - 2$ edges in every cluster, so in G (where all edges have unit lengths) we get $\text{dist}_\ell(u, v) \leq d + (d + 1)(2k - 2)$.

Combining these observations we obtain

$$\text{dist}_{\ell_C}(u, v) = d \geq d - \frac{1}{2k-1} = \frac{d + (d+1)(2k-2)}{2k-1} - 1 \geq \frac{\text{dist}_\ell(u, v)}{2k-1} - 1,$$

proving the claim. It remains to show that the contracted graph G/C has at most $n^{1+1/k}$ edges, which is an immediate consequence of the upper bound $m(G/C) = \rho(P) \leq n^{1+1/k}$ given by Lemma 13. This completes the proof of the theorem. ◀

Erdős' girth conjecture [13] asserts that there exist graphs with $\Omega(n^{1+1/k})$ edges and girth $2k + 1$. It has been verified for $k = 1, 2, 3, 5$ [20] and the strongest spanner lower bounds depend on it. We use the conjecture to derive the following (not tight) lower bound. For the proof we refer to [12].

▶ **Theorem 14.** *Assuming Erdős' girth conjecture, there exists for any integer $k \geq 2$ a graph G such that any $(k - 1, 1)$-contraction of G results in a graph G/C with $\Omega(n^{1+1/k})$ edges.*

Turning to the case of a purely additive error, we observe two simple $(1, k)$-contractions.

▶ **Theorem 15.** *Let G be a graph with unit length edges.*

(i) *For any even integer $0 \leq k \leq n$, the set of edges incident to the $k/2$ vertices of highest degrees is a $(1, k)$-contraction C in G with $\Phi(C) \geq km/(2n)$.*

(ii) *For any real number $0 < k \leq n$, the set of edges incident to two vertices of degree at least n/k is a $(1, k)$-contraction C in G such that G/C has $\mathcal{O}(n^2/k)$ edges.*

These contractions can be computed in time $\mathcal{O}(m)$.

As mentioned in the introduction, Bernstein and Chechik used the contraction in Theorem 15 2 in their dynamic shortest paths algorithm [8].

Note that the information theoretic lower bound in [1] implies that for all $\varepsilon > 0$, any contraction C such that G/C has $\mathcal{O}(n^{4/3-\varepsilon})$ edges does not admit a constant additive error.

In contrast to spanners, contractions also reduce the number of vertices. Unfortunately, for constant distortion it is impossible to guarantee more than a constant-factor reduction in this parameter, as the example of a path shows. The same problem applies to general dense graphs, since they could still contain a long path within them. That being said, it seems likely that in practice contractions can lead to significant vertex reductions in many dense graphs. We ground this practical intuition with a theoretical result for the special case of graphs with large minimum degree.

▶ **Theorem 16.** *Let D be an integer. Any graph G with unit length edges and minimum degree at least D has a $(5, 1)$-contraction C such that the contracted graph G/C has at most n/D vertices, and such a contraction can be computed in time $\mathcal{O}(m)$.*

To see that we cannot guarantee less than n/D vertices, even with larger approximation error, consider the graph G that consists of n/D isolated D-cliques. We now show that even if G is connected, we cannot guarantee $o(n/D)$ vertices in the contracted graph, even if we allow a larger (constant) approximation error.

▶ **Theorem 17.** *Let D and k be integers. There exists a graph G with minimum degree D such that any $(k, 1)$-contraction C results in a graph G/C with $\Omega(n/(kD))$ vertices.*

The proofs of the two previous theorems are deferred to [12].

Acknowledgements. We thank Martin Skutella for stimulating discussions about the problems treated in this paper. We also thank the anonymous reviewers for their helpful comments and suggestions.

References

1 A. Abboud and G. Bodwin. The 4/3 additive spanner exponent is tight. In *STOC'16—Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing*, pages 351–361. ACM, New York, 2016.

2 Donald Aingworth, Chandra Chekuri, Piotr Indyk, and Rajeev Motwani. Fast estimation of diameter and shortest paths (without matrix multiplication). *SIAM J. Comput.*, 28(4):1167–1181, 1999. doi:10.1137/S0097539796303421.

3 Ingo Althöfer, Gautam Das, David P. Dobkin, Deborah Joseph, and José Soares. On sparse spanners of weighted graphs. *Discrete & Computational Geometry*, 9:81–100, 1993. doi:10.1007/BF02189308.

4 Baruch Awerbuch. Complexity of network synchronization. *J. ACM*, 32(4):804–823, 1985. doi:10.1145/4221.4227.

5 David A. Bader, Henning Meyerhenke, Peter Sanders, and Dorothea Wagner, editors. *Graph Partitioning and Graph Clustering, 10th DIMACS Implementation Challenge Workshop, Georgia Institute of Technology, Atlanta, GA, USA, February 13-14, 2012. Proceedings*, volume 588 of *Contemporary Mathematics*. American Mathematical Society, 2013. doi:10.1090/conm/588.

6 S. Baswana, T. Kavitha, K. Mehlhorn, and S. Pettie. New constructions of (α, β)-spanners and purely additive spanners. In *Proceedings of the Sixteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 672–681. ACM, New York, 2005.

7 Surender Baswana and Sandeep Sen. A simple linear time algorithm for computing a (2k-1)-spanner of $o(n^{1+1/k})$ size in weighted graphs. In Jos C. M. Baeten, Jan Karel Lenstra, Joachim Parrow, and Gerhard J. Woeginger, editors, *Automata, Languages and Programming, 30th International Colloquium, ICALP 2003, Eindhoven, The Netherlands, June 30 - July 4, 2003. Proceedings*, volume 2719 of *Lecture Notes in Computer Science*, pages 384–296. Springer, 2003. doi:10.1007/3-540-45061-0_32.

8 Aaron Bernstein and Shiri Chechik. Deterministic decremental single source shortest paths: beyond the o(mn) bound. 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 389–397. ACM, 2016. doi:10.1145/2897518.2897521.

9 Hubert T.-H. Chan, Donglin Xia, Goran Konjevod, and Andréa W. Richa. A tight lower bound for the steiner point removal problem on trees. In Josep Díaz, Klaus Jansen, José D. P. Rolim, and Uri Zwick, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, 9th International Workshop on Approximation Algorithms for Combinatorial Optimization Problems, APPROX 2006 and 10th International Workshop on Randomization and Computation, RANDOM 2006, Barcelona, Spain, August 28-30 2006, Proceedings*, volume 4110 of *Lecture Notes in Computer Science*, pages 70–81. Springer, 2006. doi:10.1007/11830924_9.

10 Yun Kuen Cheung, Gramoz Goranci, and Monika Henzinger. Graph minors for preserving terminal distances approximately - lower and upper bounds. 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 131:1–131:14. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2016. doi:10.4230/LIPIcs.ICALP.2016.131.

11 E. Chlamtáč, M. Dinitz, G. Kortsarz, and B. Laekhanukit. Approximating spanners and directed steiner forest: Upper and lower bounds. In *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '17, pages 534–553, Philadelphia, PA, USA, 2017. Society for Industrial and Applied Mathematics. URL: http://dl.acm.org/citation.cfm?id=3039686.3039720.

12 Karl Däubel, Yann Disser, Max Klimm, Torsten Mütze, and Frieder Smolny. Distance-preserving graph contractions. *CoRR*, abs/1705.04544, 2017. arXiv:1705.04544.

13 P. Erdős. Extremal problems in graph theory. In *Theory of Graphs and its Applications (Proc. Sympos. Smolenice, 1963)*, pages 29–36. Publ. House Czechoslovak Acad. Sci., Prague, 1964.

14 A. Gupta. Steiner points in tree metrics don't (really) help. In *Proceedings of the Twelfth Annual ACM-SIAM Symposium on Discrete Algorithms (Washington, DC, 2001)*, pages 220–227. SIAM, Philadelphia, PA, 2001.

15 G. Kortsarz. On the hardness of approximation spanners. In *Proceedings of the International Workshop on Approximation Algorithms for Combinatorial Optimization*, APPROX '98, pages 135–146, London, UK, UK, 1998. Springer-Verlag. URL: http://dl.acm.org/citation.cfm?id=646687.756924.

16 Guy Kortsarz and David Peleg. Generating sparse 2-spanners. *J. Algorithms*, 17(2):222–236, 1994. doi:10.1006/jagm.1994.1032.

17 Robert Krauthgamer, Huy L. Nguyen, and Tamar Zondiner. Preserving terminal distances using minors. *SIAM J. Discrete Math.*, 28(1):127–141, 2014. doi:10.1137/120888843.

18 Arthur L. Liestman and Thomas C. Shermer. Additive graph spanners. *Networks*, 23(4):343–363, 1993. doi:10.1002/net.3230230417.

19 David Peleg and Alejandro A. Schäffer. Graph spanners. *Journal of Graph Theory*, 13(1):99–116, 1989. doi:10.1002/jgt.3190130114.

20 Rephael Wenger. Extremal graphs with no C^4's, C^6's, or C^{10}'s. *J. Comb. Theory, Ser. B*, 52(1):113–116, 1991. doi:10.1016/0095-8956(91)90097-4.

21 David Zuckerman. Linear degree extractors and the inapproximability of max clique and chromatic number. *Theory of Computing*, 3(1):103–128, 2007. doi:10.4086/toc.2007.v003a006.

Local Algorithms for Bounded Degree Sparsifiers in Sparse Graphs*

Shay Solomon

IBM Research, Yorktown Heights, NY, USA
solo.shay@gmail.com

Abstract

In graph sparsification, the goal has almost always been of *global* nature: compress a graph into a smaller subgraph (*sparsifier*) that maintains certain features of the original graph. Algorithms can then run on the sparsifier, which in many cases leads to improvements in the overall runtime and memory. This paper studies sparsifiers that have bounded (maximum) degree, and are thus *locally* sparse, aiming to improve local measures of runtime and memory. To improve those local measures, it is important to be able to compute such sparsifiers *locally*.

We initiate the study of local algorithms for bounded degree sparsifiers in unweighted sparse graphs, focusing on the problems of vertex cover, matching, and independent set. Let $\epsilon > 0$ be a slack parameter and $\alpha \geq 1$ be a density parameter. We devise local algorithms for computing:

1. A $(1+\epsilon)$-vertex cover sparsifier of degree $O(\alpha/\epsilon)$, for any graph of *arboricity* α.[1]
2. A $(1+\epsilon)$-maximum matching sparsifier and also a $(1+\epsilon)$-maximal matching sparsifier of degree $O(\alpha/\epsilon)$, for any graph of arboricity α.
3. A $(1+\epsilon)$-independent set sparsifier of degree $O(\alpha^2/\epsilon)$, for any graph of average degree α.

Our algorithms require only a single communication round in the standard message passing models of distributed computing, and moreover, they can be simulated locally in a trivial way. As an immediate application we can extend results from distributed computing and local computation algorithms that apply to graphs of degree bounded by d to graphs of arboricity $O(d/\epsilon)$ or average degree $O(d^2/\epsilon)$, at the expense of increasing the approximation guarantee by a factor of $(1+\epsilon)$. In particular, we can extend the plethora of recent local computation algorithms for approximate maximum and maximal matching from bounded degree graphs to bounded arboricity graphs with a negligible loss in the approximation guarantee.

The inherently local behavior of our algorithms can be used to amplify the approximation guarantee of any sparsifier in time roughly linear in its size, which has immediate applications in the area of dynamic graph algorithms. In particular, the state-of-the-art algorithm for maintaining $(2-\epsilon)$-vertex cover (VC) is at least linear in the graph size, even in dynamic forests. We provide a reduction from the dynamic to the static case, showing that if a t-VC can be computed from scratch in time $T(n)$ in any (sub)family of graphs with arboricity bounded by α, for an arbitrary $t \geq 1$, then a $(t+\epsilon)$-VC can be maintained with update time $\frac{T(n)}{O((n/\alpha)\cdot\epsilon^2)}$, for any $\epsilon > 0$. For planar graphs this yields an algorithm for maintaining a $(1+\epsilon)$-VC with constant update time for any constant $\epsilon > 0$.

1998 ACM Subject Classification G.4. Design and analysis of algorithms

Keywords and phrases arboricity, bounded degree, local algorithm, sparsifier

Digital Object Identifier 10.4230/LIPIcs.ITCS.2018.52

* This work was partially supported by the Herman Goldstine Postdoctoral Fellowship.

[1] In a graph of arboricity α the average degree of any induced subgraph is at most 2α.

1 Introduction

Graph sparsification has been extensively studied for many years, and is subject to increasingly growing interest due to the rapidly growing necessity of dealing with huge-sized graphs. Given such a graph $G = (V, E)$, we would like to *compress* G into a subgraph H of much smaller size that maintains certain features of G, such as distances, cuts or flows. Algorithms can then run on the compressed subgraph H, sometimes called *sparsifier*, rather than the original graph G, which may save significantly on important resources such as the overall runtime and memory of the algorithm, often at the expense of approximate rather than exact solutions or worse approximation guarantees. The most common type of sparsifiers are *edge sparsifiers*, such as graph spanners [49] and cut or spectral sparsifiers [11, 52], which span the original vertex set using a small number of edges. Another well-studied type of sparsifiers are *vertex sparsifiers*, such as flow or cut sparsifiers [32, 43, 39], which should span a small number of designated vertices.

The basic goal in this area has almost always been of *global* nature, i.e., of minimizing the overall size of the sparsifier and the overall time needed for computing it. One of the exceptions is in the area of spanners, where researchers have studied spanners of bounded (maximum) *degree*. While a sparse spanner has a low average degree, and is thus globally sparse, a bounded degree spanner has a low maximum degree, and is thus locally sparse. Although bounded degree spanners have been little studied thus far in general graphs [20, 19], they have been studied extensively in Euclidean low-dimensional spaces, see e.g., [23, 3, 24, 29, 25]). The spanner degree often determines local memory constraints when using spanners to construct network synchronizers [49] and efficient broadcast protocols [5, 6]. In compact routing schemes (e.g., [53, 18]), the use of low degree spanners may enable the routing tables to be of small size. Moreover, viewing vertices as processors, in many applications the degree of a processor represents its *load*, hence a low degree spanner guarantees that the load on all the processors in the network will be low.

This paper studies sparsifiers from a *local* perspective, aiming to improve local measures of runtime and memory. To improve those local measures, it is important to be able to compute such sparsifiers *locally*, in a manner to be defined shortly. We initiate the study of local algorithms for *bounded degree* sparsifiers in unweighted *sparse* graphs. The graphs that we consider are sparse either globally, i.e., of bounded average degree, or uniformly, i.e., of bounded *arboricity*, whence the average degree of any induced subgraph is bounded. In sparse graphs some vertices may have large degrees, as with the n-star graph. Our basic goal is to compute *locally* a sparsifier H for the original graph $G = (V, E)$, whose maximum degree is bounded in terms of the density of G and some slack parameter $\epsilon > 0$, and which approximately preserves a certain property or feature of the original graph; the sparsifier would ideally be a subgraph of G, but this cannot always be achieved. Algorithms, particularly local ones, can then run on the bounded degree sparsifier H rather than on the original graph G, which may save significantly on local resources of runtime and memory. For concreteness, we focus on the following combinatorial optimization problems: (approximate) minimum vertex cover (VC), maximum and maximal matching, and maximum independent set (IS). It would be only natural to extend the study of bounded degree sparsifiers to other fundamental problems.

For the maximum matching problem, a $(1+\epsilon)$-sparsifier for G is a *subgraph* $H = (V', E')$ of G, with $V' \subseteq V, E' \subseteq E$, such that the maximum matching size of H is within a factor of $1+\epsilon$ from that of G; thus a $(1+\epsilon)$-(approximate) maximum matching for H is a $(1+O(\epsilon))$-maximum matching for G. For the maximal matching problem the definition is similar; see Section 2.

We need to be more careful with the definitions of sparsifier for the minimum VC and maximum IS problems, since a VC (respectively, IS) for a subgraph H of G may not be a *valid* VC (resp., IS) for the entire graph G. Note that we are concerned with the validity of solutions obtained by the sparsifier rather than the approximations that they provide. Consequently, a sparsifier in these cases will not be simply a subgraph H of G, but rather a pair (H, V'), where H is a subgraph of G and V' is a vertex set of V, hereafter the *validating set* of the sparsifier, such that for any VC (resp., IS) for H, adding (resp., removing) the validating set V' to (resp., from) it provides a valid VC (resp., IS) for G; the role of the validating set is to translate the solution obtained by the sparsifier into a valid solution for G. We say that (H, V') is a $(1+\epsilon)$-sparsifier for G if for any $(1+\epsilon)$-(approximate minimum) VC (resp., (approximate maximum) IS) for H, denoted by C_H (resp., I_H), the set $C_H \cup V'$ is a valid $(1+O(\epsilon))$-VC (resp., $I_H \setminus V'$ is a $(1+O(\epsilon))$-IS) for G.

Given any $\epsilon > 0$ and any density parameter $\alpha \geq 1$, we devise *local* algorithms for computing:

1. A $(1+\epsilon)$-VC sparsifier of degree $O(\alpha/\epsilon)$, for any graph of *arboricity* bounded by α.
2. A $(1+\epsilon)$-maximum matching sparsifier and also a $(1+\epsilon)$-maximal matching sparsifier of degree $O(\alpha/\epsilon)$, for any graph of arboricity bounded by α.
3. A $(1+\epsilon)$-IS sparsifier of degree $O(\alpha^2/\epsilon)$, for any graph of average degree bounded by α.

Aiming at enhancing the applicability and usefulness of our sparsifiers, we adhere to a strict notion of locality: For any vertex v, we want to be able to compute the adjacent edges of v that belong to the sparsifier by *probing* only v and a small number (bounded by the degree of the sparsifier) of its neighbors, where the probing procedure is context-dependent. In standard centralized settings such a procedure will simply examine the data structures of v and those neighbors, but in the message passing models of distributed computing, for example, the procedure may trigger the exchange of messages between v and those neighbors. Also, we want to determine if a vertex v belongs to the validating set of the sparsifier by probing only v. The advantage in using such a strict notion of locality is three-fold, as summarized here and described in more detail later on:

1. In the rapidly growing area of *local computation algorithms* (see, e.g., [50, 2, 26, 27]), a standard assumption is that the underlying graph has bounded degree. This assumption is required since a local computation algorithm would typically probe all vertices inside a small-radius ball around the queried vertex/edge. If the maximum degree is Δ and the ball radius is r, the probe complexity is bounded by $\Delta^{O(r)}$, and sometimes the total runtime and space will also be bounded by $\Delta^{O(r)}$. Due to the local nature of our sparsification algorithms, we can restrict the probing procedure only to the sparsifier edges, which directly enables us to extend known results from bounded degree graphs to uniformly sparse graphs.
2. In dynamic centralized graph algorithms, following an update of a vertex/edge, the update algorithm would typically scan all neighbors of the updated vertices (and usually more than just those vertices), either to obtain up-to-date information from the data structures of those neighbors or to update that information. Since the adversary may choose to focus its attention on few high degree vertices, this could lead to algorithms with a poor update time. Using our notion of locality, we show that the attention can be restricted to only few edges of the sparsifier, which leads to improvements in the update time.
3. In distributed networks, we can compute the sparsifier in a single communication round. Moreover, since each vertex communicates with only few of its neighbors, the *load* on all vertices (or processors) throughout the sparsifier's computation is low. After the sparsifier has been computed, running on it distributed algorithms rather than on the original

network may significantly reduce the total runtime of the algorithms and the load on the processors.

In addition to the above applications, our sparsification algorithms can be used more broadly in computational models where there are local memory constraints, such as the distributed communication model and the massively parallel computation (MPC) model, which is an abstraction of MapReduce-style frameworks (cf. [21, 4]). Another relevant model is the dynamic distributed model (cf. [46, 17]), where some graph structure (e.g., matching) is to be maintained in a dynamically changing distributed network using low local memory at processors.

1.1 Our sparsifiers

Perhaps the most important feature of our sparsification algorithms is their simplicity, which is partly why they can be computed under such a strict notion of locality.

For any $\Delta \geq 1$, let V^{Δ}_{high} and V^{Δ}_{low} be the sets of vertices of degree $\geq \Delta$ and $< \Delta$, respectively. When Δ is clear from the context, we may omit it from the superscript. For any vertex set V' in G, denote by $G[V']$ the subgraph induced by V'. Define $G_{high} = G[V_{high}]$ and $G_{low} = G[V_{low}]$.

1. For the minimum VC problem, we take the pair (G_{low}, V_{high}) as the $(1+\epsilon)$-sparsifier for G, where G_{low} is a subgraph of G and V_{high} is the validating set of the sparsifier. It is clear that the degree of G_{low} is at most Δ, and moreover, for any VC for G_{low}, its union with V_{high} is a valid VC for G. In Section 3.2 we show that for any graph of arboricity α, taking $\Delta = O(\alpha/\epsilon)$ guarantees the following: If VC_{low} is a $(1+\epsilon)$-VC for G_{low}, then $VC_{low} \cup V_{high}$ is a $(1+O(\epsilon))$-VC for G.
2. For the maximum and maximal matching problems, a (subgraph) $(1+\epsilon)$-sparsifier G_{Δ} for G with degree bounded by Δ can be obtained as follows: Mark up to Δ arbitrary adjacent edges on every vertex v, and add to G_{Δ} all edges that are marked by both endpoints. It is clear that the degree of G_{Δ} is at most Δ. (Note that if we took to G_{Δ} edges that are marked just once, the degree of G_{Δ} could explode.) In Section 3.1 we show that for any graph of arboricity α, taking $\Delta = O(\alpha/\epsilon)$ guarantees that the subgraph G_{Δ} is a $(1+\epsilon)$-sparsifier for G.
3. For the maximum IS problem, we take G_{low} as the $(1+\epsilon)$-sparsifier for G. (Although we may also use a validating set for the sparsifier, there is no need to do that here; thus in this case the IS sparsifier is a subgraph of G.) It is clear that the degree of G_{low} is at most Δ, and moreover, any IS for G_{low} is a valid IS for G. In Section 3.3 we show that for any graph of average degree α, taking $\Delta = O(\alpha^2/\epsilon)$ guarantees that any $(1+\epsilon)$-IS for G_{low} is a $(1+O(\epsilon))$-IS for G.

Note that our sparsifiers are obtained by essentially "ignoring" the high degree vertices, where what is meant by ignoring is context-dependent. For the minimum VC and maximum IS problems, we take all high degree vertices to the VC and take none of them to the IS, respectively, whereas for the maximum and maximal matching problems, we ignore all but at most Δ edges adjacent on any high degree vertex. This approach of ignoring the high degree vertices can be viewed as a general paradigm, and it would be interesting to apply it to additional fundamental graph problems.

1.2 Local computation algorithms

The model of *local computation algorithms* was introduced by Rubinfeld et al. [50], motivated by the fact that it is prohibitively expensive and sometimes infeasible for an algorithm to read

and process the entire input as well as to report the entire output, when dealing with massive data sets. Local computation algorithms should answer *queries* regarding global solutions to computational problems by examining only a small part of the input. The goal is to reach a global solution by performing local (sublinear time) computations on the input, and answer only regarding the queried part of the output. If there are multiple possible solutions, the answers to all queries must be consistent with a single solution. (More technical details on this model are given in Section 2; see also [50].) For the $(1+\epsilon)$-maximum matching problem, each query is an edge in the graph, and the algorithm needs to answer whether the queried edge belongs to a $(1+\epsilon)$-maximum matching; note that the answers to all queries must be with respect to the same matching. [41] devised a randomized local computation algorithm for $(1+\epsilon)$-maximum matching with time and space complexities $\texttt{poly}(\log n) \cdot \exp(\Delta)$, where Δ is the maximum degree of the graph. This result was improved in [26] to a deterministic algorithm with time complexity $O(\log^* n) \cdot \exp(\Delta)$ and zero space complexity. [40] devised a randomized algorithms with time and space complexities of $\texttt{poly}(\log n, \Delta)$. [28] obtained a deterministic algorithm for $(2+\epsilon)$-maximum matching with time complexity $O(\log^* n) \cdot 2^{O(\Delta^2)}$. (We ignore the dependencies on ϵ in the results of [41, 26, 40, 28]; in fact, in some of these results it is assumed that ϵ is constant.)

We can extend the results of [41, 26, 40, 28] from graphs of bounded degree to graphs of bounded arboricity. Specifically, for any graph with arboricity bounded by α, our matching sparsifier G_Δ has a degree bounded by $\Delta = O(\alpha/\epsilon)$. We get this extension by exploiting the local nature of G_Δ, and in particular, the fact that for any vertex v, we can compute the adjacent edges of v that belong to G_Δ by probing only v and at most Δ of its neighbors. Any $(1+\epsilon)$-maximum matching computed for the sparsifier provides a $(1+\epsilon)^2 = (1+O(\epsilon))$-maximum matching for the original graph, thus there is only a negligible loss in the approximation guarantee. Since $\Delta = O(\alpha/\epsilon)$, the smaller ϵ is, the larger the time and space complexities get. Nonetheless, as long as ϵ is not too small, the loss here is quite negligible too. In this way reduce the problem of approximate maximum matching from graphs of arboricity bounded by α to graphs of degree bounded by $\approx \alpha$.

In the same way we reduce the problems of approximate minimum VC and maximum IS from bounded arboricity graphs and graphs of bounded average degree, respectively, to bounded degree graphs. These reductions show that if and when local computation algorithms for these problems are developed in bounded degree graphs (there are currently no such algorithms), they will immediately give rise to new algorithms in the respective wider families. Moreover, this can be viewed as a general paradigm: By locally computing a sparsifier for a combinatorial optimization problem in some family of graphs, we reduce the problem from that family to the family of bounded degree graphs, and the loss depends on the approximation guarantee of the sparsifier and on its degree.

1.3 Dynamic centralized graph algorithms

The problems of dynamically maintaining approximate minimum VC and maximum matching have been intensively studied in recent years, see e.g. [45, 10, 44, 30, 14, 13, 15]. The holy grail is for the approximation guarantee to approach 1 and for the (amortized or worst-case) update time to be $\texttt{poly}(\log n)$ and ideally a constant.

A dynamic algorithm for approximate VC (respectively, matching) should maintain a data structure that answers queries of whether a vertex is in the VC (resp., an edge is matched) or not in *constant* time. Constant query time is considered a standard requirement in this line of research, and the goal is to optimize the update time of the algorithm under this requirement. Almost all related works follow another requirement, of bounding the number

of *changes* to the maintained structure per step, either in the amortized or in the worst-case sense. The update time of the algorithm, which is the time it needs to update the data structure, may be significantly lower than, and is bounded by, the number of changes to the maintained structure; for a motivation of this requirement, refer to [16, 1]. It is easy to see that maintaining an exact minimum VC or a maximum matching requires $\Omega(n)$ changes per update even in the amortized sense, and even for a simple path that changes dynamically in a straightforward way.

Except for general graphs, these problems have been studied mostly in bounded arboricity graphs [44, 38, 34, 12, 13, 48]. It was shown in [44] that a maximal matching can be maintained with amortized time $O(\log n/\log\log n)$ in constant arboricity graphs, and this bound was improved in [34] to $O(\sqrt{\log n})$. [38] achieved a worst-case update time of $O(\log n)$. The algorithms of [44, 34, 38] extend to graphs with arboricity bounded by α, with the update time depending on α. A randomized algorithm for maintaining a maximal matching in *general graphs* with constant amortized update time was given in [51]. A maximal matching provides a 2-approximation for both the maximum matching and the minimum VC.

What about better-than-2 approximations? Improving upon [12, 13] and providing essentially the best result one can hope for in graphs of arboricity α, [48] showed that a $(1+\epsilon)$-maximum matching can be maintained with a worst-case update time of $O(\alpha)$. The $O(\alpha)$ bound in [48] also bounds the number of changes to the matching, and as mentioned it is impossible to maintain an exact matching with $o(n)$ matching changes even for a dynamic path. In addition, [48] showed that a $(2+\epsilon)$-VC can be maintained with a worst-case update time of $O(\alpha)$. This improves the update time of the 2-VC algorithms of [44, 34, 38] in every aspect, at the expense of increasing the approximation guarantee from 2 to $(2+\epsilon)$.

Note that in general graphs, a better-than-2 approximation to the minimum VC cannot be maintained efficiently under the unique games conjecture [37]. Although this hardness result does not apply to bounded arboricity graphs, there is currently no dynamic algorithm for maintaining a better-than-2 approximate VC with update time $o(n)$ even in the amortized sense, and even in dynamic forests![2] In fact, the only known way to maintain a better-than-2 VC dynamically is to apply the fastest static algorithm from scratch following every update step.

The local nature of our sparsification algorithms can be used to amplify the approximation guarantee of our VC sparsifier in time roughly linear in its size. As a corollary, we provide a reduction from the dynamic to the static case, showing that if a t-VC can be computed from scratch in time $T(n)$ in any (sub)family of graphs with arboricity bounded by α, for any $t \geq 1$, then a $(t+\epsilon)$-VC can be maintained with a worst-case update time of $\frac{T(n)}{O((n/\alpha)\cdot\epsilon^2)}$. This bound of $\frac{T(n)}{O((n/\alpha)\cdot\epsilon^2)}$ also bounds the amortized number of changes to the VC. For planar graphs this yields an algorithm for maintaining an $(1+\epsilon)$-VC with a constant worst-case update time for any constant $\epsilon > 0$, which is essentially the best one can hope for. For graphs of arboricity bounded by α we can maintain a VC of approximation guarantee $\approx 2 - \frac{1}{\alpha}$ with a worst-case update time of $O(\sqrt{n}\cdot\alpha^2)$.

We can also amplify our matching and IS sparsifiers and obtain reductions from the dynamic to the static case. These reductions are not useful to obtain new time bounds for the

[2] The only exception is an algorithm for maintaining a maximum matching in dynamic forests with a worst-case update time of $O(\log n)$ [31]. As a result one can maintain the *size* of the minimum VC (by Konig's theorem) in logarithmic update time. On the negative side, one cannot efficiently maintain the VC itself or even a poor approximation of it using [31], and more importantly, the result of [31] requires a logarithmic query time, hence it does not follow the standard constant query time requirement. (We believe that [31] is the only paper in this line of research that does not follow this requirement.)

maximum matching and IS problems. In particular, for approximate matchings, the result of [48] is already the best one can hope for; nevertheless, our reduction for the maximum matching problem can be used to obtain simpler and cleaner algorithms and arguments than those of [48], as discussed in Section 4.2.

1.4 Distributed networks

Our sparsification algorithms can be implemented within a single communication round in distributed networks, where each processor sends and receives a single $O(1)$-bit message along each of its adjacent edges. Moreover, if Δ is the maximum degree of the sparsifier, each processor may send messages along just Δ of its adjacent edges, which ensures that the *load* on all the processors will be low throughout the sparsifier's computation. After the sparsifier has been computed, we can run on it the required distributed algorithm rather than on the original network, which may significantly reduce the total runtime of the algorithm, the load on the processors, and in some settings it may also reduce the local memory usage at a processor.

Our distributed sparsification algorithms directly extend results from bounded degree graphs to bounded arboricity graphs or to graphs of bounded average degree, for all the problems studied in this paper. Since the performance of many distributed algorithm depend on the maximum degree of the underlying network and as our sparsification algorithms are extremely simple, we anticipate that they will be used and implemented in practice.

For the distributed approximate VC problem, [8] showed how to compute a $(2+\epsilon)$-VC in $O(\log\Delta/(\epsilon\log\log\Delta))$ rounds, where Δ is the maximum degree in the graph. We can plug our reduction to extend the result of [8] to graphs of arboricity bounded by α, getting a $(2+\epsilon)$-VC in $O(\log(\alpha/\epsilon)/(\epsilon\log\log(\alpha/\epsilon)))$ rounds.

For the distributed approximate matching problem, a reduction from bounded arboricity graphs to bounded degree graphs was already given in [22]. Nonetheless, our reduction has several advantages over that of [22] (see Section 4), one of which is that it is much simpler, another is that our degree bound has better dependence on ϵ. In particular, [27] devised a distributed algorithm for computing a $(1+\epsilon)$-maximum matching in $\Delta^{O(1/\epsilon)}+O(\epsilon^{-2})\cdot\log^* n$ rounds. Plugging our reduction (instead of that from [22]), we easily extend the result of [27] to graphs of arboricity bounded by α to get a $(1+\epsilon)$-maximum matching in $(\alpha/\epsilon)^{O(1/\epsilon)}+O(\epsilon^{-2})\cdot\log^* n$ rounds.

A reduction from bounded arboricity graphs to bounded degree graphs was given in [9] for the problems of maximal matching, maximal IS, vertex coloring and ruling sets. The reduction of [9] is based on different ideas than ours (their algorithm is randomized, the number of rounds required by their algorithm is polylogarithmic in the maximum degree, etc), and moreover, it appears that the reduction of [9] cannot be efficiently applied to the problems studied in this paper.

1.5 Organization

The definitions and notation used throughout are given in Section 2. Our matching, VC and IS sparsifiers are presented in Sections 3.1, 3.2 and 3.3, respectively. Finally, in Section 4 we provide some applications of our sparsifiers.

2 Preliminaries

Consider an unweighted graph $G = (V, E)$. A matching M for G is said to be *almost-maximal* w.r.t. some parameter $\eta > 0$, or *η-maximal*, if at most $\eta \cdot |M|$ edges can be added to it while keeping it a valid matching for G. A $(1+\epsilon)$-maximal matching sparsifier for G is a subgraph H of G, such that any η-maximal matching for H is an $(\epsilon + O(\eta))$-maximal matching for G; in particular, a maximal matching for H is ϵ-maximal for G, and a ϵ-maximal matching for H is $O(\epsilon)$-maximal for G.

For a vertex v in G, let $\Gamma_G(v)$ denote the set of neighbors (or *neighborhood*) of v in G. For any vertex set $V' \subseteq V$ in G, denote by $G[V']$ the subgraph induced by V'.

A graph $G = (V, E)$ has *arboricity* α if $\alpha = \max_{U \subseteq V} \left\lceil \frac{|E(U)|}{|U|-1} \right\rceil$, where $E(U) = \{(u, v) \in E \mid u, v \in U\}$. Alternatively, the arboricity of a graph is the minimum number of edge-disjoint forests into which it can be partitioned. The family of bounded arboricity graphs contains planar and bounded genus graphs, bounded tree-width graphs, and in general all graphs excluding fixed minors.

As mentioned in Section 1.1, for any $\Delta \geq 1$, we write V^{Δ}_{high} and V^{Δ}_{low} to denote the sets of vertices of degree $\geq \Delta$ and $< \Delta$, respectively, omitting Δ from the superscript when it is clear from the context. Define $G_{high} = G[V_{high}]$ and $G_{low} = G[V_{low}]$.

3 Our Sparsifiers

Note that a graph with arboricity α has an average degree at most 2α. Throughout we use α as an arboricity parameter and β as an average degree parameter. The next observation will be useful.

▶ **Observation 1.** *Let $G = (V_1 \cup V_2, E)$ be a graph with average degree bounded by β, and suppose that each vertex of V_1 has degree at least $(c+1)\beta$, for any c. Then $|V_1| \leq |V_2|/c$.*

Proof. Observe that $2|E| \leq \beta(|V_1| + |V_2|)$. Since every vertex in U has degree at least $(c+1)\beta$, we have $2|E| \geq |V_1| \cdot (c+1)\beta$, hence $|V_1| \cdot (c+1)\beta \leq \beta(|V_1| + |V_2|)$, and so $|V_1| \leq |V_2|/c$. ◀

3.1 The matching sparsifier

Let G be a graph of arboricity bounded by α, set $\Delta = 5(5/\epsilon + 1)2\alpha$, and define the sets V_{high}, V_{low} and the subgraphs G_{low}, G_{high} accordingly. We assume that $\epsilon \leq 1$; the argument works also for larger ϵ, by increasing Δ appropriately. (We did not try to optimize the constants in the definition of Δ.)

Recall our definition of the matching sparsifier G_Δ: Mark up to Δ arbitrary adjacent edges on every vertex v, and add to G_Δ all edges that are marked by both endpoints. It is clear that the degree of G_Δ is at most Δ. To prove that G_Δ is a matching sparsifier, we use Hall's marriage theorem.

▶ **Theorem 2** (Hall's marriage theorem [33])**.** *Let G be a bipartite graph with sides X and Y. There is a matching that entirely covers X if and only if for every subset W of X, $|W| \leq |\Gamma_G(W)|$, where $\Gamma_G(W) = \bigcup_{v \in W} \Gamma_G(v)$ is the* neighborhood *of W.*

The following theorem shows that G_Δ is a $(1+\epsilon)$-maximum matching sparsifier.

▶ **Theorem 3.** *Let G be a graph of arboricity bounded by α and define G_Δ as above, for $\Delta = 5(5/\epsilon + 1)2\alpha, \epsilon \leq 1$. Also, denote by $\mathcal{M}^*$ and $\mathcal{M}^*_\Delta$ the maximum matchings for G and G_Δ, respectively. Then $|\mathcal{M}^*| \leq (1+\epsilon) \cdot |\mathcal{M}^*_\Delta|$. (In particular, any t-matching for G_Δ is a $t(1+\epsilon)$-matching for G.)*

Proof. We shall construct a matching $\mathcal{M}_\Delta$ for G_Δ satisfying $|\mathcal{M}^*| \leq (1+\epsilon) \cdot |\mathcal{M}_\Delta|$.

Let $\mathcal{M}_1^*$ be the subset of $\mathcal{M}^*$ of all edges that belong to G_Δ, and initialize $\mathcal{M}_\Delta$ to $\mathcal{M}_1^*$. Define $\mathcal{M}_2^* = \mathcal{M}^* \setminus \mathcal{M}_1^*$ as the complementary subset of $\mathcal{M}^*$. We next show that sufficiently many additional edges of G_Δ can be added to $\mathcal{M}_\Delta$ while keeping it a valid matching.

Note that any edge with two endpoints in V_{low} must belong to G_Δ. Since the edges of $\mathcal{M}_2^*$ do not belong to G_Δ by definition, any edge of $\mathcal{M}_2^*$ is adjacent on at least one vertex of V_{high}.

Let $V^{in} = V_{high}^{in}$ be the subset of V_{high} of all vertices with at least $2\Delta/5$ neighbors in V_{high}, and let $V^{out} = V_{high}^{out} = V_{high} \setminus V^{in}$ be the complementary subset of V_{high}. Observe that G_{high} has arboricity at most α, and thus average degree at most 2α. Moreover, every vertex in V^{in} has degree at least $2\Delta/5 = 2(5/\epsilon+1)2\alpha \geq (10/\epsilon+1)2\alpha$ in G_{high}. Observation 1 thus yields

$$|V^{in}| \;\leq\; \epsilon/10 \cdot |V^{out}|. \tag{1}$$

▶ **Observation 4.** *Each vertex in V^{out} has more than $3\Delta/5$ neighbors in V_{low} within G_Δ.*

Proof. First, note that any vertex in V_{low} marks all its $< \Delta$ adjacent edges. Since each vertex in V^{out} has $< 2\Delta/5$ neighbors in V_{high} but a degree of $\geq \Delta$, the remaining $> 3\Delta/5$ neighbors must be in V_{low}. Each vertex of V^{out} marks Δ edges, and any of the $> 3\Delta/5$ edges adjacent to a vertex of V_{low} is also marked by that endpoint in V_{low}, and is thus added to G_Δ. ◀

For a vertex v, we will refer to its neighbors within G_Δ as its G_Δ-*neighbors*. Let $U = U^{out}$ be the set of vertices in V^{out} that are free w.r.t. $\mathcal{M}_1^*$. By Observation 4, each vertex of U has more than $3\Delta/5$ G_Δ-neighbors in V_{low}. Denote by $\Gamma = \Gamma(U)$ the set of all G_Δ-neighbors of vertices from U in V_{low}. We next partition Γ into two sets, the sets $\Gamma_{matched}$ and Γ_{free} of vertices that are matched and free w.r.t. $\mathcal{M}_1^*$, respectively. A vertex $u \in U$ is called *risky* if at least $2\Delta/5$ of its G_Δ-neighbors in Γ are in $\Gamma_{matched}$, otherwise it is *safe*, and then more than $\Delta/5$ of its G_Δ-neighbors are in Γ_{free}. Let U_{risky} and U_{safe} be the sets of all risky and safe vertices of U, respectively.

▶ **Claim 5.** $|U_{risky}| \leq \epsilon/5 \cdot |\mathcal{M}_1^*|$.

Proof. For each vertex $u \in U_{risky}$, let $\Gamma_{matched}(u)$ be the set of its at least $2\Delta/5$ G_Δ-neighbors in $\Gamma_{matched}$. Let $\Gamma_{risky} = \bigcup_{u \in U_{risky}} \Gamma_{matched}(u)$ denote the union of the sets $\Gamma_{matched}(u)$ over all vertices $u \in U_{risky}$. Since all vertices in Γ_{risky} are matched w.r.t. the matching $\mathcal{M}_1^*$, $|\Gamma_{risky}| \leq 2|\mathcal{M}_1^*|$. Observe that the subgraph G_{risky} of G_Δ induced by $U_{risky} \cup \Gamma_{risky}$ has arboricity at most α, and thus average degree at most 2α. Moreover, each vertex in U_{risky} has at least $2\Delta/5$ neighbors in Γ_{risky}, and thus its degree in G_{risky} is at least $2\Delta/5 = 2(5/\epsilon+1)2\alpha \geq (10/\epsilon+1)2\alpha$. Observation 1 thus yields $|U_{risky}| \;\leq\; \epsilon/10 \cdot |\Gamma_{risky}| \leq \epsilon/5 \cdot |\mathcal{M}_1^*|$. ◀

Note that any edge in G_Δ between a vertex in U_{safe} and a vertex in Γ_{free} is vertex-disjoint to all edges of $\mathcal{M}_1^*$. Consequently, the following claim implies that at least $|U_{safe}|$ edges of G_Δ can be added to the matching $\mathcal{M}_\Delta$ while preserving its validity.

▶ **Claim 6.** *There exists a matching $\mathcal{M}_{safe}$ in G_Δ that entirely covers U_{safe} by edges between U_{safe} and Γ_{free}. In particular, $|\mathcal{M}_{safe}| = |U_{safe}|$.*

Proof. For each vertex $u \in U_{safe}$, let $\Gamma_{free}(u)$ be the set of its at least $\Delta/5$ G_Δ-neighbors in Γ_{free}. Let $\Gamma_{safe} = \bigcup_{u \in U_{safe}} \Gamma_{free}(u)$ denote the union of the sets $\Gamma_{free}(u)$ over all vertices $u \in U_{safe}$. Consider the induced bipartite subgraph G_{safe} of G_Δ with sides U_{safe} and Γ_{safe}. We argue that for any subset $W \subseteq U_{safe}$, its neighborhood in G_{safe}, namely,

$$\Gamma_{G_{safe}}(W) \;=\; \bigcup_{v \in W} \Gamma_{G_{safe}}(v) \;=\; \bigcup_{v \in W} \Gamma_{free}(v),$$

is of larger size. Obsserve that the subgraph G^W_{safe} of G_{safe} induced by the vertex set $W \cup \Gamma_{G_{safe}}(W)$ has arboricity at most α, and thus average degree at most 2α. Moreover, each vertex of W has at least $\Delta/5$ neighbors in $\Gamma_{G_{safe}}(W)$, i.e., its degree in G^W_{safe} is at least $\Delta/5 = (5/\epsilon + 1)2\alpha$. Observation 1 thus yields $|W| \leq \epsilon/5 \cdot |\Gamma_{G_{safe}}(W)| < |\Gamma_{G_{safe}}(W)|$. By Hall's marriage theorem (Theorem 2), there exists a matching $\mathcal{M}_{safe}$ in G_{safe} that entirely covers U_{safe}. Claim 6 follows. ◀

We add all edges in the matching $\mathcal{M}_{safe}$ guaranteed by Claim 6 to $\mathcal{M}_\Delta$, so that $\mathcal{M}_\Delta = \mathcal{M}^*_1 \cup \mathcal{M}_{safe}$. Since $\mathcal{M}^*$ is a maximum matching for G that is a disjoint union of $\mathcal{M}^*_1$ and $\mathcal{M}^*_2$ and as $\mathcal{M}_\Delta$ is a disjoint union of $\mathcal{M}^*_1$ and $\mathcal{M}_{safe}$, it follows that $|\mathcal{M}^*_2| \geq |\mathcal{M}_{safe}|$. Although the vertices of V^{in} may be matched w.r.t. $\mathcal{M}^*_2$, we have $|V^{in}| \leq \epsilon/10 \cdot |V^{out}|$ by Equation (1). By definition, all vertices of $V^{out} \setminus U$ are matched w.r.t. $\mathcal{M}^*_1$, thus $|V^{out} \setminus U| \leq 2|\mathcal{M}^*_1||$. Hence $|V^{out}| \leq 2|\mathcal{M}^*_1| + |U|$, and so

$$|V^{in}| \;\leq\; \epsilon/10 \cdot |V^{out}| \;\leq\; \epsilon/10 \cdot (2|\mathcal{M}^*_1| + |U_{risky}| + |U_{safe}|). \tag{2}$$

Since any edge of $\mathcal{M}^*_2$ is adjacent on at least one vertex of V_{high} and as all vertices of $V^{out} \setminus U$ are matched w.r.t. $\mathcal{M}^*_1$, $|\mathcal{M}^*_2| \leq |V^{in}| + |U_{risky}| + |U_{safe}|$. Combined with Equation (2) and Claim 5,

$$\begin{aligned}
|\mathcal{M}^*_2| &\leq |V^{in}| + |U_{risky}| + |U_{safe}| \;\leq\; \epsilon/10 \cdot (2|\mathcal{M}^*_1| + |U_{risky}| + |U_{safe}|) + |U_{risky}| + |U_{safe}| \\
&\leq \epsilon/10 \cdot (2|\mathcal{M}^*_1| + \epsilon/5 \cdot |\mathcal{M}^*_1| + |M^*_2|) + \epsilon/5 \cdot |\mathcal{M}^*_1| + |M_{safe}| \\
&= (\epsilon/5 + \epsilon^2/50 + \epsilon/5) \cdot |\mathcal{M}^*_1| + \epsilon/10 \cdot |M^*_2| + |M_{safe}| \\
&\leq \epsilon/2 \cdot |M^*_1| + \epsilon/10 \cdot |M^*_2| + |M_{safe}|,
\end{aligned}$$

yielding

$$\begin{aligned}
|\mathcal{M}_\Delta| &= |\mathcal{M}^*_1| + |\mathcal{M}_{safe}| \;\geq\; |\mathcal{M}^*_1| + |\mathcal{M}^*_2| - \epsilon/2 \cdot |\mathcal{M}^*_1| - \epsilon/10 \cdot |\mathcal{M}^*_2| \\
&\geq (1 - \epsilon/2) \cdot (|\mathcal{M}^*_1| + |\mathcal{M}^*_2|) \;=\; (1 - \epsilon/2) \cdot |\mathcal{M}^*|.
\end{aligned}$$

To complete the proof of Lemma 3, observe that

$$|\mathcal{M}^*| \leq \frac{1}{1 - \epsilon/2} \cdot |\mathcal{M}_\Delta| \;\leq\; (1 + \epsilon) \cdot |\mathcal{M}_\Delta| \;\leq\; (1 + \epsilon) \cdot |\mathcal{M}^*_\Delta|. \qquad ◀$$

The following theorem shows that G_Δ is a $(1 + \epsilon)$-maximal matching sparsifier.

▶ **Theorem 7.** *Let G be a graph of arboricity bounded by α and G_Δ defined as above, for $\Delta = 5(5/\epsilon + 1)2\alpha, \epsilon \leq 1$. Any η-maximal matching for G_Δ is an $(\epsilon + 3\eta)$-maximal matching for G, for any $\eta > 0$.*

Proof. The proof of this theorem is similar to that of Theorem 3, thus we aim for conciseness.

Consider an arbitrary η-maximal matching $\mathcal{M}_\Delta = \mathcal{M}^\eta_\Delta$ for G_Δ, and let $\mathcal{M}'_\Delta$ be any matching for G obtained by adding edges to $\mathcal{M}_\Delta$. We will show that $|\mathcal{M}'_\Delta \setminus \mathcal{M}_\Delta| \leq (\epsilon + 2\eta) \cdot |\mathcal{M}_\Delta|$.

Since $\mathcal{M}_\Delta$ is η-maximal w.r.t. G_Δ, at most $\eta \cdot |\mathcal{M}_\Delta|$ edges of $\mathcal{M}'_\Delta \setminus \mathcal{M}_\Delta$ may belong to G_Δ; in what follows we refer to those edges as *special edges* and to the remaining edges of $\mathcal{M}'_\Delta \setminus \mathcal{M}_\Delta$ as ordinary edges. Denote the set of ordinary edges in $\mathcal{M}'_\Delta \setminus \mathcal{M}_\Delta$ by O. Since any edge with two endpoints in V_{low} belongs to G_Δ, it follows that any edge of O has at least one endpoint in V_{high}. Denote the set of vertices in V_{high} that are free w.r.t. $\mathcal{M}_\Delta$ by F_{high}, and note that $|O| \leq |F_{high}|$.

Defining the sets V^{in} and V^{out} as in the proof of Theorem 3, Equation (1) yields $|V^{in}| \leq \epsilon/10 \cdot |V^{out}|$. Notice that $|V^{out}| \leq 2|\mathcal{M}_\Delta| + |F_{high} \cap V^{out}|$, hence

$$|F_{high} \cap V^{in}| \;\leq\; |V^{in}| \;\leq\; \epsilon/10 \cdot |V^{out}| \;\leq\; \epsilon/10 \cdot (2|\mathcal{M}_\Delta| + |F_{high} \cap V^{out}|). \tag{3}$$

Observation 4 from the proof of Theorem 3 remains valid, and it implies that each vertex in $F_{high} \cap V^{out}$ has more than $3\Delta/5$ G_Δ-neighbors in V_{low}. Denote by $\Gamma = \Gamma(F_{high} \cap V^{out})$ the set of all G_Δ-neighbors of vertices from $F_{high} \cap V^{out}$ in V_{low}. We next partition Γ into two sets, the sets $\Gamma_{matched}$ and Γ_{free} of vertices in Γ that are matched and free w.r.t. $\mathcal{M}_\Delta$, respectively. A vertex $f \in F_{high} \cap V^{out}$ is called *risky* if at least $2\Delta/5$ of its G_Δ-neighbors are in $\Gamma_{matched}$, otherwise it is *safe*, and then more than $\Delta/5$ of its G_Δ-neighbors are in Γ_{free}. Let F_{risky} and F_{safe} be the sets of all risky and safe vertices of $F_{high} \cap V^{out}$, respectively. Following similar lines as those in the proof of Claim 5, we get $|F_{risky}| \leq \epsilon/5 \cdot |\mathcal{M}_\Delta|$. Also, following similar lines as those in the proof of Claim 6, we establish the existence of a matching $\mathcal{M}_{safe}$ in G_Δ that entirely covers F_{safe} by edges between F_{safe} and Γ_{free}. Since all edges of $\mathcal{M}_{safe}$ belong to G_Δ and can be added to $\mathcal{M}_\Delta$ while keeping it a valid matching for G_Δ, the fact that $\mathcal{M}_\Delta$ is η-maximal w.r.t. G_Δ yields $|F_{safe}| \leq \eta \cdot |\mathcal{M}_\Delta|$. It follows that $|F_{high} \cap V^{out}| = |F_{risky}| + |F_{safe}| \leq \epsilon/5 \cdot |\mathcal{M}_\Delta| + \eta \cdot |\mathcal{M}_\Delta|$. Recall that $|O| \leq |F_{high}|$. Combined with Equation (3), we conclude that

$$\begin{aligned}
|O| \;&\leq\; |F_{high}| \;=\; |F_{high} \cap V^{in}| + |F_{high} \cap V^{out}| \\
&\leq\; \epsilon/10 \cdot (2|\mathcal{M}_\Delta| + |F_{high} \cap V^{out}|) + \epsilon/5 \cdot |\mathcal{M}_\Delta| + \eta \cdot |\mathcal{M}_\Delta| \\
&\leq\; \epsilon/10 \cdot (2|\mathcal{M}_\Delta| + \epsilon/5 \cdot |\mathcal{M}_\Delta| + \eta \cdot |\mathcal{M}_\Delta|) + \epsilon/5 \cdot |\mathcal{M}_\Delta| + \eta \cdot |\mathcal{M}_\Delta| \\
&\leq\; (\epsilon/5 + \epsilon^2/50 + \epsilon \cdot \eta/10 + \epsilon/5 + \eta) \cdot |\mathcal{M}_\Delta| \;<\; (\epsilon + 2\eta) \cdot |\mathcal{M}_\Delta|,
\end{aligned}$$

i.e., the number $|O|$ of ordinary edges in $\mathcal{M}'_\Delta \setminus \mathcal{M}_\Delta$ is at most $(\epsilon + 2\eta) \cdot |\mathcal{M}_\Delta|$. Recall that the number of special edges in $\mathcal{M}'_\Delta \setminus \mathcal{M}_\Delta$ is at most $\eta \cdot |\mathcal{M}_\Delta|$, thus $|\mathcal{M}'_\Delta \setminus \mathcal{M}_\Delta| \leq (\epsilon + 3\eta) \cdot |\mathcal{M}_\Delta|$. ◀

3.2 The VC sparsifier

Let G be a graph of arboricity bounded by α, set $\Delta = (1/\epsilon + 1) \cdot 2\alpha$, and define the sets V_{high}, V_{low} and the subgraph G_{low} accordingly. To prove that the pair (G_{low}, V_{high}) is a VC sparsifier, we use the next lemma.

▶ **Lemma 8.** *Let VC be an arbitrary VC for G, and let $U = U_{high}$ be the set of vertices in V_{high} that are not in VC. Then $|U| \leq \epsilon \cdot |VC|$.*

Proof. Denote by $\Gamma = \Gamma_{high}$ the set of neighbors in $G \setminus U$ of all vertices of U, and note that $\Gamma \subseteq VC$, as otherwise VC cannot be a VC for G. Observe that the subgraph $G' = (U \cup \Gamma, E') = G[U \cup \Gamma]$ induced by $U \cup \Gamma$ has arboricity at most α, and thus average degree at most 2α. Moreover, every vertex in U has degree at least Δ. Observation 1 thus yields $|U| \;\leq\; \epsilon \cdot |\Gamma| \;\leq\; \epsilon \cdot |VC|$. ◀

The following theorem shows that (G_{low}, V_{high}) is a $(1 + \epsilon)$-VC sparsifier.

▶ **Theorem 9.** *Let G be a graph of arboricity bounded by α. Also, let VC^* be a minimum VC for G, let VC^t_{low} be a t-VC for G_{low}, for any $t \geq 1$, and define $\widetilde{VC} = VC^t_{low} \cup V_{high}$. Then $\widetilde{VC}$ is a $(t+\epsilon)$-VC for G.*

Proof. Since VC^* is a VC for G, $VC^* \cap V_{low}$ must be a VC for G_{low}. Hence $|VC^t_{low}| \leq t \cdot |VC^* \cap V_{low}|$. Denoting by $U = U_{high}$ the set of vertices in V_{high} that are not in VC^*, we have $|V_{high}| = |VC^* \cap V_{high}| + |U|$. Also, Lemma 8 yields $|U| \leq \epsilon \cdot |VC^*|$. Since $|VC^t_{low}| \leq t \cdot |VC^* \cap V_{low}|$, it follows that

$$\begin{aligned}|\widetilde{VC}| = |VC^t_{low}| + |V_{high}| &= |VC^t_{low}| + |VC^* \cap V_{high}| + |U| \\ &\leq t \cdot |VC^* \cap V_{low}| + |VC^* \cap V_{high}| + \epsilon \cdot |VC^*| \leq (t+\epsilon) \cdot |VC^*|.\end{aligned}$$

◀

3.3 The IS sparsifier

Let G be a graph of average degree bounded by β (the arboricity may be much larger than β). Set $\Delta = ((\beta+1)/\epsilon + 1) \cdot \beta$, and define the sets V_{high}, V_{low} and the subgraph G_{low} accordingly. We assume that $\beta \geq 1$, as we may ignore isolated vertices (adding all of them to the independent set). To show that G_{low} is an IS sparsifier, we make the following observation.

▶ **Observation 10.** *Let V_1 be any set of vertices in an arbitrary graph $G = (V, E)$, and let V_2 be the complementary subset of V. Let IS^*, IS^*_1 and IS^*_2 be maximum independent sets for the graph G and its subgraphs $G[V_1]$ and $G[V_2]$, respectively. Then $|IS^*_1| + |IS^*_2| \geq |IS^*|$.*

Proof. Both $IS^* \cap V_1$ and $IS^* \cap V_2$ are ISs, hence $|IS^* \cap V_1| \leq |IS^*_1|, |IS^* \cap V_2| \leq |IS^*_2|$. ◀

The following theorem shows that G_{low} is an $(1+\epsilon)$-IS sparsifier

▶ **Theorem 11.** *Let G be a graph of average degree bounded by β. Also, let IS^* (respectively, IS^*_{low}) be a maximum IS for G (resp., G_{low}), let IS^t_{low} be a t-IS for G_{low}, for any $t \geq 1$. Then IS^t_{low} is a $t(1+\epsilon)$-IS for G, for any $\epsilon < \beta$.*

Proof. Since IS^t_{low} is an IS for G_{low}, it must also be an IS for G.

Since the average degree in $G = (V_{low} \cup V_{high}, E)$ is bounded by β and as every vertex in V_{high} has degree at least $\Delta = ((\beta+1)/\epsilon + 1) \cdot \beta$, Observation 1 implies that $|V_{high}| \leq \epsilon \cdot (|V_{low}|/(\beta+1))$.

Since $\epsilon < \beta$, we have $\Delta = ((\beta+1)/\epsilon + 1) \cdot \beta > \beta$, hence there is at least one vertex of degree less than Δ, i.e., V_{low} is non-empty. Denote the average degree in G_{low} by β_{low}. Since every vertex in V_{high} has degree at least $\Delta = ((\beta+1)/\epsilon + 1) \cdot \beta$, we have $|V_{high}| \cdot ((\beta+1)/\epsilon + 1) \cdot \beta + |V_{low}| \cdot \beta_{low} \leq 2|E| \leq \beta(|V_{low}| + |V_{high}|)$. It follows that $|V_{high}| \cdot ((\beta+1)/\epsilon) \leq |V_{low}|(1 - \frac{\beta_{low}}{\beta})$, which, together with the fact that V_{low} is non-empty, yields $\beta_{low} \leq \beta$. By Turan's theorem, $|IS^*_{low}| \geq V_{low}/(\beta_{low}+1)$, hence

$$|V_{high}| \leq \epsilon \cdot (|V_{low}|/(\beta+1)) \leq \epsilon \cdot (|V_{low}|/(\beta_{low}+1)) \leq \epsilon \cdot |IS^*_{low}|.$$

By Observation 10, $|IS^*| \leq |V_{high}| + |IS^*_{low}|$, so $|IS^*| \leq (1+\epsilon) \cdot |IS^*_{low}| \leq t(1+\epsilon) \cdot |IS^t_{low}|$. ◀

4 Applications

4.1 Local computation algorithms

Each vertex v is represented as a unique ID from $\{1, \ldots, n\}$, and the graph $G = (V, E)$ is given through an adjacency list oracle $\mathcal{O}_G$ that answers neighbor queries: given a vertex

$v \in V$ and an index i, the ith neighbor of v is returned if v's degree is $\geq i$; otherwise a null sign is returned. Consider first our matching sparsifier G_Δ, obtained by marking up to Δ arbitrary adjacent edges on every vertex v, and adding to G_Δ all edges that are marked by both endpoints. Recall that $\Delta = O(\alpha/\epsilon)$, where α is a bound on the arboricity of G. Our goal is to simulate the execution of any local computation algorithm for approximate matching entirely within our bounded degree sparsifier, and in this way to reduce the problem from bounded arboricity graphs to bounded degree graphs. To this end we simply need to be *consistent* about the adjacent edges of a vertex v that we mark, e.g., for every vertex v, mark its first Δ neighbors on its adjacency list. Whenever a vertex is queried/probed, there is no need to probe any other neighbor of this vertex besides the first Δ on its adjacency list, since none of the edges that lead to the other neighbors is in the sparsifier. To determine which among these neighbors is also its neighbor in the sparsifier, we perform a symmetric probe for each of the (at most) Δ neighbors. In this way any probing procedure of the original graph (which may contain high degree vertices) is restricted to the matching sparsifier, at the cost of increasing the time complexity by at most a factor of Δ; this loss is considered negligible, since all the time and space complexities of algorithms in this area are anyway at least polynomial in Δ. Moreover, by Theorems 3 and 7, any $(1+\epsilon)$-maximum (respectively, ϵ-maximal) matching computed for G_Δ provides a $(1+O(\epsilon))$-maximum (resp., $O(\epsilon)$-maximal) matching for the original graph, thus there is only a negligible loss in the approximation guarantee. We remark that the local computation algorithm of [28] is actually an almost-maximal matching algorithm, and the $(2+\epsilon)$-approximation guarantee holds as any ϵ-maximal matching is also a $(2+O(\epsilon))$-maximum matching. In this way we reduce the problems of approximate-maximum matching and almost-maximal matching from graphs of arboricity bounded by α to graphs of degree bounded by $O(\alpha/\epsilon)$.

▶ **Corollary 12.** *For any graph G with arboricity bounded by α and for any constant $\epsilon > 0$:*

- *There is a deterministic local computation algorithm for $(1+\epsilon)$-maximum matching with time complexity $O(\log^* n) \cdot \exp(\alpha)$ and zero space complexity. (An extension of [26].)*
- *There is a randomized local computation algorithms for $(1+\epsilon)$-maximum matching with time and space complexities of $\mathtt{poly}(\log n, \alpha)$. (An extension of [40].)*
- *There is a deterministic local computation algorithm for $(2+\epsilon)$-maximum matching with time complexity $O(\log^* n) \cdot 2^{O(\alpha^2)}$. (An extension of [28].)*

For our VC and IS sparsifiers, things are even simpler. The IS sparsifier is simply G_{low}, i.e., the subgraph induced on the low degree vertices, hence simulating the execution of a local computation algorithm entirely within the sparsifier can be naturally done without having to probe more than Δ neighbors of any vertex. As for the VC sparsifier, we also need to reason about the validating set V^{high}, but for any vertex we can determine if it belongs to V^{high} or not by making one query to the oracle $\mathcal{O}_G$.

4.2 Dynamic centralized algorithms

In this section we employ our sparsifiers to get efficient dynamic algorithms. The starting point is a "lazy scheme" due to [30] for maintaining approximate maximum matching for general graphs, which was refined for bounded arboricity graphs in [48]. This scheme exploits a basic *stability* property of matchings: The size of the maximum matching changes by at most 1 following each update. Thus if we have a *large* matching of size close to the maximum, it will remain close to it throughout a long update sequence, or formally:

▶ **Lemma 13 (Lemma 3.1 in [30]).** *Let $\epsilon, \epsilon' \leq 1/2$. Suppose that $\mathcal{M}_i$ is a $(1+\epsilon)$-MCM for G_i. For $j = i, i+1, \ldots, i+\lfloor \epsilon' \cdot |\mathcal{M}_i| \rfloor$, let $\mathcal{M}_i^{(j)}$ denote the matching $\mathcal{M}_i$ after removing from*

it all edges that got deleted during the updates $i+1,\ldots,j$. Then $\mathcal{M}_i^{(j)}$ is a $(1+2\epsilon+2\epsilon')$-MCM for the graph G_j.

Hence, we can compute a $(1+\epsilon/4)$-maximum matching $\mathcal{M}_i$ at a certain update i, and use the same matching $\mathcal{M}_i^{(j)}$ throughout all updates $j = i, i+1, \ldots, i' = i + \lfloor \epsilon/4 \cdot |\mathcal{M}_i| \rfloor$. (By Lemma 13, $\mathcal{M}_i^{(j)}$ is a $(1+\epsilon)$-maximum matching for all graphs G_j.) Next compute a fresh $(1+\epsilon/4)$-maximum matching $\mathcal{M}_{i'}$ following update i' and use it throughout all updates $i', i'+1, \ldots, i' + \lfloor \epsilon/4 \cdot |\mathcal{M}_{i'}| \rfloor$, and repeat.

In this way the static time complexity of computing a $(1+\epsilon)$-maximum matching $\mathcal{M}$ is amortized over $1 + \lfloor \epsilon/4 \cdot |\mathcal{M}| \rfloor = \Omega(\epsilon \cdot |\mathcal{M}|)$ updates. The key insight behind the schemes of [30, 48] is not to compute the approximate matching on the entire graph, but rather on a matching sparsifier, which is derived from an $O(1)$-VC that is maintained dynamically *by other means*. [48] showed that an ϵ-maximal matching, and thus a $(2+\epsilon)$-VC, denoted by $\widetilde{VC}$, can be maintained with $O(\alpha/\epsilon)$ update time, and the argument used by [48] was quite tricky; we will get back to this point soon. Specifically, the sparsifier $\tilde{G}$ of [48] contains (1) all edges in the subgraph $G[\widetilde{VC}]$ induced by $\widetilde{VC}$, and (2) for each vertex $v \in \widetilde{VC}$, (up to) $O(\alpha/\epsilon)$ edges connecting it with arbitrary neighbors outside $\widetilde{VC}$. Although the resulting sparsifier may have large degree, it is easy to verify that it contains $O(|\mathcal{M}| \cdot \alpha/\epsilon)$ edges, and it can be computed in time linear in its size. Since a $(1+\epsilon)$-maximum matching can be computed for the sparsifier $\tilde{G}$ in time $O(|\tilde{G}|/\epsilon) = O(|\mathcal{M}| \cdot \alpha/\epsilon^2)$ [36, 42, 54], the resulting amortized update time is $O(\alpha \cdot \epsilon^{-3})$. Also, one can easily translate the amortized bound into a worst-case bound, as shown in [30].

The aforementioned stability property also applies to the minimum VC and maximum IS problems. We next consider the minimum VC problem; see Section 4.2.1 for a discussion on the maximum IS problem. Thus the size of the minimum VC changes by at most 1 following each update. In extending the lazy scheme [30, 48] to the minimum VC problem, the challenge is to efficiently compute a high quality VC sparsifier. In particular, the $(1+\epsilon)$-matching sparsifier $\tilde{G}$ of [48] cannot be used as such a VC sparsifier, since a VC for it (regardless of its approximation guarantee) may not provide a *valid* VC for the graph G.

Fix an arbitrary parameter $t \geq 1$. Consider an arbitrary (sub)family of n-vertex graphs $\mathcal{G}$ with arboricity bounded by α that is closed under edge removals (such as the family of planar graphs), and suppose that we can compute a t-VC from scratch in time $T(n)$ for any graph in this family. (More generally, we assume that for any j-vertex subgraph H of any $G \in \mathcal{G}$, for any $1 \leq j \leq n$, we can compute in time $T(j)$ a t-VC for H.) Next, we adapt the lazy scheme [30, 48] to maintain a $(t+2\epsilon)$-VC in dynamically changing graphs of $\mathcal{G}$.

Lemma 13 easily extends to the minimum VC problem and to any approximation $t \geq 1$, i.e., any $(t+\epsilon)$-VC continues to provide a $(t+2\epsilon)$-VC throughout a long update sequence. Once the approximation guarantee of that VC, denoted by VC_{old}, becomes too poor (i.e., reaches $t+2\epsilon$), we shall *amplify* it (i.e., reduce it back to $t+\epsilon$) by computing a $(t+\epsilon)$-VC within time close to linear in $|VC_{low}|$. Having done that, we can then re-use the new amplified VC, denoted by VC_{new}, throughout the subsequent $\epsilon \cdot |VC_{new}|$ update steps, and repeat.

The computation of the new amplified cover VC_{new} employs the old cover VC_{old} and our VC sparsifier from Section 3.2 as follows. Recall that the vertex set V_{high} is the validating set of our VC sparsifier. It is straightforward to maintain all vertices in V_{high} dynamically with $O(\alpha/\epsilon)$ worst-case update time; we initialize VC_{new} as V_{high}. We next need to compute the subgraph G_{low} induced on the vertices V_{low} of degree $< \Delta = O(\alpha/\epsilon)$. This can be done by simply adding, for each vertex of VC_{old} of degree $< \Delta$, all its adjacent edges to vertices of degree $< \Delta$. Although some vertices of V_{low} may not belong to VC_{old}, we must have added all edges of G_{low}, as VC_{old} is a VC for G. Clearly, the number $|V_{low}|$ of vertices in G_{low}, as well

as the time needed to compute it, are bounded by $O(|VC_{old}| \cdot \alpha/\epsilon)$. We proceed by computing a t-VC for G_{low}, denoted by VC^t_{low}; the runtime of this static computation is $T(|V_{low}|)$. Finally, we add all vertices of VC^t_{low} to VC_{new}, thus we have $VC_{new} = VC^t_{low} \cup V_{high}$. By Theorem 9, VC_{new} is a $(t+\epsilon)$-VC for the entire graph G.

Observe that the overall runtime of computing VC_{new} is bounded by $O(|VC_{old}| \cdot \alpha/\epsilon) + T(|V_{low}|) \leq T(O(|VC_{old}| \cdot \alpha/\epsilon))$. Since this runtime is amortized over $\Theta(\epsilon \cdot |VC_{old}|)$ update steps, the amortized update time is bounded by $T(O(|VC_{old}| \cdot \alpha/\epsilon))/\Theta(\epsilon \cdot |VC_{old}|)$, which is no greater than $\frac{T(n)}{O((n/\alpha)\cdot\epsilon^2)}$. Note that the amortized number of changes to the VC is also bounded by $\frac{T(n)}{O((n/\alpha)\cdot\epsilon^2)}$. Moreover, one can easily translate the amortized update time into the same (up to a constant factor) worst-case update time, as shown in [30].

Summarizing, we have proved the following result.

▶ **Theorem 14.** *Fix arbitrary $t \geq 1, \epsilon > 0$. If a t-VC can be computed in time $T(n)$ in an arbitrary (sub)family of n-vertex graphs with arboricity bounded by α that is closed under edge removals, then a $(t+\epsilon)$-VC can be maintained with a worst-case update time of $\frac{T(n)}{O((n/\alpha)\cdot\epsilon^2)}$. Moreover, the amortized number of changes to the VC is also bounded by $\frac{T(n)}{O((n/\alpha)\cdot\epsilon^2)}$.*

Remarks.

1. For planar graphs, one can compute a $(1+\epsilon)$-VC in time $O(n)$, for any constant $\epsilon > 0$ [7]. By Theorem 14, we can maintain a $(1+\epsilon)$-VC with a constant worst-case update time for any constant $\epsilon > 0$.
2. For the family of graphs with arboricity bounded by α, one can compute a $(2 - \frac{2}{\beta+1})$-VC in time $O(n^{3/2} \cdot \alpha)$, where β is the average degree in some (carefully computed) subgraph, and is thus bounded by 2α [35]. By Theorem 14, we can maintain a VC with an approximation of roughly $2 - \frac{1}{\alpha}$ and a worst-case update time of $O(\sqrt{n} \cdot \alpha^2)$.

Our matching sparsifier from Section 3.1 can be used to simplify the algorithms of [48] and their analysis. First, as mentioned, [48] used a tricky argument to maintain a $(2+\epsilon)$-VC with $O(\alpha/\epsilon)$ update time. An alternative simpler approach is to maintain a maximal matching on top of our bounded degree matching sparsifier, which can be done naively with update time linear in the degree of the sparsifier, namely, $O(\alpha/\epsilon)$. By Theorem 7, this yields an ϵ-maximal matching, which is translated into an $(2+\epsilon)$-VC in the obvious way. Second, as explained in [48], the lazy scheme of [30] is inherently non-local. Since local algorithms are advantageous, [48] also devised a local algorithm for maintaining a $(1+\epsilon)$-maximum matching, which is quite intricate. An alternative simpler approach is to maintain a $(1+\epsilon)$-maximum matching on top of our bounded degree matching sparsifier, by dynamically excluding augmenting paths of length $O(1/\epsilon)$. By Theorem 3, this yields an $(1+O(\epsilon))$-maximum matching, and the update time of the resulting algorithm is $(\alpha/\epsilon)^{O(1/\epsilon)}$, just as in [48].

4.2.1 The dynamic approximate maximum IS problem

The size of the maximum IS changes by at most 1 following each update, hence we can apply the lazy scheme of [30, 48] also for this problem. Notice, however, that there is no need to compute a sparsifier here, since the maximum IS is of size at least $n/(\beta+1)$ by Turan's theorem, where β is the average degree in the graph. Hence, one can simply compute an approximate maximum IS on the entire graph, and amortize the cost of this static computation over $\Theta(\epsilon \cdot (n/\beta))$ update steps. Consequently, using the lazy scheme, we get a simple reduction from the dynamic to the static case: The update time (both amortized

and worst-case) for maintaining a $(t+\epsilon)$-IS is smaller than the static time complexity of computing a t-IS by a factor of $\Omega((n/\beta)\cdot\epsilon)$, for any $t \geq 1$ and $\epsilon > 0$.

Remarks.

1. For planar graphs, one can compute a $(1+\epsilon)$-IS in time $O(n)$, for any constant $\epsilon > 0$ [7]. We can thus maintain a $(1+\epsilon)$-IS with a constant worst-case update time for any constant $\epsilon > 0$.
2. For graphs with average degree bounded by β, one can compute a $(k+1)/2$-IS in time $O(n^{3/2}\cdot\beta)$ [35]. We can thus maintain an IS with an approximation of roughly $(k+1)/2$ and a worst-case update time of $O(\sqrt{n}\cdot\beta^2)$.

4.3 Distributed networks

We consider the standard $\mathcal{LOCAL}$ and $\mathcal{CONGEST}$ models of communication (cf. [47]), which are standard distributed computing models capturing the essence of spatial locality and congestion. All processors wake up simultaneously, and computation proceeds in fault-free synchronous rounds during which every processor exchanges messages of either unbounded size (in the $\mathcal{LOCAL}$ model) or of $O(\log n)$-bit size (in the $\mathcal{CONGEST}$ model). It is easy to see that our sparsification algorithms can be implemented in distributed networks using a single communication round, even using $O(1)$-bit messages, during which each processor sends messages along at most Δ of its adjacent edges, where Δ is the degree bound of the sparsifier. Hence, the results mentioned in Section 1.4 hold w.r.t. both the $\mathcal{LOCAL}$ and the $\mathcal{CONGEST}$ models of communication. In this way we provide a clean and simple reduction from either bounded arboricity graphs (for the distributed approximate maximum matching and minimum VC problems) or from bounded average degree graphs (for the distributed approximate maximum IS problem) to bounded degree graphs.

For the distributed approximate VC problem, [8] showed that a $(2+\epsilon)$-VC can be computed in $O(\log\Delta/(\epsilon\log\log\Delta))$ rounds, where Δ is the maximum degree in the graph. We can plug our reduction to extend the result of [8] to graphs of arboricity bounded by α.

▶ **Theorem 15.** *For any graph of arboricity bounded by α and any $\epsilon > 0$, there is a distributed algorithm for computing a $(2+\epsilon)$-VC in $O(\log(\alpha/\epsilon)/(\epsilon\log\log(\alpha/\epsilon)))$ rounds.*

As mentioned in Section 1.4, for the distributed approximate maximum matching problem, a reduction from bounded arboricity graphs to bounded degree graphs was already given in [22]. The reduction of [22] starts by computing carefully chosen vertex sets in the graph, using which a bounded degree subgraph is computed, in $O(1)$ communication rounds. A distributed approximate matching algorithm is then run on that subgraph. Based on the matching returned as output to that algorithm, another carefully chosen bounded degree subgraph is computed, in $O(1)$ more communication rounds. A distributed approximate matching algorithm is then run on the new subgraph, and the final matching is the union of the first matching and the second. Our reduction is obtained as an immediate corollary of our matching sparsifier from Section 3.1, and it has several advantages over the one of [22]. First and foremost, our reduction is much simpler. Second, it requires a single communication round, whereas the number of rounds in the reduction of [22] depends on the time required to compute an approximate matching. In particular, throughout the computation of our sparsifier, the *load* (as well as node congestion) on all processors is low, since any vertex sends messages only along Δ of its adjacent edges is a single round, where Δ is the degree bound of the sparsifier. Moreover, the load on processors will remain low by definition throughout the subsequent run of any distributed approximate matching

algorithm, since that algorithm is run on a bounded degree subgraph. In contrast, the computation of the subgraphs in [22] involves the exchange of messages between high degree vertices, and this "high-load" message exchange proceeds throughout $O(1)$ rounds; then a distributed approximate matching algorithm is run on a bounded degree subgraph, which may require many rounds to complete, and later another "high-load" message exchange proceeds throughout $O(1)$ more rounds. Third, the degree of our matching sparsifier is at most $O(\alpha/\epsilon)$, whereas the degree bound of the subgraphs of [22] is $O(\alpha/\epsilon^2)$, i.e., there is a gap of factor $1/\epsilon$, and this gap may be amplified significantly in applications where the runtime of the distributed algorithm depends exponentially on the maximum degree. In particular, [27] devised a distributed algorithm for computing a $(1+\epsilon)$-maximum matching in $\Delta^{O(1/\epsilon)} + O(\epsilon^{-2}) \cdot \log^* n$ rounds, for graphs with degree bounded by Δ. Due to our matching sparsifier, we extend the result of [27] to graphs of arboricity bounded by α to get a $(1+\epsilon)$-maximum matching in $(\alpha/\epsilon)^{O(1/\epsilon)} + O(\epsilon^{-2}) \cdot \log^* n$ rounds, which has a better dependence on ϵ than if the reduction of [22] were to be used.

References

1. Ittai Abraham, David Durfee, Ioannis Koutis, Sebastian Krinninger, and Richard Peng. On fully dynamic graph sparsifiers. In *Proc. 57th FOCS*, pages 335–344, 2016.
2. Noga Alon, Ronitt Rubinfeld, Shai Vardi, and Ning Xie. Space-efficient local computation algorithms. In *Proc. 23rd SODA*, pages 1132–1139, 2012.
3. S. Arya, G. Das, D. M. Mount, J. S. Salowe, and M. H. M. Smid. Euclidean spanners: short, thin, and lanky. In *Proc. of 27th STOC*, pages 489–498, 1995.
4. Sepehr Assadi, MohammadHossein Bateni, Aaron Bernstein, Vahab S. Mirrokni, and Cliff Stein. Coresets meet EDCS: algorithms for matching and vertex cover on massive graphs. *CoRR*, abs/1711.03076, 2017.
5. B. Awerbuch, A. Baratz, and D. Peleg. Cost-sensitive analysis of communication protocols. In *Proc. of 9th PODC*, pages 177–187, 1990.
6. B. Awerbuch, A. Baratz, and D. Peleg. Efficient broadcast and light-weight spanners. *Technical Report CS92-22, Weizmann Institute*, October, 1992.
7. Brenda S. Baker. Approximation algorithms for np-complete problems on planar graphs. *J. ACM*, 41(1):153–180, 1994.
8. Reuven Bar-Yehuda, Keren Censor-Hillel, and Gregory Schwartzman. A distributed $(2+\epsilon)$-approximation for vertex cover in o($\log\delta/\epsilon$ log log δ) rounds. In *Proc. PODC*, pages 3–8, 2016.
9. Leonid Barenboim, Michael Elkin, Seth Pettie, and Johannes Schneider. The locality of distributed symmetry breaking. *J. ACM*, 63(3):20:1–20:45, 2016.
10. S. Baswana, M. Gupta, and S. Sen. Fully dynamic maximal matching in $O(\log n)$ update time. In *Proc. of 52nd FOCS*, pages 383–392, 2011.
11. András A. Benczúr and David R. Karger. Approximating s-t minimum cuts in $\tilde{O}(n^2)$ time. In *Proc. 28th STOC*, pages 47–55, 1996.
12. Aaron Bernstein and Cliff Stein. Fully dynamic matching in bipartite graphs. In *Proc. 42nd ICALP*, pages 167–179, 2015.
13. Aaron Bernstein and Cliff Stein. Faster fully dynamic matchings with small approximation ratios. In *Proc. of 26th SODA*, pages 692–711, 2016.
14. S. Bhattacharya, M. Henzinger, and G. F. Italiano. Deterministic fully dynamic data structures for vertex cover and matching. In *Proc. 26th SODA*, pages 785–804, 2015.
15. Sayan Bhattacharya, Monika Henzinger, and Danupon Nanongkai. New deterministic approximation algorithms for fully dynamic matching. In *Proc. 48th STOC*, pages 398–411, 2016.

16 Bartlomiej Bosek, Dariusz Leniowski, Piotr Sankowski, and Anna Zych. Online bipartite matching in offline time. In *Proc. 55th FOCS*, pages 384–393, 2014.

17 Keren Censor-Hillel, Elad Haramaty, and Zohar S. Karnin. Optimal dynamic distributed MIS. In *Proceedings of the 2016 ACM Symposium on Principles of Distributed Computing, PODC 2016, Chicago, IL, USA, July 25-28, 2016*, pages 217–226, 2016.

18 Shiri Chechik. Compact routing schemes with improved stretch. In *ACM Symposium on Principles of Distributed Computing, PODC '13, Montreal, QC, Canada, July 22-24, 2013*, pages 33–41, 2013.

19 Eden Chlamtác and Michael Dinitz. Lowest degree k-spanner: Approximation and hardness. In *Proc. of APPROX/RANDOM*, pages 80–95, 2014.

20 Eden Chlamtac, Michael Dinitz, and Robert Krauthgamer. Everywhere-sparse spanners via dense subgraphs. In *PRof. 53rd FOCS*, pages 758–767, 2012.

21 Artur Czumaj, Jakub Lacki, Aleksander Madry, Slobodan Mitrovic, Krzysztof Onak, and Piotr Sankowski. Round compression for parallel matching algorithms. *CoRR*, abs/1707.03478, 2017.

22 Andrzej Czygrinow, Michal Hanckowiak, and Edyta Szymanska. Fast distributed approximation algorithm for the maximum matching problem in bounded arboricity graphs. In *PRoc. 20th ISAAC*, pages 668–678, 2009.

23 G. Das, P. J. Heffernan, and G. Narasimhan. Optimally sparse spanners in 3-dimensional Euclidean space. In *Proc. of 9th SOCG*, pages 53–62, 1993.

24 G. Das and G. Narasimhan. A fast algorithm for constructing sparse Euclidean spanners. *Int. J. Comput. Geometry Appl.*, 7(4):297–315, 1997.

25 Michael Elkin and Shay Solomon. Optimal euclidean spanners: Really short, thin, and lanky. *J. ACM*, 62(5):35:1–35:45, 2015.

26 Guy Even, Moti Medina, and Dana Ron. Deterministic stateless centralized local algorithms for bounded degree graphs. In *Proc. 22th ESA*, pages 394–405, 2014.

27 Guy Even, Moti Medina, and Dana Ron. Distributed maximum matching in bounded degree graphs. In *Proc. ICDCN*, pages 18:1–18:10, 2015.

28 Manuela Fischer. Improved deterministic distributed matching via rounding. *CoRR*, abs/1703.00900v2, 2017.

29 J. Gudmundsson, C. Levcopoulos, and G. Narasimhan. Fast greedy algorithms for constructing sparse geometric spanners. *SIAM J. Comput.*, 31(5):1479–1500, 2002.

30 M. Gupta and R. Peng. Fully dynamic $(1 + \epsilon)$-approximate matchings. In *Proc. of 54th FOCS*, pages 548–557, 2013.

31 M. Gupta and A. Sharma. An $o(\log(n))$ fully dynamic algorithm for maximum matching in a tree. *CoRR*, abs/0901.2900, 2009.

32 Torben Hagerup, Jyrki Katajainen, Naomi Nishimura, and Prabhakar Ragde. Characterizing multiterminal flow networks and computing flows in networks of small treewidth. *J. Comput. Syst. Sci.*, 57(3):366–375, 1998.

33 P. Hall. On representatives of subsets. *J. London Math. Soc*, 10(1):26–30, 1935.

34 M. He, G. Tang, and N. Zeh. Orienting dynamic graphs, with applications to maximal matchings and adjacency queries. In *Proc. 25th ISAAC*, pages 128–140, 2014.

35 Dorit S. Hochbaum. Efficient bounds for the stable set, vertex cover and set packing problems. *Discrete Applied Mathematics*, 6(3):243–254, 1983.

36 J. E. Hopcroft and R. M. Karp. An $n^{5/2}$ algorithm for maximum matchings in bipartite graphs. *SIAM J. Comput.*, 2(4):225–231, 1973.

37 Subhash Khot and Oded Regev. Vertex cover might be hard to approximate to within 2-epsilon. *J. Comput. Syst. Sci.*, 74(3):335–349, 2008.

38 T. Kopelowitz, R. Krauthgamer, E. Porat, and S. Solomon. Orienting fully dynamic graphs with worst-case time bounds. In *Proc. 41st ICALP*, pages 532–543, 2014.

39 Frank Thomson Leighton and Ankur Moitra. Extensions and limits to vertex sparsification. In *Proceedings of the 42nd ACM Symposium on Theory of Computing, STOC 2010, Cambridge, Massachusetts, USA, 5-8 June 2010*, pages 47–56, 2010.

40 Reut Levi, Ronitt Rubinfeld, and Anak Yodpinyanee. Local computation algorithms for graphs of non-constant degrees. *Algorithmica*, 77(4):971–994, 2017.

41 Yishay Mansour and Shai Vardi. A local computation approximation scheme to maximum matching. In *Proc. APPROX/RANDOM*, pages 260–273, 2013.

42 S. Micali and V. V. Vazirani. An $O(\sqrt{|V||E|})$ algorithm for finding maximum matching in general graphs. In *Proc. of 21st FOCS*, pages 17–27, 1980.

43 Ankur Moitra. Approximation algorithms for multicommodity-type problems with guarantees independent of the graph size. In *50th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2009, October 25-27, 2009, Atlanta, Georgia, USA*, pages 3–12, 2009.

44 Ofer Neiman and Shay Solomon. Simple deterministic algorithms for fully dynamic maximal matching. In *Proc. of 45th STOC*, pages 745–754, 2013.

45 K. Onak and R. Rubinfeld. Maintaining a large matching and a small vertex cover. In *Proc. of 42nd STOC*, pages 457–464, 2010.

46 Merav Parter, David Peleg, and Shay Solomon. Local-on-average distributed tasks. In *Proc. of 27th SODA*, pages 220–239, 2016.

47 D. Peleg. *Distributed Computing: A Locality-Sensitive Approach*. SIAM, 2000.

48 D. Peleg and S. Solomon. Dynamic $(1+\epsilon)$-approximate matchings: A density-sensitive approach. In *Proc. of 27th SODA*, pages 712–729, 2016.

49 David Peleg and Jeffrey D. Ullman. An optimal synchronizer for the hypercube. *SIAM J. Comput.*, 18(4):740–747, 1989.

50 Ronitt Rubinfeld, Gil Tamir, Shai Vardi, and Ning Xie. Fast local computation algorithms. In *Prof. of ICS*, pages 223–238, 2011.

51 S. Solomon. Fully dynamic maximal matching in constant update time. In *Proc. 57th FOCS*, pages 325–334, 2016.

52 Daniel A. Spielman and Shang-Hua Teng. Nearly-linear time algorithms for graph partitioning, graph sparsification, and solving linear systems. In *Proc. 36th STOC*, pages 81–90, 2004.

53 Mikkel Thorup and Uri Zwick. Compact routing schemes. In *Proc. of 13th SPAA*, pages 1–10, 2001.

54 V. V. Vazirani. An improved definition of blossoms and a simpler proof of the MV matching algorithm. *CoRR*, abs/1210.4594, 2012.

Proofs of Proximity for Distribution Testing*†

Alessandro Chiesa[1] and Tom Gur[2]

1 **UC Berkeley, California, USA**
`alexch@berkeley.edu`
2 **UC Berkeley, California, USA**
`tom.gur@berkeley.edu`

Abstract

Distribution testing is an area of property testing that studies algorithms that receive few samples from a probability distribution $\mathcal{D}$ and decide whether $\mathcal{D}$ has a certain property or is far (in total variation distance) from all distributions with that property. Most natural properties of distributions, however, require a large number of samples to test, which motivates the question of whether there are natural settings wherein fewer samples suffice.

We initiate a study of proofs of proximity for properties of distributions. In their basic form, these proof systems consist of a tester (or verifier) that not only has sample access to a distribution but also explicit access to a proof string that depends arbitrarily on the distribution. We refer to these as NP distribution testers, or MA distribution testers if the tester is a probabilistic algorithm. We also study IP distribution testers, a more general notion where the tester interacts with an all-powerful untrusted prover.

We investigate the power and limitations of proofs of proximity for distributions and chart a landscape that, surprisingly, is significantly different from that of proofs of proximity for functions. Our main results include showing that MA distribution testers can be quadratically stronger than standard distribution testers, but no stronger than that; in contrast, IP distribution testers can be exponentially stronger than standard distribution testers, but when restricted to public coins they can be quadratically stronger at best.

1998 ACM Subject Classification F.1.2 [Modes of Computation]: Probabilistic computation

Keywords and phrases distribution testing, proofs of proximity, property testing

Digital Object Identifier 10.4230/LIPIcs.ITCS.2018.53

1 Introduction

Distribution testing, introduced by Goldreich and Ron [34] and Batu et al. [8], is an area of property testing [55, 30] that studies sublinear-time algorithms for approximate decision problems regarding probability distributions over massive domains. Such algorithms, known as *distribution testers*, are given independent samples from an unknown distribution and are required to decide whether the distribution has a certain property, or is far from having it. More precisely, a distribution tester for a property Π of distributions over a domain Ω is a probabilistic algorithm that, given a proximity parameter $\varepsilon > 0$, determines whether a distribution $\mathcal{D}$ over Ω has the property Π or is ε-far (typically, in total variation distance) from any distribution that has Π, by drawing a sublinear number of independent samples from $\mathcal{D}$.

* This work was supported in part by the UC Berkeley Center for Long-Term Cybersecurity.
† A full version of the paper is available at [23], `https://eccc.weizmann.ac.il/report/2017/155`

In the last two decades distribution testing has received much attention, not only because it asks fundamental questions about distributions but also because it has applications ranging from statistical hypothesis testing [43] and model selection [14] to property testing [34, 17] and biology [62, 48]. A long line of works, including [7, 46, 6, 47, 49, 58, 3, 12, 44, 22, 11, 24, 57], has investigated many natural properties of distributions, determining the sample complexity of core problems such as testing uniformity, support size, identity to a specified distribution, and many more (see recent surveys [53, 16] and a forthcoming book [29]).

Whereas testing properties of functions is often possible with few queries (independently of the function's domain size), testing properties of distributions typically requires many samples. In particular, the vast majority of properties of distributions studied in the literature require $\Omega(\sqrt{n})$ samples to test, where n is the domain size. This state of affairs has motivated researchers to study distribution testing using stronger types of access to the distribution [18, 26, 2, 21], in which the tester can draw samples conditioned on a subset of the domain, and models in which the tester is granted additional access to the cumulative distribution function or probability mass function of the distribution [54, 15]. In this work we take a different approach: we allow the tester to be aided by a prover, but keep the standard sample access to the distribution (without any conditioning), as we now explain.

A fundamental question that arises in any computational model is to understand the power of a 'proof'. Indeed, the famous $\mathbf{P} \neq \mathbf{NP}$ conjecture, which is concerned with the power of proofs in the setting of polynomial-time computation, is widely considered as one of the most important open problems in the theory of computation. Moreover, proof systems are studied in many other settings, such as communication complexity [4, 1, 42], quantum computation [61, 50, 59], data streams [19, 38, 20], and, most relevant to this work, property testing, as we now recall.

Proofs in the *functional* (standard) setting of property testing are known as *proofs of proximity* [25, 9]. These are probabilistic proof systems in which the verifier makes a sublinear number of queries to a statement, and is only required to reject statements that are far from true. In a Merlin–Arthur proof of proximity (MAP) [39], the verifier receives explicit access to a proof of sublinear length, in addition to query access to the statement. More generally, in an interactive proof of proximity (IPP) [52], the verifier interacts with an all-powerful untrusted prover. MAPs and IPPs have been studied in a line of recent works, including [27, 33, 41, 32, 28, 31, 51, 40, 10], and may be thought of as the MA (i.e., "randomized NP") and IP analogues of functional property testing, respectively.

In this work, we initiate a study of proof systems for testing properties of *distributions*, i.e., proofs of proximity for distribution testing. We define several natural types of proofs, and investigate their power and limitations. The landscape that we chart turns out to be completely different, both qualitatively and quantitatively, from that for proofs of proximity for *functions*. We now discuss our results, first on non-interactive proofs and then on interactive proofs.

2 Non-interactive proofs of proximity for distribution testing

We study a natural analogue of the notion of NP proofs for testing properties of distributions. Letting $\Delta(\Omega_n)$ be the set of distributions over a domain Ω, and letting $\Pi \subseteq \Delta(\Omega)$ be a property, the tester is given sample access to a distribution $\mathcal{D} \in \Delta(\Omega)$ and explicit access to a proof π and proximity parameter ε. We require that for every distribution $\mathcal{D} \in \Pi$ there exists a proof π such that the tester accepts, and for every distribution $\mathcal{D}$ that is ε-far from Π and every proof the tester rejects, both with high probability (e.g., 2/3).

Following standard conventions, if such a tester is a *deterministic* algorithm (i.e., is not allowed to toss coins), then we call it an NP distribution tester, and if it is a *probabilistic* algorithm, then we call it an MA distribution tester. As we discuss later, in stark contrast to proofs of proximity for *functions*, for which deterministic testers are degenerate [38], the power of MA distribution testers and NP distribution testers is essentially equivalent. Thus we henceforth present our results for MA distribution testers only, and remark that these results qualitatively translate to NP distribution testers as well.

Analogously to prior work in distribution testing and proximity proofs, we consider two main efficiency measures for MA distribution testers: (a) *sample complexity*, which is the number of samples drawn by the tester from the distribution; (b) *proof complexity*, which is the length of the honest proof. Both complexity measures are functions of the domain size and the proximity parameter.

Perhaps the first question that arises in this direction is whether verification can be cheaper than decision. In other words, are MA distribution testers stronger than standard distribution testers? For functional proofs of proximity the answer is immediate: every property can be tested with just $O(1)$ queries to the input, when given a linear-size proof. This proof simply contains a description of the input, in which case the tester can read the entire proof, decide membership in the property, and query the input at few random locations to check that it is close to the proof. Linear-size proofs thus trivialize testing properties of functions.

In distribution testing, however, the situation is not as simple. For starters, given a purported description of $\mathcal{D}$, checking that this description actually matches the input distribution typically requires more than a constant number of samples. Moreover, the description of a distribution $\mathcal{D}$ may be very large (even infinite), and so the proof cannot simply contain its description. Nonetheless, these difficulties can be dealt with, albeit at the cost of higher complexity.

To simplify exposition, throughout the introduction we fix a domain Ω_{n} of size n and fix the proximity parameter ε to a small constant. Our first result shows that proofs of (nearly) linear length allow testing *any* property with only $O(\sqrt{\mathsf{n}})$ samples; moreover, there are natural properties for which the sample complexity can be smoothly reduced (down to constant) using increasingly longer proofs.

▶ **Theorem 1** (informal; see full version for details)**.**

1. *For any property* $\Pi \subseteq \Delta(\Omega_{\mathsf{n}})$*, there exists an MA distribution tester with proof complexity* $O(\mathsf{n}\log(\mathsf{n}))$ *and sample complexity* $\mathsf{s} = O(\max_{\mathcal{D}\in\Pi} \|D\|_{2/3}) = O(\sqrt{\mathsf{n}})$*. (Here* $\|\cdot\|_{2/3}$ *is the* $\ell_{2/3}$ *quasi-norm.)*
2. *There exists a (natural) property* $\Pi \subseteq \Delta(\Omega_{\mathsf{n}})$ *for which every distribution tester uses* $\tilde{\Omega}(\mathsf{n})$ *samples, yet there is an MA distribution tester for* Π *with proof complexity* $O(\mathsf{n}\log(\mathsf{n}))$ *and sample complexity* $O(1)$*. Furthermore, one can trade proof against sample complexity and, e.g., make both complexities* $\tilde{O}(\sqrt{\mathsf{n}})$*.*

We remark that the second item of Theorem 1 is proved with respect to a *promise* problem.

Theorem 1 confirms the intuition that MA distribution testers are stronger than standard distribution testers. However, while in the settings of proximity proofs for functions it is possible to obtain exponential savings in query complexity, even using proofs of merely logarithmic length [39], our Theorem 1 only shows MA distribution testers in which the product of the proof and sample complexities is at least as large as the sample complexity

of standard distribution testers.[1] This discussion raises the question of whether there exist stronger MA distribution testers, or whether non-interactive proofs of proximity for distributions are indeed more limited than their functional counterparts.

Furthermore, Theorem 1 shows that the sample complexity of MA distribution testers for any property can be reduced to $O(\sqrt{n})$. Yet, for properties that can be tested (without a proof) using $O(\sqrt{n})$ samples, is it always the case that MA distribution testers can be stronger than standard distribution testers?

To answer the questions above, we study the *limitations* of non-interactive proofs of proximity for distributions. Our next result shows that for *every* property and every MA distribution tester, either its proof or its sample complexity can at best be quadratically better than the (optimal) sample complexity of a standard distribution tester. Moreover, there also exists a natural property (the property of being uniformly distributed) for which MA distribution testers cannot do better than standard distribution testers.

▶ **Theorem 2** (informal; see full version for details)**.** *Let* s_Π *be the optimal sample complexity for testing a property* Π *without the aid of any proofs.*

- *For every* $\Pi \subseteq \Delta(\Omega_{\mathsf{n}})$ *and every MA distribution tester for* Π *with proof complexity* p *and sample complexity* s*, it holds that* $\mathsf{p} \cdot \mathsf{s} = \Omega(\mathsf{s}_\Pi)$*.*
- *Every MA distribution tester for the uniformity property* U_{n} *has sample complexity* $\Omega(\mathsf{s}_{\mathsf{U}_{\mathsf{n}}}) = \Omega(\sqrt{\mathsf{n}})$*, regardless of its proof complexity.*

Theorem 2 thus shows that the upper bounds in Theorem 1 are tight, up to logarithmic factors. (The first item of Theorem 2 shows the tightness of the second item of Theorem 1, and the second item of Theorem 2 shows the tightness of the first item of Theorem 1 with respect to a particular property.)

On derandomizing MA distribution testers.

As mentioned above, the power of deterministic verification (NP proofs) and randomized verification (MA proofs) is essentially equivalent in the setting of distribution testing. More accurately, the following theorem shows that MA distribution testers can be derandomized into NP distribution testers at the price of only a small increase in sample complexity.

▶ **Theorem 3** (informal; see full version for details)**.** *Every MA distribution tester with proof complexity* p *and sample complexity* s *can be emulated by an NP distribution tester with proof complexity* p *and sample complexity* $O\big(\mathsf{s} + \log(\mathsf{n})\big)$*.*

We remark that a direct proof for the special case of standard testers (without access to a proof) is sketched in [29, Chapter 11].

3 Interactive proofs of proximity for distribution testing

While MA distribution testers are stronger than standard distribution testers, they are limited to multiplicatively trading off sample complexity for proof complexity. Can one do even better with other types of proof systems? To study this question, we consider a natural analogue of *interactive proofs* [36] in the setting of distribution testing.

[1] To see this holds with respect to the first item of Theorem 1, recall that every property can be tested using $O(\mathsf{n})$ samples (for a constant value of the proximity parameter).

An *IP distribution tester* generalizes the notion of an MA distribution tester by allowing the tester to interact with an all-powerful untrusted prover who knows everything about the input distribution $\mathcal{D}$. The prover tries to convince the tester that $\mathcal{D}$ has a certain property Π. If $\mathcal{D} \in \Pi$ then there exists a prover strategy that makes the tester accept with high probability; if instead $\mathcal{D}$ is far from Π then the tester rejects with high probability regardless of prover strategy.

Similarly to the non-interactive setting, we seek to minimize the sample complexity, as well as *communication complexity*, which is the total number of bits exchanged between the two parties (and generalizes proof complexity). We also consider the *round complexity*, which is the number of rounds of interaction, where each round consists of a message from one party to the other and its reply.

The next theorem shows that it is possible to test properties of distributions much more efficiently by interacting with a prover than by receiving a non-interactive proof. In fact, even a single round of interaction suffices to obtain *exponential* savings in communication and sample complexity compared to the sample complexity of standard distribution testers (and hence MA distribution testers as well).

▶ **Theorem 4** (informal, see full version). *There exists a property $\Pi \subseteq \Delta(\Omega_{\mathsf{n}})$ such that:*

1. *there is a 1-round IP distribution tester for Π with communication complexity $O(\log(\mathsf{n}))$ and sample complexity $O(1)$; yet*
2. *every (standard) distribution tester for Π must use $\tilde{\Omega}(\sqrt{\mathsf{n}})$ samples.*

A fundamental distinction between types of interactive proofs is according to how the tester uses its own randomness. The interaction is public-coin if the tester reveals the outcome of its coins immediately after tossing them; it is private-coin if the tester can keep such outcomes to itself. Public-coin interactive proofs are called AM proofs [5], and so we call their distribution testing analogues AM distribution testers. We stress that in these public-coin protocols, the prover does *not* see the samples drawn by the tester.

Goldwasser and Sipser [37] proved that the expressive power of private-coin interactive proofs is essentially equivalent to that of public-coin interactive proofs, despite the latter being syntactically weaker. Rothblum, Vadhan, and Wigderson [52] observed that [37]'s proof of this statement extends to the setting of interactive proofs of proximity for *functions*. The next theorem shows that, unlike in the aforementioned models, the power of public-coin interaction for testing distributions is rather limited, *regardless of round complexity*.

▶ **Theorem 5** (informal, see full version). *For every property $\Pi \subseteq \Delta(\Omega_{\mathsf{n}})$ and $\mathsf{r} \in \mathbb{N}$ (not necessarily a constant), it holds that every r-round AM distribution tester for Π with communication complexity c and sample complexity s satisfies $\mathsf{c} \cdot \mathsf{s} = \Omega(\mathsf{s}_\Pi)$. (As before, s_Π denotes the optimal sample complexity for testing property Π without the aid of any proofs.)*

We note that the combination of our Theorems 4 and 5 yields an *exponential* separation between the power of IP distribution testers and AM distribution testers, which stands in stark contrast to the equivalence of private-coin and public-coin interaction in the functional setting.

While their power is limited when compared to IP distribution testers, AM distribution testers are still stronger than standard distribution testers, and possibly MA distribution testers as well. In the full version we show an AM distribution tester for a natural property that tightly matches the lower bound in Theorem 5, and also allows for smooth communication versus sample complexity tradeoffs. It is an open problem whether this upper bound can also be obtained via MA distribution testers, or whether public coin interaction in the setting of distribution testing is strictly stronger.

Table 1 Comparison between proofs of proximity for testing distributions and testing functions.

		Testing Distributions this work	**Testing Functions** [52, 39, 27, 40]
non-interactive proofs	**Proofs of linear length**	reduce sample complexity of *any* property to $O(\sqrt{n})$	reduce sample complexity of *any* property to $O(1)$
	MA proofs of proximity vs. standard testers	quadratically stronger	exponentially stronger
	Probabilistic (MA) vs. deterministic (NP) verification	nearly equivalent	NP proofs of proximity are extremely weak
	Hardest property for non-interactive proofs	explicit and natural; no better than standard testers, regardless of proof length	non-explicit (random property); linear length proof is required to outperform standard testers
interactive proofs	**Private vs. public coin protocols**	exponential separation	almost equivalent
	AM round hierarchy coin protocols	AM complexity is quadratically related to the sample complexity of standard testers	there is a property for which the AM complexity is $\approx n^{1/r}$ for r-round protocols

4 Comparison of functional and distributional proofs of proximity

In this work we consider several fundamental questions about proofs of proximity that were previously studied for properties *of functions*. We study these questions for properties *of distributions* instead.

One may naively expect that, since we are asking similar questions, we should obtain similar answers. However our results demonstrate that proofs of proximity for distributions behave dramatically different, both qualitatively and quantitatively, from proofs of proximity for functions. We summarize these different "complexity landscapes" in Table 1.

In retrospect these dramatic differences are easily interpreted. First and foremost, even standard (function) property testing and distribution testing are dissimilar: not only the tested objects are structurally different, but, just as importantly, the *access* to these objects is different as well (query access versus sample access). Moreover, these differences are more pronounced with regard to proofs of proximity because proof techniques to reason about them are very sensitive to input representation and access type. This is indeed what we find when inspecting our proof techniques, and the reasons for why our results hold.

5 Techniques

We establish our results via an eclectic set of technical tools that varies from section to section. These include extraction and derandomization, reductions from SMP communication complexity, lifting lemmas, granular approximation, and tolerant testing. To facilitate understanding of the main ideas behind each result, in the technical sections we precede the formal proof of each result with an intuitive high-level overview.

Below, we provide a taste of our techniques, grouped according to whether they give us upper bounds (Section 5.1), lower bounds (Section 5.2), or derandomization (Section 5.3).

5.1 Upper bounds

We overview the techniques that we use to obtain: a generic upper bound for MA distribution testers (first item of Theorem 1), an improved MA upper bound for a particular property (second item of Theorem 1), and an IP distribution tester that is exponentially more efficient than any MA distribution tester (first item of Theorem 4).

A generic MA upper bound.

We sketch a proof of a special case of Theorem 1, showing that *any* property can be tested via an MA distribution tester that uses $O(\sqrt{n}/\varepsilon^2)$ samples and a proof of linear size. The idea is that a linear-size proof π can allegedly consist of a description of the input distribution $\mathcal{D} \in \Pi$. Since the tester has explicit access to π and our goal is to minimize *sample* complexity (and not *time* complexity), the MA distribution tester can directly check membership of π in the property Π, reducing the problem to testing that the input distribution $\mathcal{D}$ is identical to π, a task that can be performed via $O(\sqrt{\mathsf{n}}/\varepsilon^2)$ samples [56].

One problem that arises is that, unlike the setting of testing Boolean functions or graphs, in the setting of distribution testing the size of the description of $\mathcal{D}$ may be very large (even infinite). To overcome this, we let an honest proof consist of a *granular* approximation $\mathcal{D}'$ of $\mathcal{D}$, where the mass of each element in the support of $\mathcal{D}'$ is a multiple of $m := \Theta(1/\mathsf{n})$; this approximation has at most linear size.

Note, however, that it could be the case that $\mathcal{D} \in \Pi$, whereas its granular approximation $\mathcal{D}'$ is close to Π but not in Π (similarly, $\mathcal{D}$ may by ε-far from Π, whereas may $\mathcal{D}'$ not). Nevertheless, using a *tolerant* testing procedure, the tester can ensure that with high probability it would rule regarding $\mathcal{D}'$ just as it would regarding $\mathcal{D}$, and so the granular approximation suffices to this end.

MA distribution tester with sublinear proofs.

To simplify the following presentation, we restrict our attention to m-granular distributions over the domain $[\mathsf{n}]$, for some $m = \Omega(1/\mathsf{n})$.

Consider the *gap isolated elements* problem, which is the problem of deciding whether a distribution $\mathcal{D}$ has a large number of isolated elements, or only a small one, where an element $i \in [\mathsf{n}]$ is said to be isolated if $\mathcal{D}$ is not supported on its adjacent elements $i-1$ and $i+1$.

We sketch an MA distribution tester with proof and sample complexity $\tilde{O}(\sqrt{\mathsf{n}})$ that accepts distributions with at least $\sqrt{\mathsf{n}}$ isolated elements and rejects distributions with at most $\sqrt{\mathsf{n}}/2$. (In the full version we show proof versus sample complexity tradeoffs for a wide range of parameterizations of this problem.)

The proof string simply specifies $\sqrt{\mathsf{n}}$ allegedly isolated elements of the input distribution $\mathcal{D}$, and the MA distribution tester draws $O(\sqrt{\mathsf{n}})$ samples and accepts if and only if all of the samples are not adjacent to the elements specified by the prover. Of course, if $\mathcal{D}$ indeed has at least $\sqrt{\mathsf{n}}$ isolated elements, the proof can specify them, and the MA distribution tester will accept with probability 1.

The key point is that if $\mathcal{D}$ has at most $\sqrt{\mathsf{n}}/2$ isolated elements, then every purported proof must specify at least $\sqrt{\mathsf{n}}/2$ elements that have an adjacent element on which $\mathcal{D}$ is supported on. Denote these supported adjacent elements by B, and note that every element of B is in fact a local certificate that $\mathcal{D}$ is a no-instance; that is, if the tester draws a *single* element in B, it can safely reject. By the granularity of $\mathcal{D}$ the total mass of B is $\Omega(1/\sqrt{\mathsf{n}})$, and so it suffices to draw $O(\sqrt{\mathsf{n}})$ samples to hit B with high probability.

IP distribution tester with logarithmic complexity.

We sketch an IP distribution tester for the isolated elements problem that has logarithmic communication complexity and constant sample complexity. (In the full version we also show that any public-coin IP distribution tester, and in particular standard and MA distribution testers, has exponentially larger complexity.)

Here we use different parameter settings than above, and in fact we shall not need the gap (promise problem) variant, and simply consider the property

$$\Pi_{\mathsf{Isolated}} := \{\mathcal{D} \in \Delta([\mathsf{n}]) \mid \forall i \in [\mathsf{n}]\ i \notin \mathrm{supp}(D) \text{ or } (i+1) \notin \mathrm{supp}(D)\}\ ;$$

that is, all distributions (not necessarily granular) in which no two consecutive elements are supported.

Consider the following IP distribution tester for this property. The tester draws $O(1/\varepsilon)$ samples from the input distribution $\mathcal{D}$ and *masks* these samples by shifting each sample to its subsequent element with probability 1/2. The tester then sends the masked samples to the prover and asks the prover to recover the original samples (prior to the shifts).

The point is that if the supported elements of $\mathcal{D}$ are indeed isolated, then the prover can always determine the original samples (as $\mathcal{D}$ cannot be supported on both an element and its shift). On the other hand, if $\mathcal{D}$ is ε-far from Π_{Isolated}, then there exist adjacent supported elements whose weight is $\Omega(\varepsilon)$, an so the prover is forced to guess which samples were shifted and which not, and will get caught with constant probability.

5.2 Lower bounds

Our lower bounds are all based on the following paradigm: we first prove a lower bound on the complexity of BPP distribution testers, typically via a reduction from SMP communication complexity, and then use "lifting" lemmas that allow us to transfer this lower bound to MA and AM distribution testers (where recall that by the latter we refer to public-coin *interactive* proof systems). We illustrate this methodology by sketching a proof of lower bounds on the complexity of MA and AM distribution testers for the isolated elements property Π_{Isolated}, which consists of all distributions in which no two consecutive elements are supported.

BBP lower bound via reduction from communication complexity.

We use the SMP communication complexity method [13]. Recall that, in a private-coin SMP protocol for a predicate f, the players Alice and Bob are given strings $x, y \in \{0,1\}^k$ (respectively), and each of the players is allowed to send a message, which depends on the player's input and *private* randomness, to a referee who is then required to decide whether $f(x,y) = 1$ by only looking at the players' messages and flipping coins. It is well-known that for the equality predicate ($f(x,y) = 1 \leftrightarrow x = y$), every such protocol must communicate $\Omega(\sqrt{k})$ bits [45].

Let P contain each third element of the domain, i.e., $P := \{3j - 1 \mid j \in [(\mathsf{n}-1)/3]\}$. Our reduction will map (a) yes-instances of EQ_k to distributions that are uniform over $|P|$ isolated elements; and (b) no-instances of EQ_k to distributions wherein for an ε-fraction of $p \in P$ it holds that $\mathcal{D}(p) = \Omega(1/\mathsf{n})$ and $\mathcal{D}(p+1) = \Omega(1/\mathsf{n})$, hence D is ε-far from Π_{Isolated}. Details follow.

Assume there exists a tester for Π_{Isolated} with sample complexity s. Each of the players encodes its input string via a balanced asymptotically good code ECC (that is, ECC: $\{0,1\}^k \to \{0,1\}^n$ with constant rate and relative distance $\varepsilon = \Omega(1)$, such that each codeword of

ECC contains the same number of 0's and 1's). Alice and Bob each draw $O(s)$ samples that are uniformly distributed over P, and *shift* each sample according to $\mathrm{ECC}(x)$ and $\mathrm{ECC}(y)$, respectively. That is, Alice sends to the referee independent samples uniformly drawn from $A := \left\{ i + \mathrm{ECC}(x)_{(i+1)/3} \mid i \in P \right\}$, and Bob sends samples uniformly drawn from $B := \left\{ i + \mathrm{ECC}(y)_{(i+1)/3} \mid i \in P \right\}$. Finally, the referee invokes the tester for Π_{Isolated} with respect to the distribution $\frac{1}{2}\mathsf{U}_{\mathsf{n}}(A) + \frac{1}{2}\mathsf{U}_{\mathsf{n}}(B)$, emulating each draw by tossing a random coin and deciding accordingly whether to use a sample by Alice or Bob.

The point is that if $x = y$, then $\mathrm{ECC}(x) = \mathrm{ECC}(y)$, and so both players shift their samples (which are in P, and so separated by two non-supported elements) in the same way, and so the resulting mixed distribution is uniform over isolated elements. On the other hand, if $x \neq y$, then $\mathrm{ECC}(x)$ is ε-far from $\mathrm{ECC}(y)$, and so the resulting distribution will have roughly $\varepsilon \cdot |P|$ non-isolated elements of weight $\Omega(1/|P|)$ each. Thus, we have $s = \tilde{\Omega}(\sqrt{k}) = \tilde{\Omega}(\sqrt{n})$.

Lifting the BPP lower bound to MA and r-round AM distribution testers.

We begin with the simpler task of proving an MA lower bound on Π_{Isolated}. To lift the BPP lower bound we proved above to MA, we show that any MA distribution tester T for any property Π (in particular, Π_{Isolated}) with proof complexity p and sample complexity s can be emulated by a BPP distribution tester T' with sample complexity $O(\mathsf{p} \cdot \mathsf{s})$.

The key observation is that the samples that T draws are completely independent of the *proof* that it receives. Since we aim to minimize sample complexity (rather than time complexity), we can hope to emulate all possible proofs, while reusing the samples. However, since there are exponentially many (2^{p}) possible proofs, we need to amplify the soundness to assure no error occurs with high probability. To this end, at the cost of increasing the sample complexity to $O(\mathsf{p} \cdot \mathsf{s})$, we invoke the tester $O(\mathsf{p})$ times to obtain soundness error $\exp(-\mathsf{p})$, which suffices to take a union bound over invocations of the amplified T with respect to all possible proofs.

To lift the BPP lower bound to r-round AM distribution testers, for *any* (possibly non-constant) $r \geq 1$, we need a significantly more involved argument. Recall that an AM distribution tester works as follows. In each round, the tester samples fresh randomness ρ_i and sends it to the prover, which replies with a message m_i that may arbitrarily depend on the input distribution $\mathcal{D} \in \Delta(\Omega_{\mathsf{n}})$, proximity parameter ε, and transcript of the interaction so far. After receiving the last message from the prover, the tester draws samples from $\mathcal{D}$ and decides according to these samples, proximity parameter, and transcript of the entire interaction.

Analogously to the proof of the MA lifting lemma, the high-level idea is that since the samples drawn from $\mathcal{D}$ are independent of the transcript of interaction, a BPP distribution tester can emulate all possible interactions, while using the *same* samples for *all* invocations. However, several difficulties arise when trying to naively implement the foregoing idea.

First, note that the tester cannot simply emulate the optimal prover, because it is determined by a distribution from which it only has few samples. Second, we cannot afford to enumerate over all prover *strategies*, as there is a doubly exponential number of them (each strategy is a function from the space of previous transcripts to the next message). Instead, we can only afford enumerating over all possible *transcripts*, which are *not* uniformly generated. Third, as before, since we invoke the tester with respect to exponentially many transcripts, we need to reduce its soundness error accordingly. Unfortunately, amplifying the soundness would result in an increase in communication complexity, which we cannot afford. Finally, even given exponentially small soundness error, whereas for MA it suffices to find a single proof that is accepted with high probability, here there may exist specific

transcripts in which the prover fools the tester with probability 1 (this is because we consider transcripts, rather than prover strategies).

A key step towards overcoming these difficulties is to rely on a simple yet important observation: each AM distribution tester induces a family of BPP distribution testers that are determined by the interaction. That is, since the *transcript* of the interaction is a random variable that is *independent of the samples* drawn by the AM distribution tester, the interaction phase can be viewed as a procedure that defines a BPP distribution tester that is invoked after this phase. In particular, this allows us to perform soundness amplification *solely on the induced BPP distribution testers*.

The procedure above implies that, with high probability over the random messages of the tester, each of the corresponding induced BPP distribution testers decides correctly, with only an exponentially small probability of error, without incurring any blowup in communication complexity. (Note, however, that the total soundness of the AM distribution tester does not necessarily increase significantly.)

Thus, we can invoke all the BPP distribution testers that are induced by all possible transcripts, while reusing the same samples for all invocations, such that with high probability no error will occur in any of the relevant invocations. Finally, we show that the the interaction tree induced by these invocations is significantly different for yes-instance and no-instances, and so the tester can consider it and decide whether there exists a prover strategy that would have been accepted with high probability by the AM distribution tester.

5.3 Derandomization

The key observation behind the derandomization of MA distribution testers (Theorem 3) is that while an NP distribution tester is a *deterministic* algorithm, it receives *random* samples from the input distribution $\mathcal{D}$. Thus we can hope to simulate the coin tosses of the MA distribution tester by deterministically extracting the necessary randomness from the samples.

To deterministically extract uniform bits from independent samples drawn from a distribution $\mathcal{D} \in \Delta([\mathsf{n}])$, we arbitrarily group the samples into pairs, discard pairs in which both samples are the same, then write 1 (respectively, 0) for every pair in which the first element is larger (respectively, smaller) than the second. Since the samples are independent, the first sample of each pair is equally likely to be larger as it is to be smaller than the second sample, and so we obtain a uniformly distributed string. This procedure can be thought of as generalizing the seedless extractor of Von Neumann [60].

The foregoing approach raises two concerns: (a) if $\mathcal{D}$ has small entropy, each bit we extract will require many samples (as many pairs would be discarded); and (b) even if $\mathcal{D}$ has large entropy, the MA distribution tester may toss a large number of coins, and so we shall need to draw many samples accordingly.

The first concern can be easily handled by observing that distributions with small entropy can be efficiently *learned*, and so we can test them with few samples, even without the aid of a prover. Dealing with the second concern is significantly more involved, and requires proving a randomness reduction lemma for MA distribution testers, which shows that it suffices to extract a *small* number of uniformly random bits, roughly logarithmic in the domain size.

The proof of the aforementioned randomness reduction lemma follows the randomness reduction approach of Goldreich and Sheffet [35], but our different setting requires several new ideas. In particular, our model involves testers that access a proof and two sources of randomness and, most significantly, the argument in [35] crucially relies on a bound on the number of inputs that the tester can receive, but no such bound exists in our setting.

Acknowledgements. We are grateful to Oded Goldreich and Rocco Servedio for multiple technical and conceptual suggestions that greatly improved the results of this work and extended its scope. We thank Clément Canonne for many discussions concerning distribution testing and for offering advice regarding several specific topics. We thank Igor Shinkar and Nicholas Spooner for useful discussions.

References

1 Scott Aaronson and Avi Wigderson. Algebrization: A new barrier in complexity theory. *ACM Transactions on Computation Theory*, 1:2:1–2:54, 2009.

2 Jayadev Acharya, Clément L. Canonne, and Gautam Kamath. A chasm between identity and equivalence testing with conditional queries. In *Proceedings of the 19th International Workshop on Randomization and Computation*, RANDOM '15, pages 449–466, 2015.

3 Jayadev Acharya, Hirakendu Das, Ashkan Jafarpour, Alon Orlitsky, and Shengjun Pan. Competitive closeness testing. In *Proceedings of the 24th Annual Conference on Learning Theory*, COLT 2011, pages 47–68, 2011.

4 László Babai, Peter Frankl, and Janos Simon. Complexity classes in communication complexity theory. In *Proceedings of the 27th Annual IEEE Symposium on Foundations of Computer Science*, FOCS 1986, pages 337–347, 1986.

5 László Babai and Shlomo Moran. Arthur-merlin games: a randomized proof system, and a hierarchy of complexity classes. *Journal of Computer and System Sciences*, 36:254–276, 1988.

6 Tuğkan Batu, Sanjoy Dasgupta, Ravi Kumar, and Ronitt Rubinfeld. The complexity of approximating the entropy. *SIAM Journal on Computing*, 35(1):132–150, 2005.

7 Tuğkan Batu, Eldar Fischer, Lance Fortnow, Ravi Kumar, Ronitt Rubinfeld, and Patrick White. Testing random variables for independence and identity. In *Proceedings of the 42nd Annual Symposium on Foundations of Computer Science*, FOCS 2001, pages 442–451, 2001.

8 Tuğkan Batu, Lance Fortnow, Ronitt Rubinfeld, Warren D. Smith, and Patrick White. Testing that distributions are close. In *Proceedings of the 41st Annual Symposium on Foundations of Computer Science*, FOCS 2000, pages 259–269, 2000.

9 Eli Ben-Sasson, Oded Goldreich, Prahladh Harsha, Madhu Sudan, and Salil P. Vadhan. Robust PCPs of proximity, shorter PCPs, and applications to coding. *SIAM Journal on Computing*, 36(4):889–974, 2006.

10 Itay Berman, Ron D. Rothblum, and Vinod Vaikuntanathan. Zero-knowledge proofs of proximity. In *Proceedings of the 9th Innovations in Theoretical Computer Science Conference*, ITCS 2018, page To appear, 2018.

11 Bhaswar B. Bhattacharya and Gregory Valiant. Testing closeness with unequal sized samples. In *Proceedings of the 2015 Conference on Neural Information Processing Systems*, NIPS 2015, pages 2611–2619, 2015.

12 Arnab Bhattacharyya, Eldar Fischer, Ronitt Rubinfeld, and Paul Valiant. Testing monotonicity of distributions over general partial orders. In *Proceedings of the 2nd Innovations in Theoretical Computer Science Conference*, ITCS 2011, pages 239–252, 2011.

13 Eric Blais, Clément L. Canonne, and Tom Gur. Distribution testing lower bounds via reductions from communication complexity (Alice and Bob don't talk to each other anymore.). In *Proceedings of the 32th Conference on Computational Complexity*, CCC 2017, pages 1–42, 2017.

14 Kenneth P Burnham and David R Anderson. *Model selection and multimodel inference: a practical information-theoretic approach.* Springer Science & Business Media, 2003.

15 Clément Canonne and Ronitt Rubinfeld. Testing probability distributions underlying aggregated data. In *International Colloquium on Automata, Languages, and Programming*, ICALP '14, pages 283–295, 2014.

16 Clément L. Canonne. A survey on distribution testing. your data is big. But is it blue?, 2017.

17 Clément L. Canonne, Elena Grigorescu, Siyao Guo, Akash Kumar, and Karl Wimmer. Testing k-monotonicity. *Electronic Colloquium on Computational Complexity (ECCC)*, 23:136, 2016. URL: `http://eccc.hpi-web.de/report/2016/136`.

18 Clément L. Canonne, Dana Ron, and Rocco A. Servedio. Testing probability distributions using conditional samples. *SIAM Journal on Computing*, 44(3):540–616, 2015.

19 Amit Chakrabarti, Graham Cormode, Andrew McGregor, and Justin Thaler. Annotations in data streams. *ACM Transactions on Algorithms*, 11, 2014.

20 Amit Chakrabarti, Graham Cormode, Andrew McGregor, Justin Thaler, and Suresh Venkatasubramanian. Verifiable stream computation and Arthur-Merlin communication. In *Proceedings of the 30th Conference on Computational Complexity*, CCC 2015, pages 217–243, 2015.

21 Sourav Chakraborty, Eldar Fischer, Yonatan Goldhirsh, and Arie Matsliah. On the power of conditional samples in distribution testing. *SIAM Journal on Computing*, 45(4):1261–1296, 2016.

22 Siu-On Chan, Ilias Diakonikolas, Gregory Valiant, and Paul Valiant. Optimal algorithms for testing closeness of discrete distributions. In *Proceedings of the 25th Symposium on Discrete Algorithms*, SODA 2014, pages 1193–1203, 2014.

23 Alessandro Chiesa and Tom Gur. Proofs of proximity for distribution testing. *Electronic Colloquium on Computational Complexity (ECCC)*, 24:155, 2017. URL: `https://eccc.weizmann.ac.il/report/2017/155`.

24 Ilias Diakonikolas and Daniel M. Kane. A new approach for testing properties of discrete distributions. In *Proceedings of the 57th Annual Symposium on Foundations of Computer Science*, FOCS 2016, pages 685–694, 2016.

25 Funda Ergün, Ravi Kumar, and Ronitt Rubinfeld. Fast approximate probabilistically checkable proofs. *Information and Computation*, 189(2):135–159, 2004.

26 Moein Falahatgar, Ashkan Jafarpour, Alon Orlitsky, Venkatadheeraj Pichapati, and Ananda Theertha Suresh. Faster algorithms for testing under conditional sampling. In *Conference on Learning Theory*, COLT '15, pages 607–636, 2015.

27 Eldar Fischer, Yonatan Goldhirsh, and Oded Lachish. Partial tests, universal tests and decomposability. In *Proceedings of the 5th Innovations in Theoretical Computer Science Conference*, ITCS 2014, pages 483–500, 2014.

28 Eldar Fischer, Oded Lachish, and Yadu Vasudev. Trading query complexity for sample-based testing and multi-testing scalability. In *Proceedings of the 56th Symposium on Foundations of Computer Science*, FOCS 2015, pages 1163–1182, 2015.

29 Oded Goldreich. Introduction to property testing, 2017.

30 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.

31 Oded Goldreich and Tom Gur. Universal locally verifiable codes and 3-round interactive proofs of proximity for CSP. *Electronic Colloquium on Computational Complexity (ECCC)*, 23:192, 2016. URL: `http://eccc.hpi-web.de/report/2016/192`.

32 Oded Goldreich, Tom Gur, and Ilan Komargodski. Strong locally testable codes with relaxed local decoders. In *Proceedings of the 30th Conference on Computational Complexity*, CCC 2015, pages 1–41, 2015.

33 Oded Goldreich, Tom Gur, and Ron D. Rothblum. Proofs of proximity for context-free languages and read-once branching programs. In *Proceedings of the 42nd International Colloquium on Automata, Languages, and Programming*, ICALP 2015, pages 666–677, 2015.

34 Oded Goldreich and Dana Ron. On testing expansion in bounded-degree graphs. In *Studies in Complexity and Cryptography*, pages 68–75. Springer, 2011.

35 Oded Goldreich and Or Sheffet. On the randomness complexity of property testing. *Computational Complexity*, 19(1):99–133, 2010.

36 Shafi Goldwasser, Silvio Micali, and Charles Rackoff. The knowledge complexity of interactive proof systems. *SIAM Journal on computing*, 18(1):186–208, 1989.

37 Shafi Goldwasser and Michael Sipser. Private coins versus public coins in interactive proof systems. In *Proceedings of the 18th a Annual ACM Symposium on Theory of Computing*, STOC 1986, pages 59–68, 1986.

38 Tom Gur and Ran Raz. Arthur-Merlin streaming complexity. *Information and Computing*, 243:145–165, 2015.

39 Tom Gur and Ron Rothblum. Non-interactive proofs of proximity. *Computational Complexity*, To appear, 2017.

40 Tom Gur and Ron D. Rothblum. A hierarchy theorem for interactive proofs of proximity. In *Proceedings of the 8th Innovations in Theoretical Computer Science Conference*, ITCS 2017, 2017.

41 Yael Tauman Kalai and Ron D. Rothblum. Arguments of proximity. In *Proceedings of the 35th Annual International Cryptology Conference*, CRYPTO 2015, pages 422–442, 2015.

42 Hartmut Klauck. On Arthur Merlin games in communication complexity. In *Proceedings of the 26th Conference on Computational Complexity*, CCC 2017, pages 189–199, 2011.

43 Erich L Lehmann and Joseph P Romano. *Testing statistical hypotheses*. Springer Science & Business Media, 2006.

44 Reut Levi, Dana Ron, and Ronitt Rubinfeld. Testing properties of collections of distributions. *Theory of Computing*, 9:295–347, 2013.

45 Ilan Newman and Mario Szegedy. Public vs. private coin flips in one round communication games. In *Proceedings of the 28th a Annual ACM Symposium on Theory of Computing*, STOC 1996, pages 561–570, 1996.

46 Liam Paninski. Estimating entropy on m bins given fewer than m samples. *IEEE Transactions on Information Theory*, 50(9):2200–2203, 2004.

47 Liam Paninski. A coincidence-based test for uniformity given very sparsely sampled discrete data. *IEEE Transactions on Information Theory*, 54(10):4750–4755, 2008.

48 Aditi Raghunathan, Gregory Valiant, and James Zou. Estimating the unseen from multiple populations. *CoRR*, abs/1707.03854, 2017. `arXiv:1707.03854`.

49 Sofya Raskhodnikova, Dana Ron, Amir Shpilka, and Adam Smith. Strong lower bounds for approximating distribution support size and the distinct elements problem. *SIAM Journal on Computing*, 39:813–842, 2009.

50 Ran Raz and Amir Shpilka. On the power of quantum proofs. In *Proceedings of the 19th Conference on Computational Complexity*, CCC 2004, pages 260–274, 2004.

51 Omer Reingold, Ron Rothblum, and Guy Rothblum. Constant-round interactive proofs for delegating computation. In *Proceedings of the 48th ACM Symposium on the Theory of Computing*, STOC 2016, pages 49–62, 2016.

52 Guy N. Rothblum, Salil P. Vadhan, and Avi Wigderson. Interactive proofs of proximity: delegating computation in sublinear time. In *Proceedings of the 45th Symposium on Theory of Computing*, STOC 2013, pages 793–802, 2013.

53 Ronitt Rubinfeld. Taming big probability distributions. *ACM Crossroads*, 19(1):24–28, 2012.

54 Ronitt Rubinfeld and Rocco Servedio. Testing monotone high-dimensional distributions. *Random Structures & Algorithms*, 34:24–44, 2009.

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

56 Gregory Valiant and Paul Valiant. Estimating the unseen: an $n/\log(n)$-sample estimator for entropy and support size, shown optimal via new CLTs. In *Proceedings of the 43rd Symposium on Theory of Computing*, STOC 2011, pages 685–694, 2011.

57 Gregory Valiant and Paul Valiant. An automatic inequality prover and instance optimal identity testing. *SIAM Journal on Computing*, 46(1):429–455, 2017.

58 Paul Valiant. Testing symmetric properties of distributions. *SIAM Journal on Computing*, 40:1927–1968, 2011.

59 Thomas Vidick and John Watrous. Quantum proofs. *Foundations and Trends in Theoretical Computer Science*, 11:1–215, 2016.

60 John Von Neumann. Various techniques used in connection with random digits. *National Bureau of Standards Applied Math Series*, 12:36–38, 1951.

61 John Watrous. Succinct quantum proofs for properties of finite groups. In *Proceedings of the 41st Annual IEEE Symposium on Foundations of Computer Science*, FOCS 2000, pages 537–546, 2000.

62 James Zou, Gregory Valiant, Paul Valiant, Konrad Karczewski, Siu On Chan, Kaitlin Samocha, Monkol Lek, Shamil Sunyaev, Mark Daly, and Daniel G MacArthur. Quantifying unobserved protein-coding variants in human populations provides a roadmap for large-scale sequencing projects. *Nature Communications*, 7, 2016.

Efficient Testing without Efficient Regularity

Lior Gishboliner[1] **and Asaf Shapira**[*2]

1 **School of Mathematical Sciences, Tel Aviv University, Tel Aviv, 69978, Israel**
liorgis1@post.tau.ac.il.
2 **School of Mathematical Sciences, Tel Aviv University, Tel Aviv 69978, Israel**
asafico@tau.ac.il.

Abstract

The regularity lemma of Szemerédi turned out to be the most powerful tool for studying the testability of graph properties in the dense graph model. In fact, as we argue in this paper, this lemma can be used in order to prove (essentially) all the previous results in this area. More precisely, a barrier for obtaining an efficient testing algorithm for a graph property $\mathcal{P}$ was having an efficient regularity lemma for graphs satisfying $\mathcal{P}$. The problem is that for many natural graph properties (e.g. triangle freeness) it is known that a graph can satisfy $\mathcal{P}$ and still only have regular partitions of tower-type size. This means that there was no viable path for obtaining reasonable bounds on the query complexity of testing such properties.

In this paper we consider the property of being induced C_4-free, which also suffers from the fact that a graph might satisfy this property but still have only regular partitions of tower-type size. By developing a new approach for this problem we manage to overcome this barrier and thus obtain a merely exponential bound for testing this property. This is the first substantial progress on a problem raised by Alon in 2001, and more recently by Alon, Conlon and Fox. We thus obtain the first example of an efficient testing algorithm that cannot be derived from an efficient version of the regularity lemma.

1998 ACM Subject Classification G.2.2 Graph Theory

Keywords and phrases Property testing, induced C_4-freeness

Digital Object Identifier 10.4230/LIPIcs.ITCS.2018.54

1 Introduction

The area of *property testing* was introduced in the seminal papers of Rubinfeld and Sudan [23] and Goldreich, Goldwasser and Ron [16]. As opposed to classical decision problems, where one is asked to decide if an input satisfies a predetermined property $\mathcal{P}$ or not, in property testing one is only asked to decide if the input satisfies $\mathcal{P}$ or is *far* from satisfying it. By now, problems of this type have been studied in so many areas that it will be impossible to survey them here. We thus refer the reader to the upcoming book of Goldreich [15] for more background and references on the subject.

Our focus in this paper will be testing graph properties in the *dense graph model*, introduced in the aforementioned [16], which was the first model in which property testing problems have been systematically studied. In this model, the input graph G is given via its $n \times n$ adjacency matrix, and we assume that there is an oracle that can answer queries of the form: is (i, j) an edge of G? We say that an n-vertex graph G is ε-far from satisfying property $\mathcal{P}$ if one should add/remove at least εn^2 edges in order to turn G into a graph

* Supported in part by ISF Grant 1028/16 and ERC Starting Grant 633509.

satisfying $\mathcal{P}$. An ε-tester for $\mathcal{P}$ is an algorithm that can distinguish with high probability (say, 2/3) between the case that G satisfies $\mathcal{P}$ and the case that G is ε-far from satisfying it. The maximum number of queries made by an ε-tester on n-vertex graphs is called its query complexity, and is denoted by $q(\varepsilon, n)$. We say that $\mathcal{P}$ is *testable* if it has an ε-tester which makes only $q(\varepsilon)$ queries, that is, whose query complexity depends only on ε and not on the size of the input. We say that $\mathcal{P}$ is easily testable if $q(\varepsilon) = \text{poly}(1/\varepsilon)$.

In Subsection 1.1 we (tersely) describe the main results of this paper. We elaborate on the relevant background, motivation, implications and significance of our results in Subsection 1.2.

1.1 The short story

A graph is induced H-free if it does not contain an induced copy of H. Alon, Fischer, Krivelevich and Szegedy [2] proved that for every fixed graph H, the property of being induced H-free is testable. Equivalently[1], this can be stated as saying that if an n-vertex graph G is ε-far from being induced H-free then G contains at least $n^h/q_H(\varepsilon)$ induced copies of H, where $h = |V(H)|$ and $q_H(\varepsilon)$ depends only on ε. The proof in [2] relied on the regularity lemma, and thus supplied very poor tower-type[2] bounds for $q_H(\varepsilon)$.

Alon [1] asked for which graphs H we have $q_H(\varepsilon) = \text{poly}(1/\varepsilon)$, that is, for which graphs H the property of being induced H-free is easily testable. This question was addressed by Alon and the second author [6] who resolved this problem for all graphs H save for P_3 (the path on 4 vertices) and C_4 (the 4-cycle). The former case was recently solved by Alon and Fox [5], who proved that $q_{P_3}(\varepsilon) = \text{poly}(1/\varepsilon)$. They further asked to determine if $q_{C_4}(\varepsilon) = \text{poly}(1/\varepsilon)$. This problem was also later raised by Conlon and Fox [10].

Prior to this work the best bound for $q_{C_4}(\varepsilon)$ was the same tower-type bound that holds for all graphs H. Our main result in this paper makes the first substantial progress on this problem.

▶ **Theorem 1. [Main Result]** *If an n-vertex graph G is ε-far from being induced C_4-free then G contains at least $n^4/2^{(1/\varepsilon)^c}$ induced copies of C_4, where c is an absolute constant. In particular, induced C_4-freeness is testable with query complexity $2^{(1/\varepsilon)^c}$.*

We strongly believe that the exponential bound in Theorem 1 can be further improved to a polynomial one, which would thus show that induced C_4-freeness is easily testable.

Given a (possibly infinite) family of graphs $\mathcal{F}$, we say that a graph is induced $\mathcal{F}$-free if it is induced H-free for every $H \in \mathcal{F}$. The result of [2] was extended by Alon and the second author [7] who showed that for every family of graphs $\mathcal{F}$, the property of being induced $\mathcal{F}$-free is testable. Needless to say that as in [2], the bounds involved were also of tower-type. It is natural to ask if Theorem 1 can be extended to properties defined by forbidding a family of graphs $\mathcal{F}$, one of which is C_4. The most notable and natural example is the property of being *chordal*, which is the property of not containing an induced cycle of length at least 4. Previously, the best bound for testing this property was the tower-type bound which follows from the general result of [7]. Here we obtain the following improved bound.

---

[1] This statement is usually referred to as a *removal lemma*, after the triangle removal lemma of Ruzsa and Szemerédi [24] from 1976. So in some sense, the result of [24] was the first statement in graph property testing. Also, see [1] for the short argument showing the equivalence between these two formulations.

[2] tower(x) is a tower of exponents of height x, so tower(3) $= 2^{2^2}$. In fact, the proof in [2] gave wowzer-type bounds which were later improved in [9].

▶ **Theorem 2.** *If an n-vertex graph G is ε-far from being chordal then for some $4 \leq \ell \leq O(\varepsilon^{-18})$, G contains at least $n^\ell/2^{(1/\varepsilon)^c}$ induced copies of C_ℓ, where c is an absolute constant. In particular, chordality is testable with query complexity $2^{(1/\varepsilon)^c}$.*

It is now natural to ask if Theorem 2 can be further extended to an arbitrary family $\mathcal{F}$, one of which is C_4. As our final theorem shows, this is not the case in a very strong sense.

▶ **Theorem 3.** *For every (monotone increasing) function $g : (0, 1/2) \to \mathbb{N}$ there is a family of graphs $\mathcal{F} = \mathcal{F}(g)$ so that $C_4 \in \mathcal{F}$ and the following holds. For every (small enough) $\varepsilon > 0$ and every $n \geq n_0(\varepsilon)$, there is an n-vertex graph G which is ε-far from being induced $\mathcal{F}$-free, and yet does not contain an induced copy of any $F \in \mathcal{F}$ on fewer than $g(\varepsilon)$ vertices.*

1.2 The long story

In this subsection we would like to describe the main significance of the results stated in the previous subsection. The famous regularity lemma of Szemerédi [25] is one of the most powerful tools for tackling problems in extremal graph theory. Roughly speaking, the lemma supplies a short description of a graph via a highly structured *regular partition* of its vertices. Given the nature of the problems studied in the area of property testing, it is no surprise that this lemma has also turned out to be a powerful tool in this area. In fact, it was shown in [4] that a property can be tested if and only if it is (more or less) equivalent to the property of having certain regular partitions. In other words, the regularity lemma gives a *qualitative* explanation as to which properties are testable.

Prior to this work, the relation between the regularity lemma and graph property testing was not only qualitative but also *quantitative*. In other words, the bounds one could obtain for the regularity lemma in graphs satisfying $\mathcal{P}$ determined the bounds one could obtain for testing $\mathcal{P}$ (with one important exception discussed below). Thanks to the work of Gowers [18], we know that in the *worst case*, a graph can have only regular partitions of tower-type size. However, when designing a property testing algorithm for a property $\mathcal{P}$, one can try to prove that graphs satisfying $\mathcal{P}$ must possess much smaller regular partitions. And indeed, as we have recently shown [14], almost all the known results giving non-tower-type bounds for testing graph properties $\mathcal{P}$ in the dense model, stem from the fact that graphs satisfying $\mathcal{P}$ have small regular partitions. For example, the result of [5] showing that induced P_3-freeness is easily testable can be derived from the fact that an induced P_3-free graph has a regular partition of polynomial size. Same goes for the polynomial testability results of [1, 6, 8, 17].

We now describe the only exception to the above quantitative relation between regularity and testing. In 1984 Erdős [12] (implicitly) conjectured that k-colorability is testable. This was verified by Rödl and Duke [22] who used the regularity lemma in order to show that k-colorability is testable with a tower-type bound. This tower-type bound was dramatically improved by Goldreich, Goldwasser and Ron [16], who showed that various *partition properties*, such as Max-Cut and k-colorability are easily testable, while *not* relying on the regularity lemma. Let us try to explain the reason for this exception: first, as opposed to triangle-freeness or induced C_4-freeness which are *local* properties, the partition properties of [16] are *global*. Perhaps the best way to see this is from the perspective of graph homomorphisms: triangle-freeness means that there is no edge preserving mapping from the vertices of the triangle to the vertices of G, while 3-colorability means that there is such a mapping from the vertices of G to the vertices of the triangle[3]. The second difference, which is more important

[3] In the language of graph limits, this is the distinction between left and right homomorphisms, see [20].

for our quantitative investigation here, is that at their core, these partition properties are "edge density" properties. This can explain (at least in hindsight), why one does not need any structure theorem in order to handle these problems, and can instead rely on sampling arguments that boil down to estimating various edge densities (this is not to say that devising such proofs is an easy task!).

Given the above discussion, one can ask why then one cannot get better bounds for testing induced H-freeness for every H. It is not hard to see that there are bipartite versions of Gowers's [18] example. Therefore, even for simple properties $\mathcal{P}$ such as triangle-freeness, a graph can satisfy $\mathcal{P}$ but still only have regular partitions of tower-type size. This means that any algorithm for testing triangle-freeness that relies on the regularity lemma is bound to produce tower-type bounds. In a major breakthrough, Fox [13] managed to prove the testability of triangle-freeness while avoiding Szemerédi's version of the regularity lemma, obtaining bounds that are still of tower-type, but only of height $O(\log 1/\varepsilon)$ instead of $\text{poly}(1/\varepsilon)$. A different formulation of his proof was later given in [10] and [21]. The latter proof shows that Fox's result can be derived from a variant of the regularity lemma. Unfortunately, it was shown in [21] that this variant of the regularity lemma must also produce partitions of tower-type size. Recapping, there is currently no viable approach for getting non-tower-type bounds, even for testing triangle-freeness.

With regards to induced C_4-freeness, it is not hard to check that every split graph is induced C_4-free, where a split graph is a graph whose vertex set can be partitioned into two sets, one spanning an independent set and the other spanning a complete graph. This means that if we take a bipartite version of Gowers' lower bound [18], and put a complete graph on one of the vertex sets, we get an induced C_4-free graph that has only regular partitions of tower-type size. In particular, arguments similar to those that were previously used in order to devise efficient testing algorithms cannot give better-than-tower-type bounds for this problem.

Summarizing the above discussion, Theorem 1 is the first example showing that one *can* obtain an efficient testing algorithm for a property $\mathcal{P}$ (or equivalently, an efficient removal lemma for $\mathcal{P}$) even though graphs satisfying $\mathcal{P}$ might have only regular partitions of tower-type size. In particular, Theorem 1 exhibits the strongest separation between bounds for testing a property $\mathcal{P}$ and bounds for the regularity lemma on graphs satisfying $\mathcal{P}$. We are hopeful that bounds similar to those obtained in Theorem 1, can be obtained for other properties for which the best known bounds are of tower-type, most notably triangle freeness.

1.3 Paper overview

The main idea of the proof is to show that (very roughly speaking) every induced C_4-free graph is a split graph. To be more precise, every induced C_4-free graph is close to being a union of an independent set and few cliques, so that the bipartite graphs between these cliques are highly structured. Note that we have no guarantee on the structure of the bipartite graph connecting the independent set and the cliques[4]. Towards this goal, in Section 2 we describe some preliminary lemmas, mostly regarding the structure of bipartite graphs that do not contain an induced matching of size 2. In Section 3 we give the main partial structure theorem, stated as Lemma 13. In the course of the proof we will make a surprising application of one of the main results of Goldreich, Goldwasser and Ron [16]. In Section 4 we

[4] This unstructured part is unavoidable due to the example we mentioned earlier of putting Gowers' construction between a clique and an independent set

give the proof of Theorems 1 and 2. We will make use of the structure theorem from Section 3 but will also have to deal with the (unavoidable) *unstructured* part of the graph. This will be done in Lemma 15. Finally, in Section 5, we give the proof of Theorem 3. Throughout the paper, we make no effort to optimize the constant c appearing in Theorems 1 and 2.

2 Forbidding an induced 2-matching

Our goal in this section is to introduce several definitions and prove Lemma 7 stated below, regarding graphs not containing induced matchings of size 2 of a specific type, which we now formally define. Let G be a graph and let $X, Y \subseteq V(G)$ be disjoint sets of vertices. An *induced copy of* M_2 in (X,Y) is an (unordered) quadruple x, x', y, y' such that $x, x' \in X$, $y, y' \in Y$, $(x,y), (x',y') \in E(G)$ and $(x,y'), (x',y) \notin E(G)$. We say that (X,Y) is *induced* M_2*-free* if it does not contain induced copies of M_2 as above. Observe that if X and Y are cliques then $G[X \cup Y]$ is induced C_4-free if and only if (X,Y) is induced M_2-free. For $x \in X$, we denote $N_Y(x) = \{y \in Y : (x,y) \in E(G)\}$.

▶ **Claim 4.** *(X,Y) is induced M_2-free if and only if there is an enumeration $x_1, \ldots, x_m$ of the elements of X such that $N_Y(x_i) \subseteq N_Y(x_j)$ for every $1 \leq i < j \leq m$.*

Proof. Observe that (X,Y) contains an induced M_2 if and only if there are $x, x' \in X$ for which there exist $y \in N_Y(x) \setminus N_Y(x')$ and $y' \in N_Y(x') \setminus N_Y(x)$. Therefore, (X,Y) is induced M_2-free if and only if for every $x, x' \in X$ it holds that either $N_Y(x) \subseteq N_Y(x')$ or $N_Y(x') \subseteq N_Y(x)$. Consider the poset on X in which x precedes x' if and only if $N_Y(x) \subseteq N_Y(x')$. This poset is a total order by the above. Enumerate the elements of X from minimal to maximal to get the required enumeration. ◀

We say that (X,Y) is *homogeneous* if the bipartite graph between X and Y is either complete or empty. We say that a partition $\mathcal{P} = \{P_1, \ldots, P_r\}$ of a set V is an *equipartition* if $||P_i| - |P_j|| \leq 1$ for every $1 \leq i, j \leq r$.

▶ **Lemma 5.** *If (X,Y) is induced M_2-free then for every integer $r \geq 1$ there is an equipartition $X = X_1 \cup \cdots \cup X_r$ and a partition $Y = Y_1 \cup \cdots \cup Y_{r+1}$ such that (X_i, Y_j) is homogeneous for every $1 \leq i \leq r$ and $1 \leq j \leq r+1$ satisfying $i \neq j$.*

Proof. Let $x_1, \ldots, x_m$ be the enumeration of the elements of X from Claim 4. For $1 \leq i \leq r$ define $X_i = \{x_j : \frac{(i-1)m}{r} < j \leq \frac{im}{r}\}$. Here we assume, for simplicity of presentation, that $|X|$ is divisible by r; if that is not the case then we partition X into "consecutive intervals" of sizes $\lfloor \frac{|X|}{r} \rfloor$ and $\lceil \frac{|X|}{r} \rceil$. Let now $y_1, ..., y_n$ be an enumeration of the elements of Y with the property that for every $x \in X$, the set $N_Y(x)$ is a "prefix" of the enumeration, that is, so that $N_Y(x) = \{y_1, \ldots, y_k\}$ for some $0 \leq k \leq n$. Define $Y_1 = N_Y(x_{m/r})$, $Y_i = N_Y(x_{im/r}) \setminus N_Y(x_{(i-1)m/r})$ for $i = 2, \ldots, r$ and $Y_{r+1} = Y \setminus N_Y(x_m)$.

It remains to show that (X_i, Y_j) is homogeneous for every $i \neq j$. Assume first that $i < j$. Then for every $x \in X_i$ we have $N_Y(x) \subseteq N_Y(x_{im/r}) \subseteq N_Y(x_{(j-1)m/r})$. By the definition of Y_j we have $Y_j \cap N_Y(x_{(j-1)m/r}) = \emptyset$. Thus, $Y_j \cap N_Y(x) = \emptyset$ for every $x \in X_i$, implying that the bipartite graph (X_i, Y_j) is empty. Now assume that $i > j$. For every $x \in X_i$ we have $N_Y(x_{jm/r}) \subseteq N_Y(x_{(i-1)m/r}) \subseteq N_Y(x)$. By the definition of Y_j we have $Y_j \subseteq N_Y(x_{jm/r})$. Thus, $Y_j \subseteq N_Y(x)$ for every $x \in X_i$, implying that the bipartite graph (X_i, Y_j) is complete. ◀

For two partitions $\mathcal{P}_1, \mathcal{P}_2$ of the same set, we say that $\mathcal{P}_2$ is a *refinement* of $\mathcal{P}_1$ if every part of $\mathcal{P}_2$ is contained in one of the parts of $\mathcal{P}_1$. A vertex partition $\mathcal{P}$ of an n-vertex graph

G is called *δ-homogeneous* if the sum of $|U||V|$ over all non-homogeneous unordered distinct pairs $U, V \in \mathcal{P}$ is at most δn^2. It is easy to see that a refinement of a δ-homogeneous partition is itself δ-homogeneous.

▶ **Lemma 6.** *Let $\delta > 0$, let G be an n-vertex graph and let $V(G) = X_1 \cup \cdots \cup X_k$ be a partition such that $X_1, \ldots, X_k$ are cliques and (X_i, X_j) is induced M_2-free for every $1 \leq i < j \leq k$. Then there is a δ-homogeneous partition which refines $\{X_1, \ldots, X_k\}$ and has at most $k\,(2/\delta)^k$ parts.*

Proof. For every $1 \leq i < j \leq k$, we apply Lemma 5 to (X_i, X_j) with parameter $r = \frac{1}{\delta}$ to get partitions $\mathcal{P}_{i,j}$ of X_i and $\mathcal{P}_{j,i}$ of X_j, $\mathcal{P}_{i,j} = \{X_{i,j}^1, ..., X_{i,j}^r\}$, $\mathcal{P}_{j,i} = \{X_{j,i}^1, ..., X_{j,i}^{r+1}\}$, such that $\mathcal{P}_{i,j}$ is an equipartition and $(X_{i,j}^p, X_{j,i}^q)$ is homogeneous for every $p \neq q$. Note that

$$\sum_{p=1}^{r} |X_{i,j}^p||X_{j,i}^p| = \sum_{p=1}^{r} \frac{1}{r}|X_i||X_{j,i}^p| \leq \frac{1}{r}|X_i||X_j| = \delta|X_i||X_j|. \tag{1}$$

For every $i = 1, ..., k$, define $\mathcal{P}_i$ to be the common refinement of the partitions $(\mathcal{P}_{i,j})_{1 \leq j \leq k,\, j \neq i}$. We have $|\mathcal{P}_i| \leq (r+1)^{k-1} \leq (2/\delta)^k$. The partition $\mathcal{P} := \bigcup_{i=1}^{k} \mathcal{P}_i$ refines $\{X_1, \ldots, X_k\}$ and has at most $k\,(2/\delta)^k$ parts. For every $U, V \in \mathcal{P}$, if (U, V) is not homogeneous, then there are $1 \leq i < j \leq k$ and $1 \leq p \leq r$ such that $U \subseteq X_{i,j}^p$ and $V \subseteq X_{j,i}^p$. This follows from the fact that $X_1, \ldots, X_k$ are cliques and the property of the partitions $(\mathcal{P}_{i,j})_{1 \leq i \neq j \leq k}$. By (1), we have

$$\sum_{1 \leq i < j \leq k} \sum_{p=1}^{r} |X_{i,j}^p||X_{j,i}^p| \leq \delta \sum_{1 \leq i < j \leq k} |X_i||X_j| \leq \delta n^2,$$

implying that $\mathcal{P}$ is δ-homogeneous, as required. ◀

▶ **Lemma 7.** *Let $\delta > 0$, let G be an n-vertex graph and let $V(G) = X_1 \cup \cdots \cup X_k$ be a partition such that $X_1, \ldots, X_k$ are cliques and (X_i, X_j) is induced M_2-free for every $1 \leq i < j \leq k$. Then there is a set $Z \subseteq V(G)$ of size $|Z| < \delta n$, a partition $V(G) \setminus Z = Q_1 \cup \cdots \cup Q_q$ which refines $\{X_1 \setminus Z, \ldots, X_k \setminus Z\}$ and subsets $W_i \subseteq Q_i$ such that the following hold.*

1. *The sum of $|Q_i||Q_j|$ over all non-homogeneous pairs (Q_i, Q_j), $1 \leq i < j \leq q$, is at most δn^2.*
2. *$|W_i| \geq (\delta/2k)^{10k^2} n$ for every $1 \leq i \leq q$ and (W_i, W_j) is homogeneous for every pair $1 \leq i < j \leq q$.*

Proof. Apply Lemma 6 to G with parameter δ to obtain a δ-homogeneous partition $\mathcal{P}$ which refines $\{X_1, \ldots, X_k\}$. Define $\mathcal{Q} = \{U \in \mathcal{P} : |U| \geq \frac{\delta n}{|\mathcal{P}|}\}$ and write $\mathcal{Q} = \{Q_1, \ldots, Q_q\}$. Then Item 1 holds since $\mathcal{P}$ is δ-homogeneous. Setting $Z = \bigcup_{U \in \mathcal{P} \setminus \mathcal{Q}} U$, notice that $\mathcal{Q}$ refines $\{X_1 \setminus Z, \ldots, X_k \setminus Z\}$ and that $|Z| < |\mathcal{P}| \cdot \frac{\delta n}{|\mathcal{P}|} = \delta n$. Apply Lemma 6 to G again (with respect to the same partition $\{X_1, \ldots, X_k\}$), now with parameter $\delta' := \frac{\delta^2}{8|\mathcal{P}|^4}$, to get a δ'-homogeneous partition $\mathcal{V}$ with at most $k(16|\mathcal{P}|^4/\delta^2)^k$ parts. Let $\mathcal{W}$ be the common refinement of $\mathcal{P}$ and $\mathcal{V}$ and note that $\mathcal{W}$ is δ'-homogeneous since it is a refinement of $\mathcal{V}$. Moreover,

$$|\mathcal{W}| \leq |\mathcal{P}| \cdot |\mathcal{V}| \leq |\mathcal{P}| \cdot k(16|\mathcal{P}|^4/\delta^2)^k. \tag{2}$$

For each $1 \leq i \leq q$, define $\mathcal{W}_i = \{W \in \mathcal{W} : W \subseteq Q_i\}$, choose a vertex $w_i \in Q_i$ uniformly at random and let $W_i \in \mathcal{W}_i$ be such that $w_i \in W_i$. We will show that with positive probability, the sets $W_1, ..., W_q$ satisfy the statement in Item 2. For $1 \leq i \leq q$, the probability that

$|W_i| < \frac{|Q_i|}{2q|\mathcal{W}|}$ is smaller than $\frac{|\mathcal{W}|\cdot\frac{|Q_i|}{2q|\mathcal{W}|}}{|Q_i|} = \frac{1}{2q}$. By the union bound, with probability larger than $\frac{1}{2}$, every $1 \leq i \leq q$ satisfies

$$|W_i| \geq \frac{|Q_i|}{2q|\mathcal{W}|} \geq \frac{\left(\frac{\delta^2}{16|\mathcal{P}|^4}\right)^k \delta n}{2k|\mathcal{P}|^3} \geq \frac{\delta^{3k}n}{k(2|\mathcal{P}|)^{7k}} \geq \frac{\delta^{3k}n}{k2^k(2/\delta)^{7k^2}} \geq \left(\frac{\delta}{2k}\right)^{10k^2} n \ ,$$

where in the second inequality we used $|Q_i| \geq \frac{\delta n}{|\mathcal{P}|}$, $q \leq |\mathcal{P}|$ and (2), and in the fourth inequality we used the bound on $|\mathcal{P}|$ given by Lemma 6. For $1 \leq i < j \leq q$, the probability that the pair (W_i, W_j) is not homogeneous is

$$\sum \frac{|W||W'|}{|Q_i||Q_j|} \leq \frac{4|\mathcal{P}|^2}{\delta^2 n^2} \sum |W||W'| \leq \frac{4|\mathcal{P}|^2}{\delta^2 n^2} \cdot \delta' n^2 \leq \frac{1}{2|\mathcal{P}|^2} \ ,$$

where the sums are taken over all non-homogeneous pairs $(W, W') \in \mathcal{W}_i \times \mathcal{W}_j$, the first inequality uses $|\mathcal{Q}_i|, |\mathcal{Q}_j| \geq \frac{\delta n}{2|\mathcal{P}|}$ and the second the fact that $\mathcal{W}$ is δ'-homogeneous. By the union bound, with probability at least $1 - \binom{q}{2}\frac{1}{|\mathcal{P}|} \geq 1 - \binom{|\mathcal{P}|}{2}\frac{1}{|\mathcal{P}|} > \frac{1}{2}$, all pairs (W_i, W_j) are homogeneous. We conclude that Item 2 holds with positive probability. ◀

It is worth mentioning that the bounds in the above lemma are the sole reason why our bound in Theorem 1 is exponential rather than polynomial.

3 A partial structure theorem for induced C_4-free graphs

Our main goal in this section is to prove Lemma 13 stated below, which gives an approximate partial structure theorem for induced C_4-free graphs. The "approximation" will be due to the fact that the graph will only be close to having a certain nice structure, while the "partial" will be since there will be a (possibly) big part of the graph about which we will have no control. As we discussed in Section 1, this partialness is unavoidable as evidenced by split graphs.

In addition to the lemmas from the previous section, we will also need the following theorems of Goldreich, Goldwasser and Ron [16] and of Gyárfás, Hubenko and Solymosi [19]. In both cases, $\omega(G)$ denotes the maximum size of a clique in G.

▶ **Theorem 8** ([16], Theorem 7.1). *For every $\varepsilon \in (0,1)$ there is $q_8(\varepsilon) = O(\varepsilon^{-5})$ with the following property. Let $\rho \in (0,1)$ be such that $\varepsilon < \rho^2/2$ and let G be a graph which is ε-far from containing a clique with at least ρn vertices. Suppose $q \geq q_8(\varepsilon)$ and let $Q \in \binom{V(G)}{q}$ be a randomly chosen set of q vertices of G. Then with probability at least $\frac{3}{4}$ we have $\omega(G[Q]) < (\rho - \frac{\varepsilon}{2})q$.*

▶ **Theorem 9** ([19]). *Every induced C_4-free graph G with n vertices and at least αn^2 edges satisfies $\omega(G) \geq 0.4\alpha^2 n$.*

Let use derive the following important corollary of the the above two theorems. For a non-empty set $X \subseteq V(G)$, define $d(X) = e(X)/\binom{|X|}{2}$, where $e(X)$ is the number of edges of G with both endpoints in X.

▶ **Lemma 10.** *Let $\alpha \in [0, \frac{1}{2})$ and let G be a graph on n vertices with at least αn^2 edges. Then for every $\beta \in (0,1)$, either G contains $\Omega(\alpha^{80}\beta^{20}n^4)$ induced copies of C_4 or there is a set $X \subseteq V(G)$ with $|X| \geq 0.1\alpha^2 n$ and $d(X) \geq 1 - \beta$.*

In the proof of Lemma 10 we need the following simple fact.

▶ **Claim 11.** *Let $\alpha \in (0,1)$ and let G be a graph with n vertices and at least αn^2 edges. Then for every $r \geq \frac{100}{\alpha^2}$, a sample of r vertices from G spans at least $\frac{\alpha}{2}r^2$ edges with probability at least $\frac{2}{3}$.*

The proof of Claim 11 is a standard application of Chebyshev's inequality, and is thus omitted.

Proof of Lemma 10. Set $\rho = 0.1\alpha^2$, $\varepsilon = \frac{\rho^2\beta}{4} = \frac{\alpha^4\beta}{400}$ and $r = \max\{q_8(\varepsilon), \frac{100}{\alpha^2}\}$. By Theorem 8 we have $r = O(\alpha^{-20}\beta^{-5})$. We assume that there is no $X \subseteq V(G)$ with $|X| \geq 0.1\alpha^2 n$ and $d(X) \geq 1-\beta$, and prove that G contains $\Omega(\alpha^{80}\beta^{20}n^4)$ induced copies of C_4. Let $X \subseteq V(G)$ be such that $|X| \geq \rho n$. Since $d(X) \leq 1-\beta$, we have $\binom{|X|}{2} - e(G) \geq \beta\binom{|X|}{2} \geq \beta\frac{|X|^2}{4} \geq \frac{\rho^2\beta}{4}n^2 = \varepsilon n^2$. This shows that G is ε-far from containing a clique of size ρn or larger. By our choice of r via Theorem 8, a random sample R of r vertices of G satisfies $\omega(G[R]) < (\rho - \frac{\varepsilon}{2})r < 0.1\alpha^2 r$ with probability at least $\frac{2}{3}$. By Claim 11, we also have $e(R) > \frac{\alpha}{2}r^2$ with probability at least $\frac{2}{3}$. So with probability at least $\frac{1}{3}$ we have both $\omega(G[R]) < 0.1\alpha^2 r$ and $e(R) > \frac{\alpha}{2}r^2$. If both events happen, then $G[R]$ must contain an induced copy of C_4, by Theorem 9. We conclude that G contains at least $\frac{1}{3}\binom{n}{r}/\binom{n-4}{r-4} = \frac{1}{3}\binom{n}{4}/\binom{r}{4} = \Omega(\alpha^{80}\beta^{20}n^4)$ induced copies of C_4. ◀

The last ingredient we need is the following result of Alon, Fischer and Newman [3]. For a pair of disjoint vertex sets X, Y, we say that (X,Y) is *ε-far* from being induced M_2-free if one has to add/delete at least $\varepsilon|X||Y|$ of the edges between X and Y to make (X,Y) induced M_2-free.

▶ **Lemma 12** ([3])**.** *There is an absolute constant $d > 0$ such that the following holds. If (X,Y) is ε-far from being induced M_2-free then (X,Y) contains at least $\varepsilon^d|X|^2|Y|^2$ induced copies of M_2.*

The following is the key lemma of this section. Note that it gives us a lot of information about $G[Y]$ and $G[X_1 \cup \cdots \cup X_k]$ but no information about the bipartite graph connecting $X_1 \cup \cdots \cup X_k$ and Y.

▶ **Lemma 13.** *There is an absolute constant $c > 0$, such that for every $\alpha, \gamma \in (0,1)$, every n-vertex graph G either contains $\Omega(\alpha^c\gamma^c n^4)$ induced copies of C_4, or admits a vertex partition $V(G) = X_1 \cup \cdots \cup X_k \cup Y$ with the following properties.*
1. *$e(Y) < \alpha n^2$.*
2. *$|X_i| \geq 0.1\alpha^3 n$ and $d(X_i) \geq 1-\gamma$ for every $1 \leq i \leq k$.*
3. *For every $1 \leq i < j \leq k$, the pair (X_i, X_j) is γ-close to being induced M_2-free.*

Proof. We prove the lemma with $c = \max(84, 20d)$, where d is the constant from Lemma 12. We inductively define two sequences of sets, $(V_i)_{i\geq 0}$ and $(X_i)_{i\geq 1}$. Set $V_0 = V(G)$. At the i'th step (starting from $i = 0$), if $e(V_i) < \alpha n^2$ then we stop. Note that if we did not stop then $|V_i| \geq \alpha n$. If $e(V_i) \geq \alpha n^2$ then by Lemma 10, applied to $G[V_i]$ with parameters α and $\beta = 0.25\gamma^d$, either $G[V_i]$ contains $\Omega(\alpha^{80}\gamma^{20d}|V_i|^4) \geq \Omega(\alpha^{84}\gamma^{20d}n^4)$ induced copies of C_4 or there is $X_{i+1} \subseteq V_i$ with $|X_{i+1}| \geq 0.1\alpha^2|V_i| \geq 0.1\alpha^3 n$ and $d(X_i) \geq 1 - 0.25\gamma^d$. If the former case happens then the assertion of the lemma holds, so we may assume that the latter case happens, in which case we set $V_{i+1} = V_i \setminus X_{i+1}$ and continue. Suppose that this process stops at the k'th step for some $k \geq 0$. Set $Y = V_k$. We clearly have $V(G) = X_1 \cup \cdots \cup X_k \cup Y$. For every $1 \leq i \leq k$ we have $|X_i| \geq 0.1\alpha^3 n$ and $d(X_i) \geq 1 - 0.25\gamma^d \geq 1 - \gamma$. Since the process stopped at the k'th step, we must have $e(Y) = e(V_k) < \alpha n^2$.

To finish the proof, we show that if Item 3 in the lemma does not hold then G contains at least $0.5 \cdot 10^{-4}\alpha^{12}\gamma^d n^4$ induced copies of C_4. Assume that for some $1 \leq i < j \leq k$, the

pair (X_i, X_j) is γ-far from being induced M_2-free. By Lemma 12, (X_i, X_j) contains at least $\gamma^d|X_i|^2|X_j|^2$ induced copies of M_2. Let (x_i, x_i', x_j, x_j') be such a copy, where $x_i, x_i' \in X_i$ and $x_j, x_j' \in X_j$. If $(x_i, x_i'), (x_j, x_j') \in E(G)$ then x_i, x_i', x_j, x_j' span an induced copy of C_4. Since $d(X_i), d(X_j) \geq 1 - 0.25\gamma^d$, There are at most $0.5\gamma^d|X_i|^2|X_j|^2$ quadruples of distinct vertices $(x_i, x_i', x_j, x_j') \in X_i \times X_i \times X_j \times X_j$ for which either $(x_i, x_i') \notin E(G)$ or $(x_j, x_j') \notin E(G)$. Thus, G contains at least $0.5\gamma^d|X_i|^2|X_j|^2 \geq 0.5 \cdot 10^{-4}\alpha^{12}\gamma^d n^4$ induced copies of C_4. ◀

We finish this section with the following corollary of the above structure theorem, which will be more convenient to use when proving Theorems 1 and 2 in the next section.

▶ **Lemma 14.** *There is an absolute constant $c > 0$ such that for every $\alpha, \gamma \in (0,1)$, every n-vertex graph G either contains $\Omega(\alpha^c\gamma^c n^4)$ induced copies of C_4 or there is a graph G' on $V(G)$, a partition $V(G) = X_1 \cup \cdots \cup X_k \cup Y$, where $k \leq 10\alpha^{-3}$, a subset $Z \subseteq X := X_1 \cup \cdots \cup X_k$, a partition $X \setminus Z = Q_1 \cup \cdots \cup Q_q$ which refines $\{X_1 \setminus Z, \ldots, X_k \setminus Z\}$, and subsets $W_i \subseteq Q_i$ with the following properties.*

1. *$G'[X_i \setminus Z]$ is a clique for every $1 \leq i \leq k$, and $G'[Y]$ is an independent set.*
2. *$|Z| < \alpha n$ and every $z \in Z$ is an isolated vertex in G'.*
3. *In G', the sum of $|Q_i||Q_j|$ over all non-homogeneous pairs (Q_i, Q_j), $1 \leq i < j \leq q$, is at most αn^2.*
4. *(W_i, W_j) is homogeneous in G' for every $1 \leq i < j \leq q$ and $|W_i| \geq (\alpha/20)^{4000\alpha^{-6}}|X|$ for every $1 \leq i \leq q$.*
5. *$|E(G')\triangle E(G)| < (2\alpha + \gamma)n^2$ and $|E(G'[X \setminus Z]) \triangle E(G[X \setminus Z])| < \gamma n^2$.*

Proof. The constant c in this lemma is the same as in Lemma 13. Apply Lemma 13 to G with the given α and γ. If G contains $\Omega\left(\alpha^c\gamma^c n^4\right)$ induced copies of C_4 then the assertion of the lemma holds, and otherwise let $X_1, \ldots, X_k, Y$ be as in the statement of Lemma 13. Note that $k \leq 10\alpha^{-3}$ since $|X_i| \geq 0.1\alpha^3$ for every $1 \leq i \leq k$. Let G'' be the graph obtained from G by making Y an independent set, making $X_1, \ldots, X_k$ cliques and making (X_i, X_j) induced M_2-free for every $1 \leq i < j \leq k$. By Lemma 13 we have $|E(G''[Y])\triangle E(G[Y])| < \alpha n^2$ and $|E(G''[X])\triangle E(G[X])| < \gamma \sum_{i=1}^{k} \binom{|X_i|}{2} + \gamma \sum_{i<j} |X_i||X_j| < \gamma n^2$. We now apply Lemma 7 to $G''[X]$ with parameter $\delta = \alpha$ (and with respect to the partition $\{X_1, \ldots, X_k\}$) and obtain a subset $Z \subseteq X$ of size $|Z| < \alpha|X| \leq \alpha n$, a partition $X \setminus Z = Q_q \cup \cdots \cup Q_q$ which refines $\{X_1 \setminus Z, \ldots, X_k \setminus Z\}$, and subsets $W_i \subseteq Q_i$ such that $|W_i| \geq (\alpha/2k)^{10k^2}|X| \geq (\alpha^4/20)^{1000\alpha^{-6}}|X| \geq (\alpha/20)^{4000\alpha^{-6}}|X|$ for every $1 \leq i \leq q$.

Let G' be the graph obtained from G'' by making every $z \in Z$ an isolated vertex. Then Item 2 is satisfied. The second part of Item 5 holds because $G'[X \setminus Z] = G''[X \setminus Z]$ and $|E(G''[X])\triangle E(G[X])| < \gamma n^2$. For the first part of Item 5, note that $|E(G')\triangle E(G'')| < |Z|n < \alpha n^2$, which implies that $|E(G')\triangle E(G)| \leq |E(G')\triangle E(G'')| + |E(G'')\triangle E(G)| < (2\alpha + \gamma)n^2$. Since $G'[X \setminus Z] = G''[X \setminus Z]$ and $G'[Y] = G''[Y]$, it is enough to establish that Items 1, 3 and 4 hold if G' is replaced by G''. For Item 1, this is immediate from the definition of G''; for items 3-4, this follows from our choice of $\mathcal{Q} = \{Q_1, \ldots, Q_q\}$ and $W_1, \ldots, W_q$ via Lemma 7 (with parameter $\delta = \alpha$). ◀

4 Proofs of main results

In this section we prove Theorems 1 and 2. The last ingredient we need is the following key lemma.

▶ **Lemma 15.** *Let $\mathcal{F}$ be a (finite or infinite) family of graphs such that*

1. *$C_4 \in \mathcal{F}$.*
2. *For every $F \in \mathcal{F}$ and $v \in V(F)$, the neighbourhood of v in F is not a clique.*

Suppose G is a graph with vertex partition $V(G) = X \cup Y$ such that Y is an independent set and $G[X]$ is induced $\mathcal{F}$-free. Then, if one must add/delete at least $\varepsilon|X||Y|$ of the edges between X and Y to make G induced $\mathcal{F}$-free, then G contains at least $\frac{\varepsilon^4}{2^8}|X|^2|Y|^2$ induced copies of C_4.

Proof. Let us pick for every $y \in Y$ a maximal anti-matching $\mathcal{M}(y)$ in $G[N_X(y)]$, that is, a maximal collection of pairwise-disjoint non-edges contained in $N_X(y)$. For every pair of non edges $(u,v),(u',v') \in \mathcal{M}(y)$, there must be at least one non-edge between $\{u,v\}$ and $\{u',v'\}$, as otherwise u,v,u',v' would span an induced C_4 in X, in contradiction to the assumptions that $G[X]$ is induced $\mathcal{F}$-free and $C_4 \in \mathcal{F}$. Therefore, for every y there are at least $\binom{|\mathcal{M}(y)|}{2} + |\mathcal{M}(y)| \geq |\mathcal{M}(y)|^2/2$ non-edges inside the set $N_X(y)$. For every $y \in Y$ let $d_2(y)$ denote the number of pairs of distinct vertices in $N_X(y)$ that are non-adjacent. Then the above discussion implies that every $y \in Y$ satisfies

$$d_2(y) \geq \frac{|\mathcal{M}(y)|^2}{2} \,. \tag{3}$$

Let G' be the graph obtained from G by deleting, for every $y \in Y$, all edges going between y and the vertices of $\mathcal{M}(y)$. Since $\mathcal{M}(y)$ is spanned by $2|\mathcal{M}(y)|$ vertices, we have

$$|E(G') \triangle E(G)| = 2 \sum_{y \in Y} |\mathcal{M}(y)| \,. \tag{4}$$

We now claim that G' is induced $\mathcal{F}$-free. Indeed, suppose $U \subseteq V(G)$ spans an induced copy of some $F \in \mathcal{F}$. Since by assumption $G[X]$ is induced $\mathcal{F}$-free and since $G'[X] = G[X]$, there must be some $y \in U \cap Y$. Since the neighbourhood of y in F is not a clique and since $G'[Y] = G[Y]$ is an empty graph, there must be $u,v \in U \cap X$ for which $u,v \in N_X(y)$ and $(u,v) \notin E(G')$. Now, the fact that u,v are connected to y in G' means that neither of them participated in one of the non-edges of $\mathcal{M}(y)$. But then the fact that $(u,v) \notin E(G')$ implies that also $(u,v) \notin E(G)$ (because we did not change $G[X]$) which in turn implies that (u,v) could have been added to $\mathcal{M}(y)$ contradicting its maximality.

By the assumption of the lemma we thus have $|E(G') \triangle E(G)| \geq \varepsilon|X||Y|$. Combining this with (3), (4) and Jensen's inequality thus gives

$$\begin{aligned}\sum_{y \in Y} d_2(y) \geq \frac{1}{2} \sum_{y \in Y} |\mathcal{M}(y)|^2 &\geq \frac{1}{2}|Y| \cdot \left(\frac{\sum_{y \in Y} |\mathcal{M}(y)|}{|Y|} \right)^2 \\ &= \frac{1}{2}|Y| \cdot \left(\frac{|E(G') \triangle E(G)|}{2|Y|} \right)^2 \geq \frac{\varepsilon^2}{8}|X|^2|Y|.\end{aligned}$$

For a pair of distinct vertices $u,v \in X$ set $t(u,v) = 0$ if $(u,v) \in E(G)$ and otherwise set $t(u,v)$ to be the number of vertices $y \in Y$ connected to both u and v. Recalling that Y is an independent set in G, we see that u,v belong to at least $\binom{t(u,v)}{2}$ induced copies of C_4. Hence, G contains at least

$$\begin{aligned}\sum_{u,v \in X} \binom{t(u,v)}{2} &\geq \binom{|X|}{2} \cdot \binom{\sum_{u,v \in X} t(u,v) / \binom{|X|}{2}}{2} \\ &= \binom{|X|}{2} \cdot \binom{\sum_{y \in Y} d_2(y) / \binom{|X|}{2}}{2} \\ &\geq \frac{|X|^2}{4} \cdot \frac{(\varepsilon^2|Y|/4)^2}{4} = \frac{\varepsilon^4}{2^8}|X|^2|Y|^2,\end{aligned}$$

induced copies of C_4, where the first inequality is Jensen's, the following equality is double-counting, and the last inequality uses our above lower bound for $\sum_{y \in Y} d_2(y)$. ◀

Proof of Theorem 1. We first observe that the "in particular" part of the statement, namely the testing algorithm, follows immediately from the first assertion of the theorem; indeed, the first assertion implies that if G is ε-far from being induced C_4-free, then sampling $2^{(1/\varepsilon)^c}$ 4-tuples of vertices will contain at least one induced 4-cycle with probability at least 2/3.

We now turn to prove the first assertion of the theorem. Set

$$\alpha = \frac{\varepsilon^6}{2^{11}}, \quad \gamma = \frac{1}{2}(\alpha/20)^{16000\alpha^{-6}}(\varepsilon/2)^4.$$

and notice that $\gamma \geq 2^{-(1/\varepsilon)^{c'}}$ for some absolute constant c'. We apply Lemma 14 to G with the α and γ defined above. If G contains $\Omega\left(\alpha^c\gamma^c n^4\right)$ induced copies of C_4 then we are done. Otherwise, let G', $X = X_1 \cup \cdots \cup X_k$, Y, Z, $\mathcal{Q} = \{Q_1, \ldots, Q_q\}$ and $W_i \subseteq Q_i$ be as in Lemma 14. Let G'' be the graph obtained from G' by doing the following: for every $1 \leq i < j \leq q$, if (W_i, W_j) is a complete (resp. empty) bipartite graph then we turn (Q_i, Q_j) into a complete (resp. empty) bipartite graph. By Item 4 in Lemma 14, one of these options holds. By Item 3 in Lemma 14, the number of changes made is at most αn^2. By Item 5 in Lemma 14 we have $|E(G'') \triangle E(G)| \leq |E(G'') \triangle E(G')| + |E(G') \triangle E(G)| < (3\alpha + \gamma)n^2 < \frac{\varepsilon}{2}n^2$, implying that G'' is $\frac{\varepsilon}{2}$-far from being induced C_4-free. Note that $|X \setminus Z| \geq \frac{\varepsilon}{2}n$, as otherwise deleting all edges incident to the vertices of $X \setminus Z$ would make G'' an empty graph (and hence induced C_4-free) by deleting $|X \setminus Z| \cdot n \leq \frac{\varepsilon}{2}n^2$ edges.

Let us assume first that $G''[X \setminus Z]$ contains an induced copy of C_4, say on the vertices v_1, v_2, v_3, v_4. For $1 \leq s \leq 4$, let i_s be such that $v_s \in Q_{i_s}$. It is easy to see that by the definition of G'', every quadruple $(w_1, \ldots, w_4) \in W_{i_1} \times W_{i_2} \times W_{i_3} \times W_{i_4}$ spans an induced copy of C_4 *in the graph* G'. By Item 4 in Lemma 14, G' contains

$$|W_{i_1}| \cdot |W_{i_2}| \cdot |W_{i_3}| \cdot |W_{i_4}| \geq (\alpha/20)^{16000\alpha^{-6}}|X|^4 \geq (\alpha/20)^{16000\alpha^{-6}}(\varepsilon/2)^4 n^4 = 2\gamma n^4$$

induced copies of C_4. By Item 5 in Lemma 14, $G[X \setminus Z]$ and $G'[X \setminus Z]$ differ on less than γn^2 edges, each of which can participate in at most n^2 induced copies of C_4. Thus, G contains at least γn^4 induced copies of C_4, as required.

From now on we assume that $G''[X \setminus Z]$ is induced C_4-free, implying that $G''[X]$ is induced C_4-free (as every $z \in Z$ is isolated in G''). Since G'' is $\frac{\varepsilon}{2}$-far from being induced C_4-free, one cannot make G'' induced C_4-free by adding/deleting less than $\frac{\varepsilon}{2}n^2 \geq \varepsilon|X||Y|$ edges between X and Y. In particular, we have $|X||Y| \geq \varepsilon n^2$. Notice that the conditions of Lemma 15 hold (with respect to the family $\mathcal{F} = \{C_4\}$) since $G''[Y] = G'[Y]$ is an independent set (by Item 1 in Lemma 14) and $G''[X]$ is induced C_4-free by assumption. By Lemma 15, G'' contains at least $\frac{\varepsilon^4}{2^8}|X|^2|Y|^2 \geq \frac{\varepsilon^6}{2^8}n^4 = 8\alpha n^4$ induced copies of C_4. Since $|E(G'') \triangle E(G)| < (3\alpha + \gamma)n^2 < 4\alpha n^2$, at least $4\alpha n^4 = \frac{\varepsilon^6}{2^9}n^4$ of these copies are also present in G. This completes the proof of the theorem. ◀

Proof of Theorem 2. Set

$$\alpha = \frac{\varepsilon^6}{2^{11}}, \quad \gamma = \frac{1}{2}(\alpha/20)^{10^5\alpha^{-9}}(\varepsilon/2)^{20\alpha^{-3}}.$$

and notice that $\gamma \geq 2^{-(1/\varepsilon)^{c'}}$ for some absolute constant c'. As in the proof of Theorem 1, we apply Lemma 14 to G with the α and γ defined above. If G contains $\Omega\left(\alpha^c\delta^c n^4\right)$ induced copies of C_4 then we are done. Otherwise, let G', $X = X_1 \cup \cdots \cup X_k$, Y, Z, $\mathcal{Q} = \{Q_1, \ldots, Q_q\}$ and $W_i \subseteq Q_i$ be as in Lemma 14.

Let G'' be the graph obtained from G' by doing the following: for every $1 \leq i < j \leq q$, if (W_i, W_j) is a complete (resp. empty) bipartite graph then we make (Q_i, Q_j) a complete

(resp. empty) bipartite graph. As in the proof of Theorem 1, G'' is $\frac{\varepsilon}{2}$-far from being chordal, and we have $|X \setminus Z| \geq \frac{\varepsilon}{2}n$.

Assume first that $G''[X \setminus Z]$ is not chordal, namely that it contains an induced cycle $C = v_1 \dots v_\ell$ of length $\ell \geq 4$. By Item 1 in Lemma 14, $G''[X_i \setminus Z] = G'[X_i \setminus Z]$ is a clique for every $1 \leq i \leq k$. Since the cycle C does not contain a triangle, it can contain at most 2 vertices from each of these cliques, implying that $\ell = |C| \leq 2k \leq 20\alpha^{-3} = O(\varepsilon^{-18})$. The bound on k comes from Lemma 14. For $1 \leq s \leq \ell$, let i_s be such that $v_s \in Q_{i_s}$. It is easy to see that by the definition of G'', ℓ-tuple $(w_1, \dots, w_\ell) \in W_{i_1} \times \dots \times W_{i_\ell}$ spans an induced ℓ-cycle *in the graph* G'. By Item 4 in Lemma 14, G' contains

$$\prod_{j=1}^{\ell} |W_{i_j}| \geq (\alpha/20)^{4000\alpha^{-6}\ell} |X|^\ell \geq (\alpha/20)^{10^5\alpha^{-9}} (\varepsilon/2)^{20\alpha^{-3}} n^\ell = 2\gamma n^\ell$$

induced copies of C_ℓ. By Item 5 in Lemma 14, $G[X]$ and $G'[X]$ differ on less than γn^2 edges, each of which can participate in at most $n^{\ell-2}$ induced copies of C_ℓ. Thus, G contains at least γn^ℓ induced copies of C_ℓ, as required.

We now assume that $G''[X]$ is chordal. Since G'' is $\frac{\varepsilon}{2}$-far from being chordal, one must add/delete at least $\frac{\varepsilon}{2}n^2 \geq \varepsilon|X||Y|$ of the edges between X and Y to make G'' chordal. In particular, we have $|X||Y| \geq \varepsilon n^2$. Note that the family $\mathcal{F} = \{C_\ell : \ell \geq 4\}$, i.e. the family of forbidden induced subgraphs for chordality, satisfies Conditions 1-2 of Lemma 15. Observe that Lemma 15 is applicable to G'' (with respect to the family $\mathcal{F} = \{C_\ell : \ell \geq 4\}$), as $G''[Y] = G'[Y]$ is an independent set (by Item 1 in Lemma 14), and $G''[X]$ is induced $\mathcal{F}$-free (i.e. chordal) by assumption. By Lemma 15, G'' contains at least $\frac{\varepsilon^4}{2^8}|X|^2|Y|^2 \geq \frac{\varepsilon^6}{2^8}n^4 = 8\alpha n^4$ induced copies of C_4. Since $|E(G'') \triangle E(G)| < 4\alpha n^2$, at least $4\alpha n^4 = \frac{\varepsilon^6}{2^9}n^4$ of these copies are also present in G. ◀

5 An impossibility result

In this section we prove Theorem 3. It will in fact be more convenient to prove the following equivalent statement.

▶ **Theorem 16.** *For every function $g : (0, \frac{1}{2}) \to \mathbb{N}$ there is a graph family $\mathcal{F}$ which contains C_4 and there is a sequence $\{\varepsilon_k\}_{k=1}^{\infty}$ with $\varepsilon_k > 0$ and $\varepsilon_k \to 0$, such the following holds. For every $k \geq 1$ and $n \geq n_0(k)$ there is an n-vertex graph G which is ε_k-far from being induced $\mathcal{F}$-free, but still every induced subgraph of G on $g(\varepsilon_k)$ vertices is induced $\mathcal{F}$-free.*

We will need the following theorem due to Erdős [11].

▶ **Theorem 17.** *For every integer f there is $n_{17} = n_{17}(k, f)$ such that every k-uniform hypegraph with $n \geq n_{17}$ vertices and $n^{k-f^{1-k}}$ edges contains a complete k-partite k-uniform hypergraph with f vertices in each part.*

For integers $k, f \geq 1$, let $B_{k,f}$ be the graph obtained by replacing each vertex of the cycle C_k by a clique of size f, and replacing each edge by a complete bipartite graph.

▶ **Lemma 18.** *For every pair of integers $k \geq 3$ and $f \geq 1$ there is $n_{18} = n_{18}(k, f)$ such that for every $n \geq n_{18}$, the graph $B_{k,n/k}$ is $\frac{1}{2k^2}$-far from being induced $\{C_4, B_{k,f}\}$-free.*

Proof. Let $V_1, \dots, V_k$ be the sides of $G := B_{k,n/k}$ (each a clique of size n/k). Let G' be a graph obtained from G by adding/deleting at most $\frac{v(G)^2}{2k^2} = \frac{n^2}{2k^2}$ edges. Our goal is to show that G' is not induced $\{C_4, B_{k,f}\}$-free. Let H be the k-partite k-uniform hypergraph

with parts $V_1, \dots, V_k$ whose edges are all k-tuples $(v_1, \dots, v_k) \in V_1 \times \cdots \times V_k$ such that $v_1 v_2 \dots v_k v_1$ is an induced cycle in G'. Note that in G, every such k-tuple spans an induced cycle, and that adding/deleting an edge can destroy at most $\left(\frac{n}{k}\right)^{k-2}$ such cycles. Thus, G' contains at least $\left(\frac{n}{k}\right)^k - \frac{n^2}{2k^2}\left(\frac{n}{k}\right)^{k-2} = \frac{1}{2}\left(\frac{n}{k}\right)^k$ of these induced cycles, implying that $e(H) \geq \frac{1}{2}\left(\frac{n}{k}\right)^k$. For a large enough n we have $\frac{1}{2}\left(\frac{n}{k}\right)^k \geq n^{k-f^{1-k}}$ and $n \geq n_{17}(k, f)$. Thus, by Theorem 17, H contains a complete k-partite k-uniform hypergraph with parts $U_i \subseteq V_i$, each of size f. This means that in the graph G', (U_i, U_j) is a complete bipartite graph if $j - i \equiv \pm 1 \pmod{k}$ and an empty bipartite graph otherwise. If $G'[U_i]$ is a clique for every $1 \leq i \leq k$ then $U_1 \cup \cdots \cup U_k$ spans an induced copy of $B_{k,f}$ in G'. Suppose then that U_i is not a clique for some $1 \leq i \leq k$, say $i = 1$, and let $x, y \in U_1$ be such that $(x, y) \notin E(G')$. Then for every $z \in U_2$ and $w \in U_k$, $\{x, y, z, w\}$ spans an induced copy of C_4 in G'. Thus, in any case G' is not induced $\{C_4, B_{k,f}\}$-free. ◀

Proof of Theorem 16. For $k \geq 5$ put $\varepsilon_k = \frac{1}{2k^2}$ and $f_k = g(\varepsilon_k)$. We will show that the family $\mathcal{F} = \{C_4\} \cup \{B_{k,f_k} : k \geq 5\}$ satisfies the requirement. Let $k \geq 5$, let $n \geq n_{18}(k, f_k)$ and set $G = B_{k,n/k}$. By Lemma 18, G is ε_k-far from being induced $\{C_4, B_{k,f_k}\}$-free. Since $C_4, B_{k,f_k} \in \mathcal{F}$, we get that G is ε_k-far from being induced $\mathcal{F}$-free.

We claim that for every $4 \leq \ell < k$, G is induced C_ℓ-free. Suppose, for the sake of contradiction, that $x_1, \dots, x_\ell, x_1$ is an induced ℓ-cycle in G. Let $V_1, \dots, V_k$ be the sides of $G = B_{k,n/k}$. If $|\{x_1, \dots, x_\ell\} \cap V_i| \leq 1$ for every $1 \leq i \leq k$ then $x_1, \dots, x_\ell$ are contained in an induced path, which is impossible. So there is some $1 \leq i \leq k$ for which $|\{x_1, \dots, x_\ell\} \cap V_i| \geq 2$. Suppose without loss of generality that $x_1, x_2 \in V_1$ (recall that $V_1, \dots, V_k$ are cliques). Then $x_3 \in V_2$ or $x_3 \in V_k$, and in either case x_1, x_2, x_3 span a triangle, a contradiction.

We conclude that the smallest $F \in \mathcal{F}$ which is an induced subgraph of G, is $F = B_{k,f_k}$. Thus, every induced subgraph of G on less than $v(B_{k,f_k}) = k \cdot g(\varepsilon_k)$ vertices is induced $\mathcal{F}$-free, completing the proof. ◀

References

1. N. Alon. Testing subgraphs in large graphs. *Random Struct. Alg.*, 21:359–370, 2002.
2. N. Alon, E. Fischer, M. Krivelevich, and M. Szegedy. Efficient testing of large graphs. *Combinatorica*, 20:451–476, 2000.
3. N. Alon, E. Fischer, and I. Newman. Testing of bipartite graph properties. *SIAM Journal on Computing*, 37:959–976, 2007.
4. N. Alon, E. Fischer, I. Newman, and A. Shapira. A combinatorial characterization of the testable graph properties: it's all about regularity. *SIAM Journal on Computing*, 39:143–167, 2009.
5. N. Alon and J. Fox. Easily testable graph properties. *Combin. Probab. Comput.*, 24:646–657, 2015.
6. N. Alon and A. Shapira. A characterization of easily testable induced subgraphs. *Combin. Probab.Comput.*, 15:791–805, 2006.
7. N. Alon and A. Shapira. A characterization of the (natural) graph properties testable with one-sided error. *SIAM Journal on Computing*, 37:1703–1727, 2008.
8. L. Avigad and O. Goldreich. Testing graph blow-up. In *Proc. of APPROX-RANDOM*, pages 389–399. Springer, 2011.
9. D. Conlon and J. Fox. Bounds for graph regularity and removal lemmas. *GAFA*, 22:1191–1256, 2012.
10. D. Conlon and J. Fox. Graph removal lemmas. *Surveys in Combinatorics*, pages 1–50, 2013.

11 P. Erdős. On extremal problems of graphs and generalized graphs. *Israel J. Math*, 2:183–190, 1964.

12 P. Erdős. On some problems in graph theory, combinatorial analysis and combinatorial number theory. *Graph Theory and Combinatorics (Cambridge, 1983), Academic Press, London*, pages 1–17, 1984.

13 J. Fox. A new proof of the graph removal lemma. *Ann. of Math.*, 174:561–579, 2011.

14 L. Gishboliner and A. Shapira. Removal lemmas with polynomial bounds. *Proc. of STOC*, pages 510–522, 2017.

15 O. Goldreich. Introduction to property testing, Forthcoming book, 2017.

16 O. Goldreich, S. Goldwasser, and D. Ron. Property testing and its connection to learning and approximation. *J. ACM*, 45:653–750, 1998.

17 O. Goldreich and D. Ron. On proximity-oblivious testing. *SIAM J. on Computing*, 40:534–566, 2011.

18 T. Gowers. Lower bounds of tower type for szemerédi's uniformity lemma. *GAFA*, 7:322–337, 1997.

19 A. Gyárfás, A. Hubenko, and J. Solymosi. Large cliques in c_4-free graphs. *Combinatorica*, 22:269–274, 2002.

20 L. Lovász. *Large networks and graph limits*, volume 60. American Mathematical Society Providence, 2012.

21 G. Moshkovitz and A. Shapira. A sparse regular aproximation lemma.

22 V. Rödl and R. Duke. On graphs with small subgraphs of large chromatic number. *Graphs and Combinatorics*, 1:91–96, 1985.

23 R. Rubinfeld and M. Sudan. Robust charachterization of polynomials with applications to program testing. *SIAM Journal on Computing*, 25:252–271, 1996.

24 I.Z. Ruzsa and E. Szemerédi. Triple systems with no six points carrying three triangles. *Combinatorics (Keszthely, 1976), Coll. Math. Soc. J. Bolyai*, 18:939–945, 1976.

25 E. Szemerédi. Regular partitions of graphs. *In:* Proc. Colloque Inter. CNRS*, J. C. Fournier, M. Las Vergnas and D. Sotteau, eds.*, pages 399–401, 1978.

Agnostic Learning by Refuting*

Pravesh K. Kothari[1] **and Roi Livni**[2]

1 **Princeton University and Institute of Advanced Study, Princeton, NJ, USA**
kothari@cs.princeton.edu
2 **Princeton University, Princeton, NJ, USA**
rlivni@cs.princeton.edu

Abstract

The sample complexity of learning a Boolean-valued function class is precisely characterized by its Rademacher complexity. This has little bearing, however, on the sample complexity of *efficient* agnostic learning.

We introduce *refutation complexity*, a natural computational analog of Rademacher complexity of a Boolean concept class and show that it exactly characterizes the sample complexity of *efficient* agnostic learning. Informally, refutation complexity of a class $\mathcal{C}$ is the minimum number of example-label pairs required to efficiently distinguish between the case that the labels correlate with the evaluation of some member of $\mathcal{C}$ (*structure*) and the case where the labels are i.i.d. Rademacher random variables (*noise*). The easy direction of this relationship was implicitly used in the recent framework for improper PAC learning lower bounds of Daniely and co-authors [6, 8, 10] via connections to the hardness of refuting random constraint satisfaction problems. Our work can be seen as making the relationship between agnostic learning and refutation implicit in their work into an explicit equivalence. In a recent, independent work, Salil Vadhan [25] discovered a similar relationship between refutation and PAC-learning in the realizable (i.e. noiseless) case.

1998 ACM Subject Classification I.2.6 Learning

Keywords and phrases learning theory, computational learning, Rademacher complexity

Digital Object Identifier 10.4230/LIPIcs.ITCS.2018.55

1 Introduction

Statistical complexity characterizes the information theoretic threshold for the amount of data required for any supervised learning task. However, the amount of data required for *efficient* learning, whenever it is possible, can be significantly different from the statistical complexity. For example, algorithms based on polynomial regression ([18, 19, 20]) guarantee efficient (improper, i.e. return a hypothesis not necessarily in the target class) learning while using data that is a polynomial factor larger than the statistical complexity. There is a systematic effort to study the trade-offs between computational and statistical complexity [4, 5] and a growing body of work has provided explicit examples [11, 7, 2] of natural settings where efficient learning provably requires data that is at least a polynomial factor larger than the statistical complexity under some plausible complexity theoretic assumptions.

In the light of the above work, we focus on obtaining a simple and useful characterization of the sample complexity of *efficient* supervised learning. There's a simple and elegant char-

* This research was supported by funding from Eric and Wendy Schmidt Fund for Strategic Innovation.

acterization of the statistical complexity of learning in terms of the Rademacher complexity [1]. In this note, we give a natural analog of Rademacher complexity that precisely characterizes the amount of data required for *efficient agnostic* (i.e. noisy, see Definition 2) learning.

For a class $\mathcal{C}$ of concepts on $\mathbb{R}^n$, any distribution $\mathcal{D}$ on $\mathbb{R}^n$, the *Rademacher Complexity* of $\mathcal{C}$, $\mathcal{R}_m(\mathcal{C})$ is the following quantity:

$$\mathcal{R}_m(\mathcal{C}) = \mathop{\mathbb{E}}_{\substack{x_i \sim_{i.i.d.} \mathcal{D} \\ 1 \leqslant i \leqslant m}} \left[\mathop{\mathbb{E}}_{\substack{\sigma_i \sim_{i.i.d.} \{\pm 1\} \\ 1 \leqslant i \leqslant m}} \left[\frac{1}{m} \sup_{c \in \mathcal{C}} \sum_{i=1}^{m} \sigma_i c(x_i) \right] \right]. \tag{1.1}$$

Classical results [3] establish that $\mathcal{R}_m(\mathcal{C}) = \varepsilon$ if and only if there's an algorithm to learn $\mathcal{C}$ over $\mathcal{D}$ with error at most ε with $\Theta(m)$ samples, thus characterizing the sample-complexity of ε-error agnostic learning.

In this note, we propose a natural computational analog of Rademacher complexity, called as the *Refutation complexity* and show that it exactly determines the sample complexity of efficient agnostic learning. Given random labeled examples $\{(x_i, y_i)\}_{i \leqslant m}$ where x_is are chosen i.i.d. according to $\mathcal{D}$, we define the problem of *refutation* as the task of distinguishing between the following two cases:

(a) Structure: $\{(x_i, y_i)\}_{i \leqslant m}$ are i.i.d. from some distribution $\mathcal{D}'$ with marginal on x_is being $\mathcal{D}$ and $\mathbb{E}_{(x,y) \sim \mathcal{D}'}[c(x)y] = \Omega(1)$. That is, the given example-label pairs come from a distribution that correlates with some $c \in \mathcal{C}$, and

(b) Noise: y_is are uniform and independent Rademacher random variables.

We define refutation complexity of $\mathcal{C}$ with respect to the distribution $\mathcal{D}$ at a running time of $T(n)$ as the smallest m for which there's a $T(n)$-time test for distinguishing between structure and noise cases above.

To motivate this definition, observe that we can interpret the statistical complexity (via the connection to Rademacher complexity outlined above) of $\mathcal{C}$ over $\mathcal{D}$ as the smallest m for which no concept in $\mathcal{C}$ correlates with purely random noise (the i.i.d. draws from $\{\pm 1\}$.) Thus, if the Rademacher complexity of $\mathcal{C}$ on $\mathcal{D}$ with m samples is small enough, then, given random labeled examples $\{(x_i, y_i)\}_{i \leqslant m}$, we can (via an inefficient procedure) distinguish between the above two cases by computing the largest correlation of any $c \in \mathcal{C}$ when evaluated at x_is with the y_is. Thus, we can equivalently define statistical complexity as the smallest m for which the above structure vs noise test succeeds. Thus, refutation complexity can be seen as a computational analog of Rademacher complexity.

The main result of this note is the following theorem:

▶ **Theorem 1** (Refutation Complexity = Agnostic Learning Complexity, Informal)**.** *$\mathcal{C}$ has an efficient agnostic learning algorithm over a distribution $\mathcal{D}$ with m samples if and only if the refutation complexity of $\mathcal{C}$ at some polynomial running time is at most $O(m)$.*

1.1 Comparison with [25]

In a recent, independent work, Vadhan [25] used similar arguments to establish a similar equivalence to Theorem 1 between *distribution independent PAC learning* in the realizable case (i.e. when the labels perfectly correlate with some concept in the target class) and a slightly different notion of refutation. In this notion, the refutation algorithm is required to distinguish the case that the sample that realizable (i.e., the labels agree with some concept

[1] The related notion of VC Dimension of $\mathcal{C}$ characterizes the data required to learn $\mathcal{C}$ over *worst-case* distributions.

from the class) from the case that the labels in the sample are i.i.d. Rademacher random variables.

Since agnostic learning is provably different from realizable PAC learning in general, the notions of refutation that characterize the complexity of learning in the two models have to be necessarily different. Another interesting point of difference is that our equivalence is *distribution-specific* and thus slightly more fine-grained in that it allows relating learnability on a given distribution to refutation on the same distribution. In contrast, Vadhan's characterization holds for distribution independent PAC learning. This difference arises entirely due to the the difference in the black-box boosting algorithms one can use in PAC vs agnostic settings[2]: in the PAC learning case, the boosting algorithms modify the distribution of examples over the course of the execution and thus the characterization holds only in a distribution independent setting. In the agnostic setting, there are distribution specific boosting algorithms (such as that of [17, 14]) that work by changing only the distributions of the labels while keeping the distribution of the example points unchanged. It is an interesting direction to investigate notions of refutation that allow *distribution-specific* characterization of PAC learning in realizable case.

It's interesting to note how slight changes to in the formulation of the refutation problem changes the model of learning that it characterizes.

1.2 Discussion

Proper vs Improper Learning and the Framework of [9]

The agnostic learning algorithm we obtain using a refutation algorithm is *improper* - that is, it doesn't necessarily produce a hypothesis from the class $\mathcal{C}$. This is not accidental - it's well known that the flexibility of *improper* learning allows circumventing computational hardness results that afflict *proper* learning. A simple example is the class of 3-term DNF formulas in n variables: unless RP = NP, there's no polynomial time *proper* learning algorithm for this class [21], however, there's a simple poly$(n, 1/\varepsilon)$-time *improper* learning algorithm for it (for a discussion see, [24]). On the flip side, the power of *improper* learning makes the task of proving lower bound against such algorithms harder. The equivalence between refutation and agnostic learning holds for all (and thus, also improper) learning algorithms and thus can serve as a useful handle in understanding the complexity of improper learning.

Indeed this connection and in particular, the implication that learning implies refutation is implicit in the influential work of Daniely and co-authors [6, 10, 8] who showed (in the language of this paper) that a refutation algorithm for the concept classes of halfspaces and DNF formulas can be used to refute certain random constraint satisfaction problems [1, 12]. These works used such a reduction along with standard hardness assumptions for refuting random CSPs to obtain the first hardness results for improper PAC learning for the above classes.

Our equivalence establishes the converse of the connection in these works and makes the connection between refutation and agnostic learning explicit. While a priori, it might appear that refutation (which asks for distinguishing between a pure noise in the labels from a correlated set of labels) is easier than agnostic learning, this work shows that any lower bound on (improper) learning has to necessarily be a lower bound for an associated refutation problem. Thus, to an extent, it shows that the above framework for improper agnostic learning lower bounds is essentially complete.

[2] We thank Salil Vadhan for pointing this out to us.

Connections to Boosting/Property Testing

It is also illuminating to view the equivalence we show as saying that an oracle for refutation is sufficient for agnostic learning. This naturally leads to the question of what kind of oracle access to $\mathcal{C}$ is sufficient for (agnostic) learning. We discuss two natural oracles here: a weak-learning oracle and a property-testing oracle.

Known boosting (see [15, 23], [17, 14]) algorithms imply that a *weak-learning oracle* is sufficient for agnostically learning of $\mathcal{C}$. A weak learning oracle takes random example-label pairs and returns a hypothesis whose correlation with the labels from the input distribution is at least an inverse polynomial fraction of the correlation of the best-fitting hypothesis from $\mathcal{C}$. In learning literature, this is sometimes referred to as a *weak-optimization* oracle for $\mathcal{C}$ - in that, it gives a inverse polynomial (potentially improperly) approximation to the correlation of the best fitting hypothesis from $\mathcal{C}$. It is not hard to see that such an oracle is enough to solve the refutation problem and thus is a potentially stronger access to $\mathcal{C}$ than the refutation algorithm.

Our result implies that an much weaker algorithm is enough to get an agnostic learning algorithm - the refutation oracle doesn't return any hypothesis, it "merely" distinguishes between the case that the labels are completely random and independent of the examples from the case that the labels come from some distribution that correlates with some concept in $\mathcal{C}$.

It is also instructive to compare a refutation oracle (or a "structure" vs "noise" tester) for $\mathcal{C}$ with a "property-tester" for $\mathcal{C}$. An α-approximate property-testing algorithm for $\mathcal{C}$ uses random example-label pairs [3] from some distribution and accepts if the labels achieve a correlation of at least α with $\mathcal{C}$ and rejects if tevery $c \in \mathcal{C}$ has a correlation of at most $\alpha - \varepsilon$ with the labels. We can interpret a property tester, thus, as a variant of the refutation oracle that must treat a distribution on example-label pairs that has a correlation of at most $\alpha - \varepsilon$ with every $c \in \mathcal{C}$ as "noise." In particular, the notion of what is "unstructured/noise" for a property tester is more stringent compared to a refutation algorithm. Indeed, this is not surprising: while testing is known to be no harder than *proper* learning, it can be harder than *improper* learning for some concept classes, once again illustrating the difference between proper and improper learning [16].

Using Refutation to get Learning Algorithms

It will be extremely interesting to understand if the equivalence between refutation and learning allows an application in the direction opposite to the one employed in the work of Daniely and co-authors and get new algorithms for agnostic learning. This is perhaps not too optimistic. The works of Daniely and co-authors establish a natural connection between the refutation problem for a concept class and refuting random CSPs. There are known algorithms for refuting random CSPs (see for e.g. [22, 13, 1]) that use techniques that appear different from the usual tool-kit in agnostic learning (for e.g. the use of semi-definite programming) that might prove useful in obtaining new agnostic learning algorithms by building the required refutation algorithms.

[3] Property testers are usually defined with $\alpha = 1$ and are in general also allowed to use membership queries. We use a definition that is similar in spirit but is more relevant for the comparison here.

1.3 Proof Overview

It is easy to see that efficient learning implies efficient refutation. For the other direction, we give an explicit, efficient algorithm that invokes the refutation algorithm a small number of times to get an agnostic learner for the class $\mathcal{C}$. This algorithm works in two steps - in the first step, it uses a refutation algorithm to come up with a *weak-agnostic* learner: i.e. a hypothesis that achieves a correlation with the labels that is some tiny fraction of the correlation of the best hypothesis from $\mathcal{C}$. In the second step, it combined an off-the-shelf boosting algorithm with the weak learner above to get an agnostic learner with small error.

The key idea in the transformation of a refutation algorithm into a weak-learner is to view the black-box refutation algorithm as a "code" for computing a function by manipulating the example-label pairs that it takes as input. A simple hybrid argument then shows that there's a small list of hypotheses generated by manipulating the inputs to the refutation algorithm that contains a good weak learner. We can find the best weak learner from the list by evaluating the error of each of the hypotheses in the list over a fresh batch of samples from the underlying distribution.

1.4 Preliminaries

We use $\mathcal{U}_m$ to denote the uniform distribution over $\{\pm 1\}^m$ for any $m \in \mathbb{N}$. We define agnostic learning here.

▶ **Definition 2** (Agnostic Learning with respect to a distribution $\mathcal{D}$)**.** Let $\mathcal{C}$ be a class of Boolean concepts $\mathcal{C} \subseteq \{f : \{\pm 1\}^n \to \{\pm 1\}\}$. $\mathcal{C}$ is said to be ε-agnostically learnable in time $T(n, 1/\varepsilon)$ and samples $S(n, 1/\varepsilon)$ if there's an algorithm $\mathcal{A}$ running in time $T(n, 1/\varepsilon)$ that takes $S(n, 1/\varepsilon)$ random labeled examples $\{(x_i, y_i) \mid 1 \leqslant i \leqslant m\}$ where (x_i, y_i)s are i.i.d. from $\mathcal{D}'$, such that the marginal on x_i is $\mathcal{D}$ and outputs with probability at least $3/4$, a hypothesis $h : \{\pm 1\}^n \to \{\pm 1\}$ such that $\mathbb{E}_{(x,y)\sim\mathcal{D}'}[\mathbf{1}[h(x) \neq y]] \leqslant \inf_{c\in\mathcal{C}} \mathbb{E}_{(x,y)\sim\mathcal{D}'}[\mathbf{1}[c(x) \neq y]] + \varepsilon$.

2 Refutation Complexity

In this section, we define refutation complexity of a class of hypothesis with respect to a distribution $\mathcal{D}$.

▶ **Definition 3** (Refutation Algorithm for Distribution $\mathcal{D}$)**.** Let $\mathcal{C} \subseteq \{f : \mathbb{R}^n \to \{\pm 1\}\}$ be a class of Boolean concepts. Let $\mathcal{D}$ be a distribution on $\mathbb{R}^n$.

A δ-*refutation* algorithm $\mathcal{A}$ for $\mathcal{C}$ on $\mathcal{D}$ with $m = m(n)$ samples is a (possibly randomized) algorithm that takes input an m-tuple of points $\{x_1, x_2, \ldots, x_m\} \subseteq \{\pm 1\}^n$ and an m-tuple of labels $(\sigma_1, \sigma_2, \ldots, \sigma_m) \in \{\pm 1\}^m$ and outputs either noise or structure with the following guarantees:

1. **Completeness:** If $\{(x_i, \sigma_i)\}_{i\leqslant m}$ are i.i.d. from a distribution $\mathcal{D}'$ on $\mathbb{R}^n \otimes \{\pm 1\}$ such that the marginal on $\mathbb{R}^n$ equals $\mathcal{D}$ and $\sup_{c\in\mathcal{C}} \mathbb{E}_{(x,\sigma)\sim\mathcal{D}'}[c(x)\sigma] \geqslant \delta$, then,

$$\mathop{\mathbb{P}}_{\substack{\{(x_i,y_i)\}_{i\leqslant m}\sim_{i.i.d.}\mathcal{D}' \\ \text{internal randomness of } \mathcal{A}}} [\text{ output} = \textsf{structure}] \geqslant 2/3.$$

2. **Soundness:**

$$\mathop{\mathbb{P}}_{\substack{(\sigma_1,\sigma_2,\ldots,\sigma_m)\sim\mathcal{U}_m \\ x_1,x_2,\ldots,x_m\sim\mathcal{D} \\ \text{internal randomness of } \mathcal{A}}} \mathbb{P}[\text{ output} = \textsf{noise}] \geqslant 2/3.$$

▶ **Definition 4** (δ-Refutation Complexity)**.** Let $\mathcal{C} \subseteq \{f : \{\pm 1\}^n \to \{\pm 1\}\}$ be a class of Boolean concepts. Let $\mathcal{D}$ be a distribution on $\{\pm 1\}^n$.

The *δ-refutation complexity* of $\mathcal{C}$ on a distribution $\mathcal{D}$ with running time $T(n)$ denoted by $\mathcal{R}_{T(n),\delta}(\mathcal{C})$, is the smallest $m = m(n, \delta)$ such that there exists a δ-refutation algorithm for $\mathcal{C}$ on $\mathcal{D}$ running in time $T(n)$ and m-samples. When $T(n)$ is not stated explicitly, we assume $T(n) = \text{poly}(n)$ for some fixed polynomial in n.

▶ Remark. Observe that the refutation complexity, just as Rademacher complexity is distribution dependent. Further, for $T(n) = \infty$, δ-refutation complexity degenerates into Rademacher complexity. At non-trivially bounded running times (of special interest, of course, is polynomial time algorithms), refutation complexity captures the sample complexity of *efficient* agnostic, improper learning $\mathcal{C}$ over $\mathcal{D}$ as we show next and thus can be much larger than the Rademacher complexity.

3 Learning vs Refutation Complexity

In this section, we establish the equivalence between agnostic learning a class $\mathcal{C}$ over a given distribution $\mathcal{D}$ and the refutation problem with respect to the distribution $\mathcal{D}$ for the concept class $\mathcal{C}$.

We begin by showing the Learning implies Refutation, which is the easy direction.

▶ **Lemma 5** (Learning implies Refutation)**.** *Suppose $\mathcal{C}$ is ε-agnostically learnable in time $T(n, \varepsilon)$ and samples $S(n, \varepsilon)$ over the distribution $\mathcal{D}$. Then, the refutation complexity of $\mathcal{C}$ with respect to the distribution $\mathcal{D}$ at the running time $T(n, \delta/4)$ is at most $2S(n, \delta/4)+128/\delta^2$.*

Proof. Let $m = S(n, \delta/4) + 64/\delta^2$.

The δ-refutation algorithm gets input $x_1, x_2, \ldots, x_{2m}$ and $\sigma_1, \sigma_2, \ldots, \sigma_{2m}$. It runs the ε-agnostic learner on examples $\{(x_i, \sigma_i)\}_{i=1}^m$ for $\varepsilon = \delta/4$ and obtains a hypothesis h. Let $\text{cor}_h = \frac{1}{m}\sum_{i=m+1}^{2m} \sigma_i \cdot h(x_i)$. If $\text{cor}_h \geqslant \delta/2$, output structure otherwise output noise.

We now analyze the completeness and the soundness properties of this algorithm.

First, suppose $\{(x_i, \sigma_i)\}_{i \leqslant 2m}$ were i.i.d. according to some $\mathcal{D}'$ such that the marginal on $\mathbb{R}^n$ equals $\mathcal{D}$. Let $\text{cor}_f(\mathcal{D}') = \mathbb{E}_{(x,y)\sim\mathcal{D}'}[f(x)y]$. Then, with probability $2/3$ over the draw of the sample, the agnostic learner produces a hypothesis h such that $\text{cor}_h \geqslant \text{cor}_h(\mathcal{D}') - \varepsilon \geqslant \text{cor}_c(\mathcal{D}') - 2\varepsilon$ for every $c \in \mathcal{C}$. Thus, if $\text{cor}_c(\mathcal{D}') \geqslant \delta$, then, $\text{cor}_h \geqslant \delta - \varepsilon/2 \geqslant \delta/2$. Thus, in this case, the algorithm above outputs structure as desired.

Now suppose σ_is are i.i.d. Rademacher and independent of x_is. Then, since $\sigma_{m+1}, \ldots, \sim_{2m}$ are independent of $\sigma_1, \ldots, \sigma_m$, $\text{cor}_h \leqslant \frac{4}{\sqrt{m}} < \delta/2$ using that $m > 64/\delta^2$. ◀

▶ **Lemma 6** (Learning by Refutation)**.** *Suppose that the δ-refutation complexity of a class of Boolean concepts $\mathcal{C}$ with respect to a distribution $\mathcal{D}$ at a running time $T(n)$ is $m = \mathcal{R}_{T(n),\delta}(\mathcal{C})$. Then, there's an algorithm that runs in time $T(n)\frac{m^2}{\varepsilon^2}$ and uses $O(\frac{m^3}{\varepsilon^2})$ samples to $(\delta + \varepsilon)$-agnostically learn $\mathcal{C}$ on $\mathcal{D}$.*

The proof is in two steps. In the first step, we show that the refutation algorithm yields a weak agnostic learner for $\mathcal{C}$ with respect to the distribution $\mathcal{D}$. In the second step, we use the distribution specific agnostic boosting algorithm (see [17]) to boost the accuracy of the weak learner to obtain an agnostic learner. We start by defining a weak-agnostic learner :

▶ **Definition 7** (Weak Agnostic Learner)**.** An (γ, α)-weak agnostic learner for a Boolean concept class $\mathcal{C}$ over a distribution $\mathcal{D}$ is an algorithm that takes input random examples from a distribution $\mathcal{D}'$ on example-label pairs (x, y) such that the marginal on x is $\mathcal{D}$ such

that with probability at least 3/4 over its random input outputs a (randomized) hypothesis $h : \{\pm 1\}^n \to \{\pm 1\}$ such that $\mathbb{E}_{(x,y)\sim\mathcal{D}'}[y \cdot h(x)] \geqslant \gamma(\sup_{c\in\mathcal{C}} \mathbb{E}_{(x,y)\sim\mathcal{D}'}[y \cdot c(x)]) - \alpha$.

▶ **Lemma 8** (Refutation to Weak Agnostic Learner)**.** *Suppose that the δ-refutation complexity of a class of Boolean concepts $\mathcal{C}$ with respect to a distribution $\mathcal{D}$ at a running time $T(n)$ is $m = \mathcal{R}_{T(n),\delta}(\mathcal{C})$. Then, there's an (γ, α)-weak agnostic learner for $\mathcal{C}$ on distribution $\mathcal{D}$ that runs in time $T(n)$ and samples $m(n)$ where $\alpha = \delta \cdot \gamma$, $\gamma = \frac{2}{3m}$.*

We describe a natural class of candidates for a weak learner that come out of running the refutation algorithm on appropriately chosen hybrids of the distribution $\mathcal{D}'$ and $\mathcal{D} \times \mathcal{U}_1$. We begin by defining a class of $2(m+2)$ different functions denoted by $W_{i,b} : \{\pm 1\}^n \to \{0, 1\}$ for $0 \leqslant i \leqslant m+1$ and $b \in \pm 1$ produced by taking these hybrids. Our weak learners will be a simple transformation of this class.

Algorithm 1 Hybrid Functions $W_{i,b}$

Input: $x \in \mathbb{R}^n$, $b \in \{\pm 1\}$.
Output: $W_{i,b}(x) = z \in \{\pm 1\}$.
Operation:

1. Draw $(x_1, \sigma_1), \ldots, (x_{i-1}, \sigma_{i-1})$ i.i.d. from $\mathcal{D} \times \mathcal{U}_1$.
   Draw $(x_{i+1}, y_{i+1}), (x_{i+2}, y_{i+2}) \ldots, (x_m, y_m)$ i.i.d. from $\mathcal{D}'$.
2. Run the δ-refutation algorithm on input
   $(x_1, \sigma_1), (x_2, \sigma_2), \ldots, (x_{i-1}, \sigma_{i-1}), (x, b), (x_{i+1}, y_{i+1}), \ldots, (x_m, y_m)$.
3. Let $W_{i,b} = 1$ if the refutation algorithm returns structure and 0 otherwise.

We make some simple observations about $W_{i,b}$ that will come handy in the argument below.

Observe that $W_{m+1,b}$ is the function that evaluates to 1 if the output of the refutation algorithm on examples drawn from $\mathcal{D}$ and labels i.i.d Rademacher variables is structure. On the other hand, $W_{0,b}$ is the function obtained when the refutation algorithm is run on example-label pairs from $\mathcal{D}$. Finally, observe that

$$\mathop{\mathbb{E}}_{b\sim\mathcal{U}_1} \mathbb{E}[W_{i,b}(x)] = \mathop{\mathbb{E}}_{(x,y)\sim\mathcal{D}} \mathbb{E}[W_{i+1,y}(x)] \tag{3.1}$$

Here, the inside expectation is over all the random choices within the procedure for computing $W_{i,b}$s above. We can now present our candidate weak learners.

Candidate Weak Learners

For every $0 \leqslant i \leqslant m+1$, let $h_i(x) = W_{i,+1}(x) - W_{i,-1}(x)$.

Proof of Lemma 8. Our weak learning algorithm is given access to random labeled examples from a distribution $\mathcal{D}'$ on $\mathbb{R}^n \otimes \{\pm 1\}$. The weak learner will draw a sample from $\mathcal{D}'$ of size $O(\log m)$ from $\mathcal{D}'$ and chooses the h_i that has the maximum correlation with the labels. Observe that with $O(\log(m))$ samples, the correlations of h_i on $\mathcal{D}'$ will be faithfully preserved with 2/3 probability. Thus, to complete the proof, we only need to argue that one of the h_is is always an (α, γ)-weak learner.

To show this, we must argue that there exists an $0 \leqslant i \leqslant m+1$ such that:

$$\mathop{\mathbb{E}}_{(x,y)\sim\mathcal{D}'}[y \cdot h_i(x)] \geqslant \frac{2}{3m} \sup_{c\in\mathcal{C}} \mathop{\mathbb{E}}_{(x,y)\sim\mathcal{D}'}[c(x) \cdot y] - \frac{2}{3m}\delta.$$

Observe that the guarantees of the weak learner are trivial if $\sup_{c\in\mathcal{C}} \mathbb{E}_{(x,y)\sim\mathcal{D}'}[y\cdot c(x)] < \delta$. Thus assume that $\sup_{c\in\mathcal{C}} \mathbb{E}_{(x,y)\sim\mathcal{D}'}[y \cdot c(x)] > \delta$. In this case, we will show that $\mathbb{E}_{(x,y)\sim\mathcal{D}'}[h_i(x)y] \geqslant \frac{2}{3m} \geqslant \frac{2}{3m}\sup_{c\in\mathcal{C}} \mathbb{E}_{(x,y)\sim\mathcal{D}'}[c(x)\cdot y] - \frac{2}{3m}\delta$.

Now, observe that over the randomness of both the refutation algorithm and over the draw of i.i.d. sample from $\mathcal{D}'$ of size $m = S(n)$, $\mathbb{E}[W_{0,b}(x)] \geqslant 2/3$ and $\mathbb{E}[W_{m+1,b}(x)] \leqslant 1/3$ for any b. Thus,

$$\sum_{i=0}^{m} \mathbb{E}[W_{i,y}(x) - W_{i+1,y}(x)] \geqslant 1/3,$$

where the expectation is over the randomness in the draw $(x, y) \sim \mathcal{D}'$ and over the randomness in $W_{i,y}$ for $0 \leqslant i \leqslant m+1$.

Thus, there must exist an i such that $\mathbb{E}[W_{i,y}(x) - W_{i+1,y}(x)] > 1/3m$. Observe that by construction

$$\begin{aligned} W_{i,y}(x) = \frac{y+1}{2}\cdot W_{i,1}(x) - \frac{y-1}{2}W_{i,-1}(x) &= y\cdot\frac{W_{i,1}(x) - W_{i,-1}(x)}{2} + \frac{1}{2}(W_{i,1}(x) + W_{i,-1}(x)) \\ &= \frac{1}{2}y\cdot h_i(x) + \frac{1}{2}(W_{i,1}(x) + W_{i,-1}(x)). \quad (3.2) \end{aligned}$$

Next, observe that by (3.1), $\mathbb{E}[\frac{1}{2}(W_{i,1}(x)+W_{i,-1}(x))] = \mathbb{E}[W_{i+1,y}(x)]$. Taking expectations on both sides of (3.2) and rearranging, we have: $\mathbb{E}[y\cdot h_i(x)] \geqslant \frac{2}{3m}$.

This establishes that for $\gamma = \frac{2}{3m}$ and $\alpha = \delta\cdot\gamma$ our algorithm returns (α,γ)-weak agnostic learner as desired. ◀

We can now use boosting to get a strong agnostic learner for $\mathcal{C}$ over $\mathcal{D}$ by using the weak learning algorithm along with a boosting algorithm. Specifically, we will use the result of Kalai and Kanade [17] (see also [14]) who showed the following agnostic boosting algorithm that takes a (γ,α)-weak learner and outputs a hypothesis whose error is competitive within α with respect to the best fitting hypothesis from the class $\mathcal{C}$.

▶ **Fact 9** (Agnostic Boosting [17]). *Let $\mathcal{C}$ be a class of Boolean concepts. Let $\mathcal{D}$ be a distribution on $\{\pm1\}^n$ and $\varepsilon > 0$.*

There's an algorithm that takes random labeled examples from a distribution $\mathcal{D}'$ on example-label pairs (x, y) such that the marginal on x is $\mathcal{D}$, invokes a (γ,α)-weak learner for $\mathcal{C}$ $O(\frac{1}{\gamma^2\varepsilon^2})$ times and outputs a hypothesis $h : \{\pm1\}^n \to \{\pm1\}$ such that

$$\mathop{\mathbb{E}}_{(x,y)\sim\mathcal{D}'}[\mathbf{1}[h(x)\neq y]] \leqslant \inf_{c\in\mathcal{C}} \mathop{\mathbb{E}}_{(x,y)\sim\mathcal{D}'}[\mathbf{1}[c(x)\neq y]] + \alpha/\gamma + \varepsilon.$$

The algorithm needs $S(n)\cdot O(\frac{1}{\gamma^2\varepsilon^2})$ samples and runs in time $T(n)\cdot O(\frac{1}{\gamma^2\varepsilon^2})$ where $S(n)$ and $T(n)$ are the sample complexity and the running time respectively of the (γ,α)-weak agnostic learner.

We get Lemma 6 as an immediate corollary of Fact 9 and Lemma 8.

Acknowledgements. We thank Salil Vadhan for sharing an early version of [25] with us and illuminating follow-up discussions. P.K. thanks Avi Wigderson for many useful comments and suggestions about this work and David Steurer for helpful discussions on related problems.

References

1. Sarah R. Allen, Ryan O'Donnell, and David Witmer. How to refute a random CSP. In *2015 IEEE 56th Annual Symposium on Foundations of Computer Science—FOCS 2015*, pages 689–708. IEEE Computer Soc., Los Alamitos, CA, 2015.
2. Boaz Barak and Ankur Moitra. Tensor prediction, rademacher complexity and random 3-xor. *CoRR*, abs/1501.06521, 2015.
3. Peter L. Bartlett and Shahar Mendelson. Rademacher and gaussian complexities: Risk bounds and structural results. *Journal of Machine Learning Research*, 3:463–482, 2002. URL: `http://www.jmlr.org/papers/v3/bartlett02a.html`.
4. Quentin Berthet and Philippe Rigollet. Computational lower bounds for sparse PCA. *CoRR*, abs/1304.0828, 2013.
5. Venkat Chandrasekaran and Michael I. Jordan. Computational and statistical tradeoffs via convex relaxation. *Proceedings of the National Academy of Sciences*, 110(13):E1181–E1190, 2013. `doi:10.1073/pnas.1302293110`.
6. Amit Daniely. Complexity theoretic limitations on learning halfspaces. 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 105–117. ACM, 2016. `doi:10.1145/2897518.2897520`.
7. Amit Daniely, Nati Linial, and Shai Shalev-Shwartz. More data speeds up training time in learning halfspaces over sparse vectors. In *NIPS*, pages 145–153, 2013.
8. Amit Daniely, Nati Linial, and Shai Shalev-Shwartz. From average case complexity to improper learning complexity. In *STOC*, pages 441–448. ACM, 2014.
9. Amit Daniely, Nati Linial, and Shai Shalev-Shwartz. From average case complexity to improper learning complexity. In *Proceedings of the forty-sixth annual ACM symposium on Theory of computing*, pages 441–448. ACM, 2014.
10. Amit Daniely and Shai Shalev-Shwartz. Complexity theoretic limitations on learning dnf's. In *COLT*, pages 815–830, 2016.
11. Scott E. Decatur, Oded Goldreich, and Dana Ron. Computational sample complexity. *SIAM J. Comput.*, 29(3):854–879, 1999.
12. Uriel Feige. Relations between average case complexity and approximation complexity. In John H. Reif, editor, *Proceedings on 34th Annual ACM Symposium on Theory of Computing, May 19-21, 2002, Montréal, Québec, Canada*, pages 534–543. ACM, 2002. `doi:10.1145/509907.509985`.
13. Uriel Feige. Refuting smoothed 3cnf formulas. In *FOCS*, pages 407–417. IEEE Computer Society, 2007.
14. Vitaly Feldman. Distribution-specific agnostic boosting. In Andrew Chi-Chih Yao, editor, *Innovations in Computer Science - ICS 2010, Tsinghua University, Beijing, China, January 5-7, 2010. Proceedings*, pages 241–250. Tsinghua University Press, 2010. URL: `http://conference.itcs.tsinghua.edu.cn/ICS2010/content/papers/20.html`.
15. Yoav Freund and Robert E Schapire. A desicion-theoretic generalization of on-line learning and an application to boosting. In *European conference on computational learning theory*, pages 23–37. Springer, 1995.
16. Oded Goldreich, Shafi Goldwasser, and Dana Ron. Property testing and its connection to learning and approximation. *J. ACM*, 45(4):653–750, 1998.
17. Adam Kalai and Varun Kanade. Potential-based agnostic boosting. In Yoshua Bengio, Dale Schuurmans, John D. Lafferty, Christopher K. I. Williams, and Aron Culotta, editors, *Advances in Neural Information Processing Systems 22: 23rd Annual Conference on Neural Information Processing Systems 2009. Proceedings of a meeting held 7-10 December 2009, Vancouver, British Columbia, Canada.*, pages 880–888. Curran Associates, Inc., 2009. URL: `http://papers.nips.cc/paper/3676-potential-based-agnostic-boosting`.

18 Adam Tauman Kalai, Adam R. Klivans, Yishay Mansour, and Rocco A. Servedio. Agnostically learning halfspaces. *SIAM J. Comput.*, 37(6):1777–1805, 2008. doi:10.1137/060649057.

19 Daniel M. Kane, Adam R. Klivans, and Raghu Meka. Learning halfspaces under log-concave densities: Polynomial approximations and moment matching. In Shai Shalev-Shwartz and Ingo Steinwart, editors, *COLT 2013 - The 26th Annual Conference on Learning Theory, June 12-14, 2013, Princeton University, NJ, USA*, volume 30 of *JMLR Workshop and Conference Proceedings*, pages 522–545. JMLR.org, 2013. URL: http://jmlr.org/proceedings/papers/v30/Kane13.html.

20 Adam R. Klivans, Ryan O'Donnell, and Rocco A. Servedio. Learning geometric concepts via gaussian surface area. In *49th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2008, October 25-28, 2008, Philadelphia, PA, USA*, pages 541–550. IEEE Computer Society, 2008. doi:10.1109/FOCS.2008.64.

21 Leonard Pitt and Leslie G. Valiant. Computational limitations on learning from examples. *J. ACM*, 35(4):965–984, 1988.

22 Prasad Raghavendra, Satish Rao, and Tselil Schramm. Strongly refuting random csps below the spectral threshold. *CoRR*, abs/1605.00058, 2016.

23 Robert E. Schapire. The strength of weak learnability. *Machine Learning*, 5:197–227, 1990. doi:10.1007/BF00116037.

24 Shai Shalev-Shwartz, Ohad Shamir, and Eran Tromer. Using more data to speed-up training time. In Neil D. Lawrence and Mark A. Girolami, editors, *Proceedings of the Fifteenth International Conference on Artificial Intelligence and Statistics, AISTATS 2012, La Palma, Canary Islands, April 21-23, 2012*, volume 22 of *JMLR Proceedings*, pages 1019–1027. JMLR.org, 2012. URL: http://jmlr.csail.mit.edu/proceedings/papers/v22/shalev-shwartz12.html.

25 Salil Vadhan. On learning vs. refutation. In *Conference on Learning Theory*, pages 1835–1848, 2017.

A Homological Theory of Functions: Nonuniform Boolean Complexity Separation and VC Dimension Bound Via Algebraic Topology, and a Homological Farkas Lemma*

Greg Yang†

MSR AI, Microsoft Research, Redmond, WA, USA
gregyang@microsoft.com

Abstract

In computational complexity, a complexity class is given by a set of problems or functions, and a basic challenge is to show separations of complexity classes $\mathtt{A} \neq \mathtt{B}$ especially when $\mathtt{A}$ is known to be a subset of $\mathtt{B}$. In this paper we introduce a homological theory of functions that can be used to establish complexity separations, while also providing other interesting consequences. We propose to associate a topological space $\mathcal{S}_{\mathtt{A}}$ to each class of functions $\mathtt{A}$, such that, to separate complexity classes $\mathtt{A} \subseteq \mathtt{B}'$, it suffices to observe a change in "the number of holes", i.e. homology, in $\mathcal{S}_{\mathtt{A}}$ as a subclass $\mathtt{B} \subseteq \mathtt{B}'$ is added to $\mathtt{A}$. In other words, if the homologies of $\mathcal{S}_{\mathtt{A}}$ and $\mathcal{S}_{\mathtt{A}\cup\mathtt{B}}$ are different, then $\mathtt{A} \neq \mathtt{B}'$. We develop the underlying theory of functions based on homological commutative algebra and Stanley-Reisner theory, and prove a "maximal principle" for polynomial threshold functions that is used to recover Aspnes, Beigel, Furst, and Rudich's characterization of the polynomial threshold degree of symmetric functions. A surprising coincidence is demonstrated, where, roughly speaking, the maximal dimension of "holes" in $\mathcal{S}_{\mathtt{A}}$ upper bounds the VC dimension of $\mathtt{A}$, with equality for common computational cases such as the class of polynomial threshold functions or the class of linear functionals over $\mathbb{F}_2$, or common algebraic cases such as when the Stanley-Reisner ring of $\mathcal{S}_{\mathtt{A}}$ is Cohen-Macaulay. As another interesting application of our theory, we prove a result that a priori has nothing to do with complexity separation: it characterizes when a vector subspace intersects the positive cone, in terms of homological conditions. By analogy to Farkas' result doing the same with *linear conditions*, we call our theorem the Homological Farkas Lemma.

1998 ACM Subject Classification F.1.3 Complexity Measures and Classes, I.2.6 Learning, G.2.1 Combinatorics, G.2.m Miscellaneous

Keywords and phrases Homology, Stanley-Reisner, Cellular resolution, VC dimension, Homological Farkas

Digital Object Identifier 10.4230/LIPIcs.ITCS.2018.56

1 Introduction

Computational complexity is one of the most important areas of theoretical computer science, within which complexity lower bounds is the aspect that is least understood. Basic questions such as P vs NP, P vs BPP, L vs P, and so on still remain open. In this work, we propose the following method for proving nonuniform Boolean lower bounds. For every class $\mathtt{C}$ of

* A full version of this paper is available at [19], https://arxiv.org/abs/1701.02302
† Work done while at Harvard University

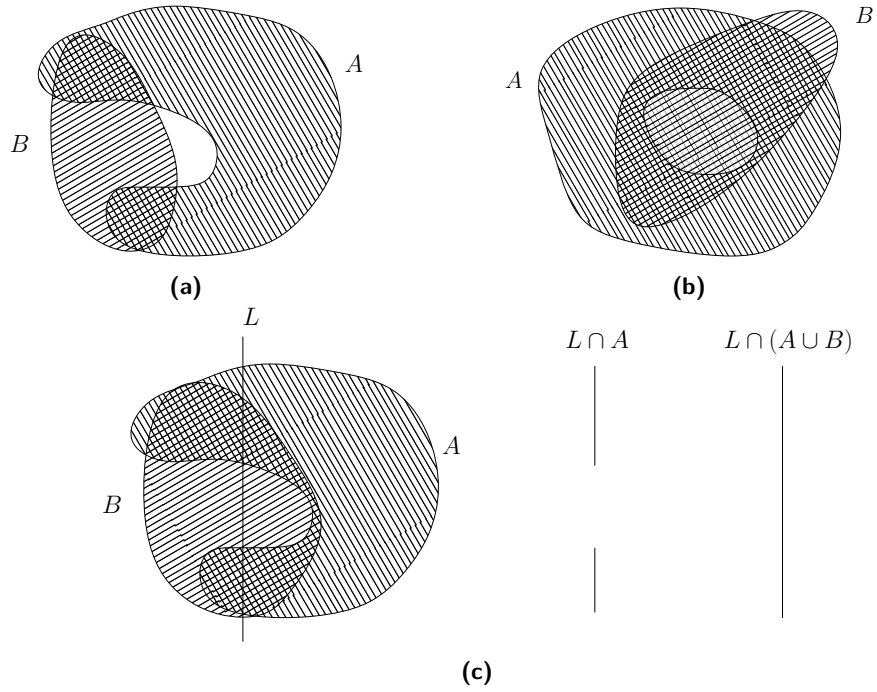


Figure 1 Suppose $A = \mathcal{S}_{\mathtt{C}}$ and $B = \mathcal{S}_{\{f\}}$. We can hope to certify $f \notin \mathtt{C} \iff A \cup B \neq A$ by noting that the numbers of 1-dimensional holes are different between $A \cup B$ and A. There can be many scenarios: (a) A and B are both contractible (do not have holes), but their union $A \cup B$ has a hole. Or (b) A has a hole in its center, but B covers it, so that $A \cup B$ is now contractible. Or (c) $A \cup B$ and A are both contractible, but if we look at a section L of $A \cup B$, we see that $L \cap A$ has 2 connected components, but $L \cap (A \cup B)$ has only 1. This shows why we need to look at homologies of subspaces or subcomplexes.

functions, we associate a simplicial complex $\mathcal{S}_{\mathtt{C}}$ to $\mathtt{C}$ in a way to be described in a moment; to show that a function f is not in $\mathtt{C}$, it suffices to show that $\mathcal{S}_{\mathtt{C}}$ is different from $\mathcal{S}_{\mathtt{C}\cup\{f\}}$ [1]. In this paper, we attempt to do so by showing that the (co)homologies [2] of $\mathcal{S}_{\mathtt{C}}$ and those of $\mathcal{S}_{\mathtt{C}\cup\{f\}}$ (or those of corresponding subcomplexes) are different. Figure 1 illustrates this idea.

▶ **Definition 1.1.** Write $[n] = \{0, 1, \ldots, n-1\}$. For a class $\mathtt{C} \subseteq [2]^{[n]}$ consisting of Boolean functions on a common domain $[n]$, $\mathcal{S}_{\mathtt{C}}$ is constructed as follows: There are $|[n] \times [2]| = 2n$ vertices, each labeled by an input/output pair $u \mapsto b$ with $u \in [n]$ and $b \in [2]$. For each function $f \in \mathtt{C}$, one adds to $\mathcal{S}_{\mathtt{C}}$ a maximal simplex on the n vertices with labels of the form $u \mapsto f(u)$. Vertices that don't belong to any such simplices are deleted.

In theoretical computer science, we will be mostly interested in the case when $[n] = [2^d] \cong [2]^d$ is the set of length d boolean strings. See Section 5 for examples. We use this idea to achieve several results in this paper.

[1] For readers unfamiliar with simplicial complexes, Appendix A provides a quick introduction.

[2] or in particular the Betti numbers, i.e. ranks of (co)homologies. For readers unfamiliar with homology, this is roughly speaking the "number of holes" of different dimensions in the simplicial complex. Intuitively speaking, the "dimension" of a hole is 1 if the hole "looks like" a circle, 2 if it "looks like" a sphere, and so on.

1.1 Aspnes-Beigel-Furst-Rudich Bound via Homology

Let POLYTHR_d^k denote the class of polynomial threshold functions of degree k on input space $\{-1,1\}^d$. The *polynomial threshold degree* of a Boolean function f is the smallest k such that $f \in \text{POLYTHR}_d^k$. We give a new proof of Aspnes et al.'s result [2] that gives the polynomial threshold degrees of symmetric functions. The primary conduit is a general "maximal principle for polynomial thresholds," which is derived by looking at how adding a function to a degree bounded polynomial threshold class changes low dimensional Betti numbers. It says

▶ **Theorem 1.2** (Maximal Principle for Polynomial Threshold). *Let* $\mathtt{C} := \text{POLYTHR}_d^k$*, and let* $f : \{-1,1\}^d \to \{-1,1\}$ *be a function. We want to know whether* $f \in \mathtt{C}$*.*

Suppose there exists a function $g \in \mathtt{C}$ *(a "local maximum" for approximating* f*) such that: for each* $h \in \mathtt{C}$ *that differs from* g *on exactly one input* u*, we have* $g(u) = f(u) = \neg h(u)$*. If* $g \neq f$*, then* $f \notin \mathtt{C}$*. (i.e., if* $f \in \mathtt{C}$*, then the "local maximum"* g *must be a "global maximum").*

We furthermore prove in the full paper that adding PARITY to $\text{POLYTHR}_d^k, k < d$, "covers up" the only hole in $\mathcal{S}_{\text{POLYTHR}_d^k}$. In general, $\mathcal{S}_{\text{POLYTHR}_d^k \cup \{f\}}$ "has no holes" iff f has no weak representation by degree k polynomials [19]. [3]

1.2 VC Dimension Bound via Homology

We exhibit a surprising connection of our framework to classical learning theory. **VC dimension** of a class $\mathtt{C}$ is defined as the size of the largest subset U of the input space such that $\mathtt{C} \restriction U$ contains all Boolean functions on U. It is roughly the number of samples needed to learn an unknown function f from a known class $\mathtt{C}$, up to multiplicative constants [10]. The **homological dimension** of a class $\mathtt{C}$, written $\dim_{\text{h}} \mathtt{C}$, is defined precisely in the full paper [19], but intuitively, for most cases, it is one plus the highest dimension of any nontrivial homology group in $\mathcal{S}_{\mathtt{C}}$ [4]. Then we prove that

▶ **Theorem 1.3.**

$$\dim_{\text{VC}} \mathtt{C} \leq \dim_{\text{h}} \mathtt{C}.$$

The equality cases include when $\mathtt{C}$ *is the class of parity functions (i.e. linear functionals over* $\mathbb{F}_2$*), the class of degree* $\leq k$ *polynomial threshold functions (for any fixed* k*), and the class of monotone conjunctions. This inequality cannot be improved to an equality, because for the class of conjunctions the gap between the two sides is 1, and for the class of delta functions on* $\{0,1\}^d$ *the homological dimension is* 2^d *but the VC dimension is 1.*

We also introduce an algebraic property of function classes called *Cohen-Macaulayness*, which is related to the corresponding notion in commutative algebra. We show that all Cohen-Macaulay classes satisfy this inequality with equality. It is nevertheless a major open problem to characterize the equality cases and the cases where the gap between the two sides is small (say, polynomial in d).

This beautiful result suggest that our homological theory captures something essential about computation, that it's not a coincidence that we can use "holes" to prove complexity separation.

---

[3] A polynomial p *weakly represents* a Boolean function f is for every input x such that $p(x) \neq 0$, we have $\text{sgn}(p(x)) = f(x)$.

[4] i.e. the "highest dimension of any hole in $\mathcal{S}_{\mathtt{C}}$." Intuitively speaking, the "dimension" of a hole is 1 if the hole "looks like" a circle, 2 if it "looks like" a sphere, and so on.

1.3 Homological Farkas Lemma

Farkas lemma [22] characterizes when a linear subspace intersects the positive cone using linear algebraic conditions. From our study of threshold function classes via the lens of homology, we obtain easily a *Homological Farkas* lemma which characterizes such situations using homological conditions. It roughly says that

▶ **Theorem 1.4** (Homological Farkas Lemma (Informal)). *Either a linear subspace intersects the positive cone, or its intersection with a part of the boundary of a neighboring cone has "holes," but not both.*

Section 6 provides a brief exposition on the precise statement and the intuition why it should be true.

In addition to the main results described above, we also provide a probabilistic interpretation of algebraic data, called Hilbert functions, derived from our theory and elucidate a connection to (co)sheaf theory, in the full paper [19]. We believe that these results are just the tip of a large, hidden (so far) iceberg that forms a multi-directional connection between computer science, algebra, and topology.

2 Related Works

2.1 Distributed Computability via Topology

Herlihy and Shavit [9] famously used topological techniques to characterize decision problems solvable in the basic shared memory model by asynchronous, wait-free protocols. While their work associates simplicial complexes to individual functions, we associate simplicial complexes to *classes* of functions. In addition, in contrast to their clever applications of elementary techniques of combinatorial topology, we leverage the more modern Stanley-Reisner theory and cellular resolutions heavily. They also focus on *continuous maps* much more than we do here, which is something our future work could possibly benefit from.

2.2 Algebraic Decision Tree Lower Bounds via Betti Numbers

A long line of work yielded lower bounds on algebraic decision trees via topological techniques [5, 6, 21, 20]. Typically these techniques first show that a set $A \subseteq \mathbb{R}^d$ of interest has high complexity in terms of some topological aspect, and then show that shallow algebraic decision trees cannot compute sets of too high complexity. Even disregarding the difference in domains ($\mathbb{R}^d$ vs a discrete set), these methods operate on a different level than what we propose in this paper. Here we compute the Betti numbers of function classes, not of functions themselves, and we prove lower bounds by observing that adding the function in question to the class of low complexity functions changes the Betti numbers of the class. In addition, we are not concerned with only Betti numbers graded by dimension, but also Betti numbers graded by partial functions (which correspond to Betti numbers of filtered subcomplexes $\mathcal{S}_{\mathsf{C}|\mathsf{g}}$; see Section 5 for definitions).

2.3 Geometric Complexity Theory

There is a superficial similarity of our work to Mulmuley's Geometric Complexity program [14] in that both associate mathematical objects to complexity classes and focus on finding obstructions to equality of complexity classes. In the case of geometric complexity, each class

is associated to a variety, and the obstructions sought are of representation-theoretic nature. In our case, each class is associated to a labeled simplicial complex, and the obstructions sought are of homological nature. But beyond this similarity, the inner workings of the two techniques are quite distinct. Whereas geometric complexity focuses on using algebraic geometry and representation theory to shed light on algebraic complexity classes (such as the permanent vs determinant question), our approach uses combinatorial algebraic topology and has a framework general enough to reason about any class of functions, not just algebraic functions. This generality allowed, for example, the unexpected connection to VC dimension. Thus there is no obvious relationship between GCT and our homological theory. However, there is a spiritual link. Indeed, Mulmuley and Sohoni proposed looking at higher dimension cohomology of the associated varieties in [14]. One possible direction for our future work is also to note that many classes have action by a symmetry group (see, e.g., [8]) and study how the Betti numbers break up into irreducible representations.

2.4 Homotopy Type Theory

A recent breakthrough in understanding the connection between algebraic topology and computer science is Homotopy Type Theory (HoTT) [17]. This theory concerns itself with rebuilding the foundation of mathematics via a homotopic interpretation of type theoretic semantics. Some of the key observations were that dependent sum types in type theory correspond to fibrations in homotopy theory, and equality types correspond to homotopies.

While HoTT only concerns itself with the B side (logic and semantics) of TCS, in this paper we primarily apply algebraic topology to the A side (complexity and learning theory). As such there really is no common ground between us in the technical details. However, early phases of our homological theory were inspired by the "fibration" philosophy of HoTT. In fact, the simplicial complex $\mathcal{S}_{\mathtt{C}}$ was first constructed as a sort of "fibration" (which turned out to be a cosheaf, and not a fibration) as explained in the full paper [19]. It remains to be seen if other aspects of HoTT could be illuminating in future research.

2.5 Computable Analysis and Topology

Computable analysis and topology study how topological spaces and functions on topological spaces can be represented in digital computers [18]. The theory builds a beautiful correspondence between computability via type II Turing machines on one hand and continuity of functions on the other hand that corroborates discoveries made in descriptive set theory [11, 13]. The initial spark for this paper was when the author realized that polylogarithmic time computation of a point in a topological space in the framework of computable analysis corresponds to polynomial time approximation schemes and also, roughly speaking, PAC learning: given more time, an algorithm should be able to pinpoint the desired point in a space more and more accurately, similar to how learning algorithms should be able to achieve better and better generalization errors with more samples and more computation time. But this correspondence to PAC learning ignores the probability of failure, which depends on the underlying data distribution. This initialized a search for a topological space encoding both the data distribution and the concept classes. The canonical suboplexes described in this paper turned out to be the right objects; see the cosheaf construction in the full paper for more details [19].

3 Does Our Proposal Run into Known Barriers?

Some of the most remarkable results in theoretical computer science in the last few decades are explicit "no-go" theorems that show that a common technique used in the past for proving complexity lower bounds cannot be extended to prove $\mathsf{P} \neq \mathsf{NP}$. These include relativization [3], algebrization [1], and natural proofs [15].

A priori, our framework is not blocked by the relativization or algebrization barriers because there is no reason to expect homology computations to relativize.

3.1 Razborov-Rudich Natural Proofs

Based on the methods presented in this paper, one might try to show $\mathsf{NP} \not\subseteq \mathsf{P/poly}$ by showing that the Betti numbers of $\mathcal{S}_{\mathsf{SIZE}(d^c)}$ differ from those of $\mathcal{S}_{\mathsf{SIZE}(d^c)\cup\{\mathsf{3SAT}_d\}}$, for any c and large enough d. Would this be a natural proof in the sense of Razborov and Rudich [15]?

A predicate $\mathcal{P}$ on functions with d-bit inputs is called **natural** if it satisfies

- (Constructiveness) It is polynomial time in its input size: there is an $2^{O(d)}$-time algorithm that on input the graph of a function f, outputs $\mathcal{P}(f)$.
- (Largeness) A random function f satisfies $\mathcal{P}(f) = 1$ with probability at least $\frac{1}{d}$.

▶ **Theorem 3.1** (Razborov-Rudich [15]). *Suppose there is no subexponentially strong one-way functions. Then there exists a constant c such that no natural predicate $\mathcal{P}$ maps $\mathsf{SIZE}(d^c)$ to 0.*

In our case, since $\mathsf{SIZE}(d^c)$ has $2^{\mathsf{poly}(d)}$ functions, naively computing the dimension-$(2^d - k)$ homology of $\mathcal{S}_{\mathsf{SIZE}(d^c)\cup\{\mathsf{3SAT}_d\}}$ for any constant k requires computing the ranks of two $2^{\mathsf{poly}(d)}$-sized matrix, which is already superpolynomial time in 2^d, violating the "constructiveness" of natural proofs. It is unknown whether the "largeness" condition is also violated, but, for any fixed dimension r, we conjecture that the probability a random total function f changes the dimension r homology of $\mathcal{S}_{\mathsf{SIZE}(d^c)}$ is exponentially small. Thus a priori this homological technique is not natural (barring the possibility that in the future, advances in the structure of $\mathcal{S}_{\mathsf{SIZE}(d^c)}$ yield efficient algorithms for its homology). [5]

4 Discussion

We anticipate several questions about our approach and provide corresponding retorts.

4.1 The Aspnes-Beigel-Furst-Rudich bound is an easy result; is your technique really new?

We agree that we are proving old results which are not particularly difficult, but we contend that the proofs really are different and serve as proof of concepts for future endeavors.

There is a *local-global philosophy* of our homological approach to complexity, inherited from algebraic topology. If we are interested in showing $f \notin \mathtt{C}$, we first examine the intersections of f with certain fragments of functions in $\mathtt{C}$, determined by the Betti numbers of $\mathtt{C}$ (this is the local step), and then piece together these fragments with nontrivial intersections with

[5] In general, given the contents of the full paper, one may also want to show that the ideal $I^{\star}_{\mathsf{SIZE}(d^c)\boxtimes\{\mathsf{3SAT}_d\}}$ is principal by showing that its Betti numbers are all zero except at dimension 0. Computing the Betti numbers of an arbitrary ideal is NP-hard in the number of generators [4], which is $\Omega(2^d)$ in this case. Thus a priori it seems unlikely this algebraic method is constructive.

f to draw conclusions about "holes" f creates or destroys (this is the global step). This is markedly different from conventional wisdom in computer science, which seeks to show that a function, such as $f = 3\text{SAT}$, has some property that no function in a class, say $\mathtt{C} = \mathsf{P}$, has. In that method, there is no *global* step that argues that some nontrivial global property of $\mathtt{C}$ changes after adding f into it.

This philosophy is evident in our maximal principle, where the "local maximum" condition is saying that when one looks at the intersections with f of g and its "neighbors" (local), these intersections together form a hole that f creates when added to $\mathtt{C}$ (global). [6] It certainly does not look like the maximal principle can be reduced to a "separation by property", as it seems to depend fundamentally on the function f and the class $\mathtt{C}$ at the same time.

The original proof of the Aspnes-Beigel-Furst-Rudich bound primarily operates through a theory of "strong" and "weak" degrees of functions built via linear programming duality, and contains no notion of "locally maximal approximating polynomial thresholds" that is central to the maximal principle. It is not clear if it is possible to reduce our proof to theirs.

4.2 Your Betti number results seem to follow from previous work on algebraic decision trees.

As explained in Section 2, the Betti number bounds on algebraic decision trees are for the Betti numbers of *semialgebraic* sets represented by *individual* functions, while we compute graded Betti numbers of the *simplicial complex* induced by a *class* of functions. All of our Betti number results for nontrivial classes such as polynomial thresholds and linear functionals are new.

4.3 Do we always have a "homological certificate" for complexity lower bounds?

We won't necessarily be able to spot a difference in homology between $\mathcal{S}_{\mathtt{C}}$ and $\mathcal{S}_{\mathtt{C}\cup\{f\}}$ (though this is the case for, for example, $\mathtt{C} = \text{POLYTHR}_d^k$ and $f = \text{PARITY}_d$). But, assuming the definitions in Section 5.2, we will always be able to spot a difference between pairs of subcomplexes $\mathcal{S}_{\mathtt{C}\downharpoonright\mathtt{g}} \subseteq \mathcal{S}_{\mathtt{C}}$ and $\mathcal{S}_{\mathtt{C}\cup\{f\}\downharpoonright\mathtt{g}} \subseteq \mathcal{S}_{\mathtt{C}\cup\{f\}}$ for some partial function $\mathtt{g}$. For example, trivially take $\mathtt{g} = f$; in general there is always some non-total partial function $\mathtt{g}$ that works (see full paper [19] for precise statement and proof).

4.4 It seems hard to scale your method to larger classes because it becomes too combinatorial too quickly.

This exponentiality in fact already occurs with the classes studied in this paper, but by using the structures of each class we were still able to obtain Betti numbers. Note that homology is polynomial time in the number of bits encoding the simplicial complex (think of it as the size of the corresponding hypergraph). So if the complexes were just polynomially large, then such an approach would probably run into the natural proof barrier.

[6] The homological intuition, in more precise terms, is that a local maximum $g \neq f \in \mathtt{C}$ implies that the filtered class $\mathtt{C} \downharpoonright (f \cap g)$ consists of a single point with label g, so that when f is added to $\mathtt{C}$, a zero-dimensional hole is created.

4.5 Would your method apply to less algebraic complexity classes?

It is important to note that there are two distinct levels of algebra involved in this paper: the algebra in the algebraic topology used on the *class* level (ex: Stanley-Reisner theory), and the algebraic structure used on the *function* level (ex: vector space structure in the class of linear functional). They are independent usages of algebra, and one can apply regardless of the other. For example, we have computed Betti numbers for classes that are not so algebraic such as conjunctions (and by symmetry disjunctions); it seems plausible to build upon these results to obtain results on the Betti numbers of circuit classes. Even for the more algebraic classes, the techniques in computing Betti numbers in the paper don't quite use all of the algebraic structures; for example the polynomial threshold computation really only depends on its linear structure, but not its multiplicative structure. So the lack of *algebraic* structure does not seem like the most pressing obstacle, but the lack of *any* structure whatsoever is probably more worrying. For most complexity classes the key seems to be to pick up an approximating class that 1) is representative of the difficulties of the class and 2) has enough structure to give rise to simplicial complexes amenable to analysis. In any case, the intuition for the homological angle of complexity is quite undeveloped at this stage; a priori, there is no reason to even think that we can compute the Betti numbers of polynomial thresholds, linear functionals, conjunctions, etc, but it is done nevertheless. So there is cause for optimism.

5 Warmups

To illustrate the main ideas of this paper without being mired in the algebraic details, we walk through some examples that require only comfort with combinatorics, basic knowledge of topology and simplicial complexes, and a geometric intuition for "holes." A short introduction to simplicial complexes is included as Appendix A. A brief note about notation: $[n]$ denotes the set $\{0, \ldots, n-1\}$, and $[n \to m]$ denotes the set of functions from domain $[n]$ to codomain $[m]$. The notation $\mathtt{f} :\subseteq A \to B$ specifies a partial function from domain A to codomain B. $\dagger$ represents the partial function with empty domain.

5.1 The Complete Class

If $\mathtt{C} = [n \to 2]$, then one can see that $\mathcal{S}_\mathtt{C}$ is isomorphic to the 1-norm unit sphere $S_1^{n-1} := \{\|x\|_1 = 1 : x \in \mathbb{R}^n\}$ (also known as an *orthoplex*, shown in Figure 2a). Indeed, each function $f \in \mathtt{C}$ adds a facet to $\mathcal{S}_\mathtt{C}$ corresponding to the standard simplex in an orthant of $\mathbb{R}^n$, and together they generate the 1-norm unit sphere. For general $\mathtt{C}$, $\mathcal{S}_\mathtt{C}$ can be realized as a subcomplex of S_1^{n-1}. For this reason, $\mathcal{S}_\mathtt{C}$ is called the **canonical suboplex** of $\mathtt{C}$, where "suboplex" is short for "sub-orthoplex."

5.2 Delta Function is Not Linear

Let $\textsc{linfun}_d \cong (\mathbb{F}_2^d)^*$ be the class of linear functionals of a d-dimensional vector space V over $\mathbb{F}_2$. If $d \geq 2$, then $\textsc{linfun}_d$ does not compute the indicator function $\mathbb{I}_\mathbf{1}$ of the singleton set $\{\mathbf{1} := 11\cdots1\}$. This is obviously true, but let's try to reason in a "homological way."

Define the partial function $\Upsilon : \mathbf{0} \mapsto 0, \mathbf{1} \mapsto 1$. Observe that for every partial linear functional $\mathtt{h} \supset \Upsilon$ *strictly extending* Υ, $\mathbb{I}_\mathbf{1}$ intersects $\mathtt{h}$ nontrivially. (Because $\mathbb{I}_\mathbf{1}$ is zero outside of Υ, and every such $\mathtt{h}$ must send at least one element to zero outside of Υ). I claim this completes the proof. *Combinatorially*, this is because if $\mathbb{I}_\mathbf{1}$ were a linear functional, then for any 2-dimensional subspace W of V containing $\{\mathbf{0}, \mathbf{1}\}$, the partial

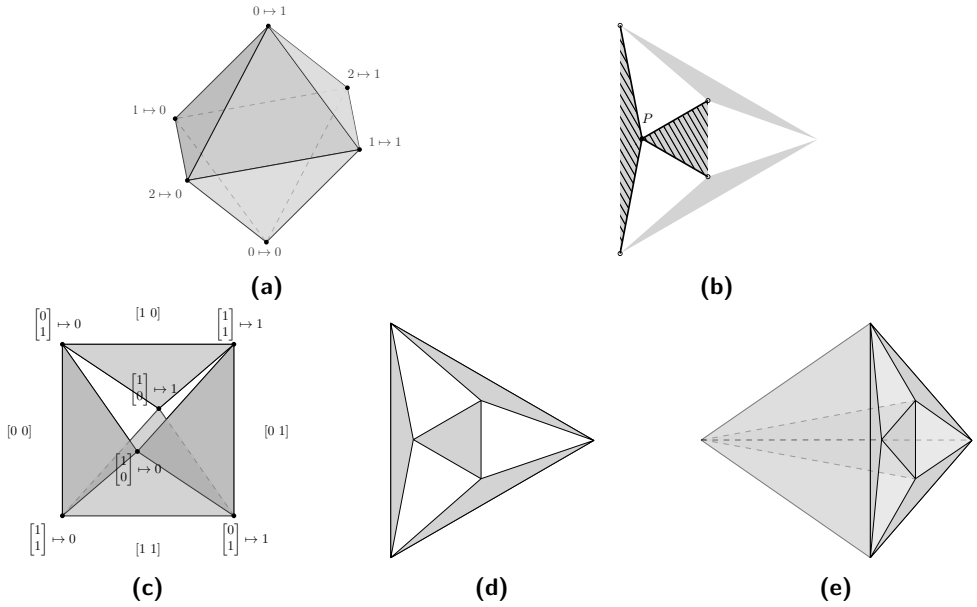


Figure 2 (a) The canonical suboplex of $[3 \to 2]$. (b) The open star $\operatorname{St} P$ of vertex P. (c) $\mathcal{S}_{\text{LINFUN}'_2}$ with vertices and facets labeled. (d) $\mathcal{S}_{\text{LINFUN}'_2}$ stretched flat. (e) $\mathcal{S}_{\text{LINFUN}_d}$ is just a cone over $\mathcal{S}_{\text{LINFUN}'_d}$.

function $\mathsf{h} :\subseteq V \to \mathbb{F}_2, \operatorname{dom}\mathsf{h} = W$,

$$\mathsf{h}(u) = \begin{cases} \Upsilon(u) & \text{if } u \in \operatorname{dom}\Upsilon \\ 1 - \mathbb{I}_{\mathbf{1}}(u) & \text{if } u \in \operatorname{dom}\mathsf{h} \setminus \operatorname{dom}\Upsilon \end{cases}$$

is a linear functional, and by construction, does not intersect $\mathbb{I}_{\mathbf{1}}$ on $W \setminus \{\mathbf{0}, \mathbf{1}\}$. *Homologically*, we are really showing the following

> *A section of* $\mathcal{S}_{\text{LINFUN}_d}$ *by an affine subspace corresponding to* Υ *"has a hole" that is "filled up" when* $\mathbb{I}_{\mathbf{1}}$ *is added to* LINFUN_d. (*)

The meaning of this statement will seem cryptic right now, so let us elaborate.

Figure 2c exhibits the complex $\mathcal{S}_{\text{LINFUN}'_2}$, where $\text{LINFUN}'_d \subseteq [\mathbb{F}_2^d \setminus \{\mathbf{0}\} \to \mathbb{F}_2]$ is essentially the same class as LINFUN_d, except we delete $\mathbf{0}$ from the domain of every function. Notice that the structure of "holes" is not trivial at all: $\mathcal{S}_{\text{LINFUN}'_2}$ has 3 holes in dimension 1 but no holes in any other dimension. An easy way to visualize this is to pick one of the triangular holes; if you put your hands around the edge, pull the hole wide, and flatten the entire complex onto a flat plane, then you get Figure 2d.

It is easy to construct the canonical suboplex of LINFUN_d from that of LINFUN'_d: $\mathcal{S}_{\text{LINFUN}_d}$ is just a cone over $\mathcal{S}_{\text{LINFUN}'_d}$, where the cone vertex has the label $[0\ 0]^T \mapsto 0$ (Figure 2e). This is because every function in LINFUN_d shares this input/output pair. Note that a cone over any base has no hole in any dimension, because any hole can be contracted to a point in the vertex of the cone. This is a fact we will use again very soon.

Let $\mathsf{C} \subseteq [n \to 2]$, and let $\mathsf{f} :\subseteq [n] \to [2]$ be a partial function. Define the **filtered class** $\mathsf{C} \lfloor \mathsf{f}$ to be

$$\{g \setminus \mathsf{f} : g \in \mathsf{C}, g \supseteq \mathsf{f}\} \subseteq [[n] \setminus \operatorname{dom}\mathsf{f} \to [2]]$$

Unwinding the definition: $\mathsf{C} \lfloor \mathsf{f}$ is obtained by taking all functions of C that extend f and ignoring the inputs falling in the domain of f. The canonical suboplex $\mathcal{S}_{\mathsf{C}\lfloor\mathsf{f}}$ can be seen

to be isomorphic to an affine section of $\mathcal{S}_{\mathsf{C}}$, when the latter is embedded as part of the L_1 unit sphere S_1^{n-1}. Figure 3a shows an example when f has a singleton domain. Indeed, recall $\textsc{linfun}'_d$ is defined as $\textsc{linfun}_d \downharpoonright \{\mathbf{0} \mapsto 0\}$, and we may recover $\mathcal{S}_{\textsc{linfun}'_d}$ as an affine cut through the "torso" of $\mathcal{S}_{\textsc{linfun}_d}$ (Figure 3b). This explains the "affine section" part of (*).

To continue our elaboration, we need a "duality principle" in algebraic topology called the

▶ **Lemma 5.1** (Nerve Lemma (Informal)). *Let $\mathcal{U} = \{U_i\}_i$ be a* "nice" *(to be explained below) cover* [7] *of a topological space X. The* **nerve** *$\mathcal{N}_{\mathcal{U}}$ of $\mathcal{U}$ is defined as the simplicial complex with vertices $\{V_i : U_i \in \mathcal{U}\}$, and with simplices $\{V_i\}_{i\in S}$ for each index set S such that $\bigcap\{U_i : i \in S\}$ is nonempty.*

Then, for each dimension d, the set of d-dimensional holes in X is bijective with the set of d-dimensional holes in $\mathcal{N}_{\mathcal{U}}$.

What kind of covers are "nice?" Open covers in general spaces, or subcomplex covers in simplicial (or CW) complexes, are considered "nice", if in addition they satisfy the following requirements (*acyclicity*).

- Each set of the cover must have no holes.
- Each nontrivial intersection of a collection of sets must have no holes.

An example is the star cover: For vertex V in a complex, the **open star** $\operatorname{St} V$ of V is defined as the union of all open simplices whose closure meets V (see Figure 2b for an example). If the cover $\mathcal{U}$ consists of the open stars of every vertex in a simplicial complex X, then $\mathcal{N}_{\mathcal{U}}$ and X are isomorphic as complexes.

It turns out that $\mathcal{S}_{\textsc{linfun}'_d} = \mathcal{S}_{\textsc{linfun}_d \downharpoonright (\mathbf{0}\mapsto 0)}$ (a complex of dimension $2^d - 2$) has holes in dimension $d-1$ — in fact, these are the only holes in $\mathcal{S}_{\textsc{linfun}'_d}$ and the homological dimension of $\textsc{linfun}'_d$ equals $d - 1 + 1 = d$, coinciding with its VC dimension. The proof is nontrivial and deferred to the full paper [19]. This can be clearly seen in our example when $d = 2$ (Figure 2d), which has 3 holes in dimension $d - 1 = 1$. Furthermore, for every partial linear functional h (a linear functional defined on a linear subspace), $\mathcal{S}_{\textsc{linfun}_d \downharpoonright \mathsf{h}}$ also has holes, in dimension $d-1-\dim(\operatorname{dom}\mathsf{h})$. Figure 3c show an example for $d = 2$ and $\mathsf{h} = [1\ 1]^T \mapsto 1$. This is in particular true for $\mathsf{h} = \Upsilon$. But when we add $\mathbb{I}_{\mathbf{1}}$ to $\textsc{linfun}_d$ to obtain $\mathtt{D} := \textsc{linfun}_d \cup \{\mathbb{I}_{\mathbf{1}}\}$, $\mathcal{S}_{\mathtt{D}\downharpoonright\Upsilon}$ now does not have any hole! Figure 3d clearly demonstrates the case $d = 2$. For general d, note that $\mathcal{S}_{\textsc{linfun}_d \downharpoonright \Upsilon}$ has a "nice" cover by the open stars

$$\mathcal{C} := \{\operatorname{St} V : V \text{ has label } u \mapsto r \text{ for some } u \in \mathbb{F}_2^d \setminus \{\mathbf{0}, \mathbf{1}\} \text{ and } r \in \mathbb{F}_2\},$$

where the stars are with respect to $\mathcal{S}_{\textsc{linfun}_d \downharpoonright \Upsilon}$. When we added $\mathbb{I}_{\mathbf{1}}$ to form $\mathtt{D}$, the collection $\mathcal{C}' := \mathcal{C} \cup (\triangle_{\mathbb{I}_{\mathbf{1}} \setminus \Upsilon})$ is a "nice" cover of $\mathcal{S}_{\mathtt{D}\downharpoonright\Upsilon}$, where $\triangle_{\mathbb{I}_{\mathbf{1}}\setminus\Upsilon}$ is the face of $\mathbb{I}_{\mathbf{1}}$'s simplex generated by vertices with labels of the form $u \mapsto \mathbb{I}_{\mathbf{1}}(u), u \neq \mathbf{0}, \mathbf{1}$. Thus the nerve $\mathcal{N}_{\mathcal{C}'}$ has the same holes as $\mathcal{S}_{\mathtt{D}\downharpoonright\Upsilon}$, by the Nerve Lemma. But observe that $\mathcal{N}_{\mathcal{C}'}$ is a cone! ... which is what our "combinatorial proof" of $\mathbb{I}_{\mathbf{1}} \notin \textsc{linfun}_d$ really showed.

More precisely,

1. a collection of stars $S := \{\operatorname{St} V : V \in \mathcal{V}\}$ has nontrivial intersection iff there is a partial linear functional extending the labels of each $V \in \mathcal{V}$.
2. We showed $\mathbb{I}_{\mathbf{1}} \setminus \Upsilon$ intersects every partial linear functional strictly extending Υ.
3. Therefore, a collection of stars S in $\mathcal{C}'$ intersects nontrivially iff $S \cup \{\triangle_{\mathbb{I}_{\mathbf{1}}\setminus\Upsilon}\}$ also intersects nontrivially.

[7] A cover of a space X is just a collection of sets whose union is X.

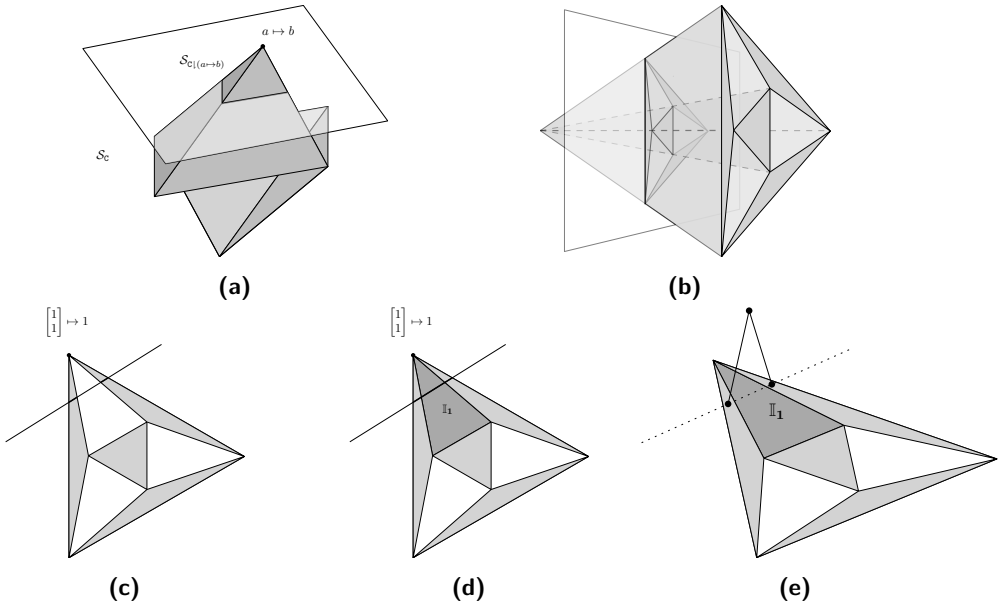


Figure 3 (a) $\mathcal{S}_{\mathtt{C}\downarrow(a\mapsto b)}$ is an affine section of $\mathcal{S}_{\mathtt{C}}$. (b) We may recover $\mathcal{S}_{\text{LINFUN}'_d}$ as a linear cut through the "torso" of $\mathcal{S}_{\text{LINFUN}_d}$. (c) $\mathcal{S}_{\text{LINFUN}_2\downarrow\{[0\ 0]^T\mapsto 0,[1\ 1]^T\mapsto 1\}}$ is isomorphic to the affine section as shown; it has "a single dimension zero hole." (d) When we add $\mathbb{I}_\mathbf{1}$ to LINFUN_d to obtain $\mathtt{D} := \text{LINFUN}_d \cup \{\mathbb{I}_\mathbf{1}\}$, $\mathcal{S}_{\mathtt{D}\downarrow\Upsilon}$ now does not have any hole! (e) The nerve $\mathcal{N}_{\mathcal{C}'}$ overlayed on $\mathtt{D} = \text{LINFUN}_2 \cup \{\mathbb{I}_\mathbf{1}\}$. Note that $\mathcal{N}_{\mathcal{C}'}$ is a cone over its base of 2 points.

In other words, in the nerve of $\mathcal{C}'$, $\triangle_{\mathbb{I}_\mathbf{1}\setminus\Upsilon}$ forms the vertex of a cone over all other $\operatorname{St} V \in \mathcal{C}$. In our example of LINFUN_2, this is demonstrated in Figure 3e.

Thus, to summarize,

- $\mathcal{N}_{\mathcal{C}'}$, being a cone, has no holes.
- By the Nerve Lemma, $\mathcal{S}_{\mathtt{D}\downarrow\Upsilon}$ has no holes either.
- Since $\mathcal{S}_{\text{LINFUN}_d\downarrow\Upsilon}$ has holes, we know $\mathtt{D} \downarrow \Upsilon \neq \text{LINFUN}_d \downarrow \Upsilon \implies \mathtt{D} \neq \text{LINFUN}_d$, i.e. $\mathbb{I}_\mathbf{1} \notin \text{LINFUN}_d$, as desired.

While this introduction took some length to explain the logic of our approach, much of this is automated in the theory we develop in this paper, which leverages existing works on Stanley-Reisner theory and cellular resolutions. The Nerve Lemma will in fact never be explicitly applied but rather is implicit in these machineries.

6 Homological Farkas Primer

We give a brief exposition on what the Homological Farkas Lemma says. Farkas Lemma is a simple result from linear algebra, but it is an integral tool for proving weak and strong dualities in linear programming [7], matroid theory [22], and game theory [12, chapter 7], among many other things.

▶ **Lemma 6.1** (Farkas Lemma). *Let $L \subseteq \mathbb{R}^n$ be a linear subspace not contained in any coordinate hyperplanes, and let $P = \{x \in \mathbb{R}^n : x > 0\}$ be the positive cone. Then either*

- *L intersects P, or*
- *L is contained in the kernel of a nonzero linear functional whose coefficients are all nonnegative.*

but not both.

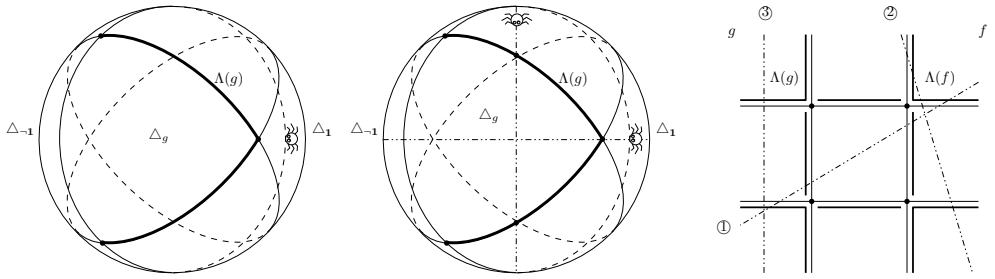


Figure 4 (a) An example of a $\Lambda(g)$. Intuitively, $\Lambda(g)$ is the part of $\partial\triangle_g$ that can be seen from an observer in $\triangle_{\mathbf{1}}$. (b) An illustration of Homological Farkas Lemma. The horizontal dash-dotted plane intersects the interior of $\triangle_{\mathbf{1}}$, but its intersection with any of the $\Lambda(f), f \neq \mathbf{1}, \neg\mathbf{1}$ has no holes. The vertical dash-dotted plane misses the interior of $\triangle_{\mathbf{1}}$, and we see that its intersection with $\Lambda(g)$ as shown has two disconnected components. (c) Example application of the affine version of homological Farkas lemma.. Let the hyperplanes (thin lines) be oriented such that the square S at the center is on the positive side of each hyperplane. The bold segments indicate the Λ of each region. Line 1 intersects S, and we can check that its intersection with any bold component has no holes. Line 2 does not intersect the closure $\overline{S}$, and we see that its intersection with $\Lambda(f)$ is two points, so has a "zeroth dimension" hole. Line 3 does not intersect $\overline{S}$ either, and its intersection with $\Lambda(g)$ consists of a point in the finite plane and another point on the circle at infinity.

Farkas Lemma is a characterization of when a linear subspace intersects the positive cone in terms of *linear conditions*. An alternate view important in computer science is that Farkas Lemma provides a *linear certificate* for when this intersection does not occur. Analogously, our Homological Farkas Lemma will characterize such an intersection in terms of *homological conditions*, and simultaneously provide a *homological certificate* for when this intersection does not occur.

Before stating the Homological Farkas Lemma, we first introduce some terminology.

For $g : [n] \to \{1, -1\}$, let $P_g \subseteq \mathbb{R}^n$ denote the open cone whose points have signs given by g. Consider the intersection $\triangle_g$ of $\overline{P_g}$ with the unit sphere S^{n-1} and its interior $\mathring{\triangle}_g$. $\mathring{\triangle}_g$ is homeomorphic to an open simplex. For $g \neq \neg\mathbf{1}$, define $\Lambda(g)$ to be the union of the facets F of $\triangle_g$ such that $\mathring{\triangle}_g$ and $\mathring{\triangle}_{\mathbb{1}}$ sit on opposite sides of the affine hull of F. Intuitively, $\Lambda(g)$ is the part of $\partial\triangle_g$ that can be seen from an observer in $\mathring{\triangle}_{\mathbf{1}}$ (illustrated by Figure 4a).

The following homological version of Farkas Lemma naturally follows from our homological technique of analyzing the complexity of threshold functions.

▶ **Theorem 6.2** (Homological Farkas Lemma). *Let $L \subseteq \mathbb{R}^n$ be a linear subspace. Then either*

- *L intersects the positive cone $P = P_{\mathbf{1}}$, or*
- *$L \cap \Lambda(g)$ for some $g \neq \mathbf{1}, \neg\mathbf{1}$ is nonempty and has holes.*

but not both.

Figure 4b illustrates an example application of this result.

One direction of the Homological Farkas Lemma has the following intuition. As mentioned before, $\Lambda(g)$ is essentially the part of $\partial\triangle_g$ visible to an observer Tom in $\mathring{\triangle}_{\mathbf{1}}$. Since the simplex is convex, the image Tom sees is also convex. Suppose Tom sits right on L (or imagine L to be a subspace of Tom's visual field). If L indeed intersects $\mathring{\triangle}_{\mathbf{1}}$, then for $L \cap \Lambda(g)$ he sees some affine space intersecting a convex body, and hence a convex body in itself. Since Tom sees everything (i.e. his vision is homeomorphic with the actual points), $L \cap \Lambda(g)$ has no holes, just as Tom observes. In other words, if Tom is inside $\mathring{\triangle}_{\mathbf{1}}$, then he cannot tell $\Lambda(g)$

is nonconvex by his vision alone, for any g. Conversely, the Homological Farkas Lemma says that if Tom is outside of $\mathring{\triangle}_{\mathbf{1}}$ and if he looks away from $\mathring{\triangle}_{\mathbf{1}}$, he will always see a nonconvex shape in some $\Lambda(g)$.

As a corollary to Theorem 6.2, we can also characterize when a linear subspace intersects a region in a linear hyperplane arrangement, and when an affine subspace intersects a region in an affine hyperplane arrangement, both in terms of homological conditions (see the full version of this paper [19] for details). A particular simple consequence, when the affine subspace either intersects the interior or does not intersect the closure at all, is illustrated in Figure 4c.

7 Overview of techniques and proofs

In this section we assume that the reader has the necessary algebraic and topological background. The complex $\mathcal{S}_{\mathtt{C}}$ is analyzed using Stanley-Reisner theory, which involves studying its face ideal $I_{\mathtt{C}}$ (i.e. the ideal consisting of monomials representing sets of vertices *not* in the complex $\mathcal{S}_{\mathtt{C}}$) and its Alexander dual $I_{\mathtt{C}}^{\star}$, primarily through the lens of free resolutions. It turns out that the Alexander dual is much easier to work with than the face ideal itself. The rank of each multigraded syzygy in its minimal resolution gives the Betti number of an appropriate dimension in the corresponding subcomplex of $\mathcal{S}_{\mathtt{C}}$ (more precisely, a certain link). This set of syzygy rank/Betti numbers is the principal topological invariant we use to separate classes in this work. Most resolutions here are computed as (co)cellular resolutions, i.e. we find labeled CW complexes whose (co)chain complexes resolve the ideals in question.

For the proof of Aspnes et al.'s theorem, we first obtain the cocellular resolutions of degree-bounded polynomial threshold classes $\textsc{polythr}_d^k$, which are supported on a natural CW decomposition of spheres. We yield the maximal principle (Theorem 1.2) by analyzing how dimension 1 Betti numbers change when a new function is added to $\textsc{polythr}_d^k$. We finish by constructing locally maximal approximating polynomial thresholds for symmetric functions. These local maxima are in general symmetric polynomial thresholds that encode the sign changes of the function in question as a polynomial of the sum of input bits.

Homological dimension is actually defined as the projective dimension of $I_{\mathtt{C}}^{\star}$, and as such it is the length of its minimal resolution. For the VC dimension bound, we give two proofs. The first observes that any minimal resolution of $I_{\mathtt{C}}^{\star}$ via relabeling supports a resolution of $I_{\mathtt{C}\restriction U}^{\star}$, for any subset U of the input space. When U is a largest shattering set, this shows that the homological dimension of $\mathtt{C}$ is at least the homological dimension of $\mathtt{C} \restriction U$, which we know is equal to $|U|$ and also equal to the VC dimension of $\mathtt{C}$. The second proof observes $\mathcal{S}_{\mathtt{D}}$ on an input space of size n has nontrivial homology in dimension $n-1$ iff $\mathcal{S}_{\mathtt{D}}$ contains every function. By applying this observation to $\mathcal{S}_{\mathtt{C}} \restriction U$ for a largest shattering set U, we get a lower bound of the regularity of $I_{\mathtt{C}}$ by the VC dimension plus one. Since regularity of $I_{\mathtt{C}}$ equals the projective dimension of $I_{\mathtt{C}}^{\star}$ plus one, we arrive at the desired result.

The much harder part of Theorem 1.3 is actually showing that equality holds in the various cases and that inequality is strict in other cases. This is done by deriving the (co)cellular resolutions of many function classes common in learning theory, such as conjunctions (supported on a pyramid over a pile of cubes) and monotone conjunctions (supported on a cube), degree bounded polynomial thresholds (supported on a CW decomposition of the sphere), linear functionals over finite fields, and so on. The topological dimension of such (co)cellular resolutions yield the homological dimension of the class itself, and we obtain the different equality and inequality cases listed in Theorem 1.3.

Finally, the Homological Farkas Lemma is obtained by drawing a parallel between, on the one hand, the intersection of a linear subspace with the positive cone, and on the other, the containment of a function in a generalized notion of a threshold class, which is possible due to the spherical structure of the cocellular resolution of such a class.

Acknowledgements. I would like to thank Madhu Sudan for hours of discussion, which were crucial in distilling the technical contents of this work into a clear exposition accessible to a general mathematical audience. I would also like to thank Josh Grochow for help fine-tuning the presentation toward a TCS audience. In addition, thanks are due to Günter Ziegler, Ezra Miller, Bernd Sturmfels, and Fatemeh Mohammadi, who provided references and discussion on algebra and combinatorics during the formative period of this theory; and to Leslie Valiant and Boaz Barak for listening to my babbles and provide encouragement.

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 James Aspnes, Richard Beigel, Merrick L. Furst, and Steven Rudich. The expressive power of voting polynomials. *Combinatorica*, 14(2):135–148, 1994. doi:10.1007/BF01215346.

3 Theodore P. Baker, John Gill, and Robert Solovay. Relativizations of the P =? NP question. *SIAM J. Comput.*, 4(4):431–442, 1975. doi:10.1137/0204037.

4 Dave Bayer and Mike Stillman. Computation of Hilbert functions. *Journal of Symbolic Computation*, 14(1):31–50, 1992. URL: http://www.sciencedirect.com/science/article/pii/074771719290024X.

5 Michael Ben-Or. Lower bounds for algebraic computation trees (preliminary report). In David S. Johnson, Ronald Fagin, Michael L. Fredman, David Harel, Richard M. Karp, Nancy A. Lynch, Christos H. Papadimitriou, Ronald L. Rivest, Walter L. Ruzzo, and Joel I. Seiferas, editors, *Proceedings of the 15th Annual ACM Symposium on Theory of Computing, 25-27 April, 1983, Boston, Massachusetts, USA*, pages 80–86. ACM, 1983. doi:10.1145/800061.808735.

6 Anders Björner, László Lovász, and Andrew CC Yao. Linear decision trees: volume estimates and topological bounds. In *Proceedings of the twenty-fourth annual ACM symposium on Theory of computing*, pages 170–177. ACM, 1992. URL: http://dl.acm.org/citation.cfm?id=129730.

7 Stephen P. Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge University Press, Cambridge, UK ; New York, 2004.

8 Joshua A. Grochow. Unifying known lower bounds via geometric complexity theory. *Computational Complexity*, 24(2):393–475, 2015. doi:10.1007/s00037-015-0103-x.

9 Maurice Herlihy and Nir Shavit. The topological structure of asynchronous computability. *Journal of the ACM (JACM)*, 46(6):858–923, 1999. URL: http://dl.acm.org/citation.cfm?id=331529.

10 Michael Kearns and Umesh Vazirani. *An Introduction to Computational Learning Theory*. MIT Press, jan 1994.

11 A. S. Kechris. *Classical descriptive set theory*. Number 156 in Graduate texts in mathematics. Springer-Verlag, New York, 1995.

12 Tjalling C Koopmans and others. *Activity analysis of production and allocation*. Wiley, 1951.

13 Yiannis N. Moschovakis. *Descriptive set theory*. Number v. 155 in Mathematical surveys and monographs. American Mathematical Society, Providence, R.I, 2nd ed edition, 2009.

14 Ketan Mulmuley and Milind A. Sohoni. Geometric complexity theory I: an approach to the P vs. NP and related problems. *SIAM J. Comput.*, 31(2):496–526, 2001. doi:10.1137/S009753970038715X.

15 Alexander A. Razborov and Steven Rudich. Natural proofs. *J. Comput. Syst. Sci.*, 55(1):24–35, 1997. doi:10.1006/jcss.1997.1494.

16 Joseph J. Rotman. *An introduction to algebraic topology*. Number 119 in Graduate texts in mathematics. Springer-Verlag, New York, 1988.

17 The Univalent Foundations Program. Homotopy Type Theory: Univalent Foundations of Mathematics. *arXiv preprint arXiv:1308.0729*, 2013. URL: https://arxiv.org/abs/1308.0729.

18 Klaus Weihrauch. *Computable Analysis*. Texts in Theoretical Computer Science. An EATCS Series. Springer Berlin Heidelberg, Berlin, Heidelberg, 2000. DOI: 10.1007/978-3-642-56999-9. URL: http://link.springer.com/10.1007/978-3-642-56999-9.

19 Greg Yang. A Homological Theory of Functions. *arXiv:1701.02302 [cs, math]*, jan 2017. arXiv: 1701.02302. URL: http://arxiv.org/abs/1701.02302.

20 Andrew Chi-Chih Yao. Lower bounds for algebraic computation trees with integer inputs. *SIAM J. Comput.*, 20(4):655–668, 1991. doi:10.1137/0220041.

21 Andrew Chi-Chih Yao. Decision tree complexity and betti numbers. In Frank Thomson Leighton and Michael T. Goodrich, editors, *Proceedings of the Twenty-Sixth Annual ACM Symposium on Theory of Computing, 23-25 May 1994, Montréal, Québec, Canada*, pages 615–624. ACM, 1994. doi:10.1145/195058.195414.

22 Günter M. Ziegler. *Lectures on Polytopes*, volume 152 of *Graduate Texts in Mathematics*. Springer New York, New York, NY, 1995. URL: http://link.springer.com/10.1007/978-1-4613-8431-1.

A A Crash Course on Simplicial Complexes

Our presentation will follow [16].

▶ **Definition 1.1.** A d-dimensional **simplex** is just the convex hull of some affine independent subset $\{v_0, \ldots, v_d\}$ of a Euclidean space. We denote such a simplex by $[v_0, \ldots, v_d]$. The **vertex set** of a simplex $\triangle$ is denoted $\mathrm{Vrt}(\triangle)$.

▶ **Definition 1.2.** If $\triangle$ is a simplex, then a **face** of $\triangle$ is a simplex $\triangle'$ with $\mathrm{Vrt}(\triangle') \subseteq \mathrm{Vrt}(\triangle)$. $\triangle'$ is a **proper face** if the inclusion is strict.

Here is the main definition.

▶ **Definition 1.3.** A **simplicial complex** K is a finite collection of simplices in some Euclidean space such that

- (Hereditary) if $\triangle \in K$, then every face of $\triangle$ is also in K.
- (Regular intersection) if $\triangle, \triangle' \in K$, then $\triangle \cap \triangle'$ is either empty or a common face of $\triangle$ and of $\triangle'$.

The maximal faces of K, i.e. $\triangle \in K$ not properly contained in another $\triangle' \in K$, are called **facets** of K.

▶ **Definition 1.4.** If K is a simplicial complex, its **underlying space** $|K|$ is the subspace of the ambient Euclidean space given by the union of its simplexes:

$$|K| := \bigcup_{\triangle \in K} \triangle.$$

Figure 5 (a) Valid simplicial complex. (b) Invalid simplicial complex, because the intersection of the two simplices is not a face of the top simplex.

Most often we represent simplicial complexes pictorially as its underlying space. Figure 5 gives an example of a valid and of an invalid simplicial complex.

▶ **Definition 1.5.** The **boundary** $\partial\triangle$ of a simplex $\triangle$ is the collection of its proper faces.

▶ **Definition 1.6.** Suppose $\triangle$ is a d-dimensional simplex. If $d = 0$, define $\mathring{\triangle} = \triangle$. Otherwise, define $\mathring{\triangle} = \triangle \setminus |\partial\triangle|$. $\mathring{\triangle}$ is called an **open simplex**. For contrast, a plain simplex $\triangle$ is also called a **closed simplex**. The **closure** of an open simplex is just the corresponding closed simplex.

The above viewpoint of simplicial complexes is geometric and concrete. There is an equivalent combinatorial view that is sometimes more convenient to work with.

▶ **Definition 1.7.** Let V be a finite set. An **abstract simplicial complex** K is a family of nonempty subsets of V, called **simplices**, such that

- (Atomic) if $v \in V$, then $\{v\} \in K$;
- (Hereditary) if $\triangle \in K$ and $\triangle' \subseteq \triangle$, then $\triangle' \in K$.

V is called the **vertex set of** K and a simplex $\triangle \in K$ with $d + 1$ elements is called a d**-dimensional simplex**.

It is not hard to see that every simplicial complex has an associated abstract simplicial complex, simply by taking the vertex set of each simplex; call this **abstraction**. It is also true that every abstract simplicial complex has a topological space, called its **geometric realization**, that is a simplicial complex and is unique up to homeomorphism. These two operations are inverse in the sense that the geometric realization of an abstraction is homeomorphic to the original simplicial complex, and the abstraction of a geometric realization is isomorphic (in a suitable sense) to the original abstract simplicial complex. For details, see [16].

In this paper, we assume the equivalence of the geometric and combinatorial views, and use them interchangeably, as appropriate in different situations.

▶ **Definition 1.8.** Given a set of subsets $\nabla \subseteq 2^V$ of the vertex set, the (abstract) simplicial complex K generated by ∇ is the hereditary closure of ∇, i.e.

$$K = \{\triangle \subseteq V : \exists \triangle' \in \nabla, \triangle \subseteq \triangle'\}.$$

One can easily check that this is an (abstract) simplicial complex.

In this language, Theorem 1.1 says that $\mathcal{S}_{\mathtt{C}}$ is the simplicial complex on the vertex set $V = [2^d] \times [2]$ generated by $\{\text{graph}\, f : f \in \mathtt{C}\}$, minus unused vertices.

Long Term Memory and the Densest K-Subgraph Problem*

Robert Legenstein[1], Wolfgang Maass[2], Christos H. Papapdimitriou[3], and Santosh S. Vempala[4]

1 **Institute for Theoretical Computer Science, Graz University of Technology, Graz, Austria**
robert.legenstein@igi.tugraz.at
2 **Institute for Theoretical Computer Science, Graz University of Technology, Graz, Austria**
maass@igi.tugraz.at
3 **Computer Science, Columbia University, NY, USA**
christos@cs.berkeley.edu
4 **Computer Science, Georgia Tech, Atlanta, USA**
vempala@gatech.edu

Abstract

In a recent experiment [9], a cell in the human medial temporal lobe (MTL) encoding one sensory stimulus starts to also respond to a second stimulus following a combined experience associating the two. We develop a theoretical model predicting that an assembly of cells with exceptionally high synaptic intraconnectivity can emerge, in response to a particular sensory experience, to encode and abstract that experience. We also show that two such assemblies are modified to increase their intersection after a sensory event that associates the two corresponding stimuli. The main technical tools employed are random graph theory, and Bernoulli approximations. Assembly creation must overcome a computational challenge akin to the Densest K-Subgraph problem, namely selecting, from a large population of randomly and sparsely interconnected cells, a subset with exceptionally high density of interconnections. We identify three mechanisms that help achieve this feat in our model: (1) a simple two-stage randomized algorithm, and (2) the "triangle completion bias" in synaptic connectivity [14] and a "birthday paradox", while (3) the strength of these connections is enhanced through Hebbian plasticity.

1998 ACM Subject Classification F.1.1 Models of Computation, I.2.6 Learning: Connectionism and neural nets

Keywords and phrases Brain computation, long term memory, assemblies, association

Digital Object Identifier 10.4230/LIPIcs.ITCS.2018.57

1 Introduction

How do sensory stimuli from entities in the outside world effect the creation of stable memories in the animal cortex, and how are such memories modified by further experience, for example by the introduction of associations between them? A recent experiment [9] provides certain interesting insights into these fundamental questions. They recorded from a total of 613 neurons in the medial temporal lobe (MTL, the brain region near the hippocampus

* This work was partially supported by the Human Brain Project of the European Union #604102 and #720270, and NSF grants CCF-1408635, CCF-1563838 and CCF-1717349.

long believed to be implicated in memory) of 14 human subjects. They presented, in a particular rigorous protocol running over several stages, many images of places and people, with repetitions and occasional superpositions. Several neurons were identified that fired consistently at the presentation of a particular place or person. One particular neuron in one subject may have fired consistently when an image of the Eiffel tower was presented, but failed to fire when other images were presented, such as an image of Barack Obama (example for the present illustration). Then a combined image of Obama in front of the Eiffel tower was presented, and, predictably, the neuron fired (as it always does when the Eiffel tower is seen). Remarkably, after this combined presentation the neuron also fired when an image of Obama was shown: the subject had *learned* the *association* between the two!

In this paper we propose a model for the formation and association of memories, based on random graphs and Hebbian plasticity, which we believe captures in a simplified way the basic mechanisms involved in this phenomenon. Our model predicts that stable memories will be formed in response to stimuli and that two such memories can "nudge closer together" in response to a simultaneous presentation of the two stimuli. Interestingly, as explained in Section 8, our results recall quite vividly the narrative of [5] about a related phenomenon in mouse olfaction.

How is it possible, by monitoring a few dozen neurons of a subject (out of many million in the MTL) and by presenting a hundred or so familiar images (as [9] and similar experiments have done), to identify several neurons consistently responding the images? About the only plausible explanation is that each image shown must excite a great number of neurons, and must do so quite consistently. This and many other experiments (see [3, 13] for reviews) confirm earlier theories and hypotheses going back to Hebb [8] that tokens of cognition (such as the Eiffel tower) are represented by assemblies of many excitatory neurons[1], often called *concept cells* [12]. These assemblies are stable, in the sense that in the short term they fire more-or-less consistently and as a whole with the same stimulus (absent new associations). They are therefore believed to be *densely connected* through many and strong synapses. Every time the corresponding cognitive entity is active in the brain, all these cells (more or less) fire. In fact, the experiment of [9] even suggests that these assemblies are *fluent* in that they can be changed dynamically in response to new experiences and associations.

Despite the emerging consensus that concept cell assemblies in the MTL are an important piece of the puzzle of memory and cognition, and simulation results verifying that assemblies can indeed emerge (see [11, 17], and [10] for related work on assembly *binding*) we are not aware[2] of theoretical models predicting the creation of cell assemblies, much less their association. Here we present such a model for the formation of assemblies in a recurrent network of *memory neurons,* in response to the spiking of a separate cell population of *sensory neurons* representing the sensory experience – we call such spiking *the presentation of the stimulus*.

We model the memory neurons as a directed $G_{n,p}$ graph, and the projection from the sensory neurons as a *bipartite, one-way* $G_{n,p}$ graph. Our model assumes that firing of neurons happens in discrete steps and synchronously. One key simplifying assumption of our model is that, at each step, *exactly* K *memory neurons fire*, namely the ones receiving at that instant the largest synaptic input. This is of course a strong simplifying assumption; it is intended to capture the way in which the firing thresholds of the excitatory memory neurons are regulated by inhibitory neurons (not modeled explicitly here), resulting in an equilibrium

[1] Naturally, two such sets can overlap – and a simple calculation suggests that they are likely to do so.
[2] Valiant's important and relevant theory [15] is discussed extensively in the sequel.

in which a relatively stable number of excitatory neurons end up firing.

We assume that the neural network is fixed; the only possible modification comes through *Hebbian plasticity:* if there is a synapse (directed edge in the graph) from i to j, and i and j fire in two consecutive steps in this order, then the *strength* of this synapse (and thus the strength of the firing signal it can transmit theretofore) increases. Our model, and especially our way of modeling inhibition, was inspired and informed by the discussion in [5] of a related phenomenon in the mouse brain, see the description in Section 8.

We prove formally that, in this model, when a stimulus is presented through the repeated firing of a set of sensory neurons representing the stimulus, a corresponding stable assembly of neurons can indeed be formed, with high probability. We also show that two such assemblies, once both formed in response to the presentations of two different stimuli at different times, can subsequently be modified by increasing their intersection in response to the simultaneous presentation of the corresponding stimuli (as happens in the "Obama at Eiffel" example). We first analyze a *linearized* version of the model, in which thresholds and saturation are ignored; we arrive at a dynamical system (Eq. 3.1), which we were able to solve through an equilibrium equation *in closed form.* We establish convergence under minimal assumptions, see Theorem 3.1. The analytical solution (see the statement of Theorem 3.1) recalls vividly the description of the related phenomenon in mouse olfaction [5], see Section 8.

We then proceed to analyze the strongly nonlinear dynamics of the full model. We prove that, here too, the description of [5] prevails: already after two steps of stimulus spiking, a set of cells has been selected comprised of two kinds, quite balanced in cardinality: cells that have strong projection from the stimulus population, and cells to which *those* project strongly (see Theorem 4.1(1)). Subsequent steps modify this assembly in rather limited ways (Theorem 4.1(2)).

Furthermore, simulations show that, if two stimuli A and B are presented in the order A, then B, then A + B (both stimuli spike), then A, then B, association happens: the assemblies responding to A and B change slightly so they intersect a little more. We prove an analytical result (Theorem 5.1) establishing that some fraction of the two assemblies will indeed migrate towards each other.

Synaptic Density of Assemblies and Valiant's Model

Ever since Hebb, assemblies were conjectured to be dense in synaptic connections. In fact, several of our proofs take advantage of the fact that synaptic density within the assembly being formed is markedly higher than random. The synaptic density of the formed assemblies is further enhanced in a more sophisticated random graph model that we call $G^{++}_{n,p}$, capturing experimental observations [14, 7] that the distribution of synaptic connections is biased towards *triangle completion* (see Section 6); in this model a combinatorial *birthday paradox* argument establishes that, for the parameter range of interest, assemblies are far more intraconnected than one would expect.

High synaptic density of assemblies is a major advantage when it comes to the maintenance of stability and consistency, but of course is a severe design burden at creation time[3]:

> *In a random graph, how do you select an induced subgraph that is much more dense than average?*

[3] According to Les Valiant (private communication to CHP, 2017) dense assemblies are "infinitely harder" to create than the *items* of Valiant's theory, discussed next.

This looks and feels like the intractable DENSEST K-SUBGRAPH problem [4]; in Section 6 we briefly discuss how our proposed mechanism can be abstracted as a rather apt algorithm for solving approximately this computational problem in a G_{np} graph, and how it compares with other known algorithms for it.

The high synaptic density of assemblies is one point of stark contrast of our theory of assemblies against L. G. Valiant's theory of *items*. More than two decades ago, Valiant articulated his important computational theory of cortex. He proposed an elegant model of cortex consisting of neurons connected through random synaptic connections and equipped with an automaton-like *vicinal* programming language. He proposed that tokens of human cognition, such as "Eiffel" and "Obama", are represented in cortex by sets of neurons called *items*, which can be combined, through vicinal algorithms, in ways akin to logical gates to form new items, and associations between such. Unlike our current theory of assemblies, an item in Valiant's theory is an arbitrary set of neurons with no particularly high connectivity. Perhaps the most critical difference between Valiant's theory and our current discussion of assemblies, and our main technical contribution, is this: Valiant's vicinal programming model allows a generous repertoire of elementary instructions (modifications of the parameters of neurons and synapses, such as threshold and synaptic strength as an arbitrary function of local state), whereas our model is far more minimalistic, relying only on the simple, and rather standard and broadly accepted, rule of Hebbian plasticity explained next, and a simplified rigorous treatment of inhibition.

2 Our Model

- There is a *memory area* M consisting of n neurons randomly connected through synapses according to the directed $G_{n,p}$ model (for every $i \neq j \in M$, the probability that there is a synapse (i, j) is p).
- There is a *sensory area* S, whose neurons project through synapses to the neurons in M according to the one-way bipartite $G_{n,p}$ model (for every $i \in S, j \in M$, the probability that there is a synapse (i, j) is p). A *stimulus* is a set of L neurons in S.
- *Firing of neurons* happens in discrete steps $1, 2, \ldots, t, \ldots$ and in synchrony. The *presentation of a stimulus* is the firing of the corresponding S neurons for a large number of subsequent steps (such repetitive firing is called *spiking*).
- Each synapse (i, j) (within M and from S to M) has a *strength* w_{ij}, initially 1.
- We denote the set of neurons in M that fire at time t by $F^t \cap M$ (the set F^t includes also neurons in S). This set is defined as follows: We first calculate for each neuron $i \in M$ its *synaptic input* $I_i^t = \sum_{j \in F^{t-1}} w_j i$, the sum of all synaptic weights of the synapses to i coming from neurons $j \in M \cup S$ that fired at the previous step. Then F^t is the set of K neurons in M with the largest I_i^t.

  Justification: In a simple model of the cortex, besides the *excitatory* neurons considered here there are also *inhibitory* neurons, whose role is, roughly speaking, to make sure that the number of firing excitatory neurons is not excessive. There are random synaptic connections between the two populations: Excitatory neurons project positively to inhibitory neurons, while inhibitory neurons project *negatively* on excitatory ones, increasing their firing threshold. Here we assume that, at the equilibrium of this random process, exactly K of the (excitatory) neurons in M will fire. This is obviously a strongly simplifying assumption, inspired by the narrative about inhibition in [5]. We are confident that a more detailed random graph model of the process described above would also result in a number of firing neurons that is strongly concentrated, by the law of large numbers, near

a value K; this would be an interesting extension of our work, which we intend to pursue.

- *Hebbian plasticity.* If there is a synapse (i, j) and it so happens that $i \in F^t$ and $j \in F^{t+1}$, then the weight of this synapse is increased by a small amount $\beta > 0$, say[4].

Indicative ranges of these parameters for the human MTL are these: $n = 10^7, p = 10^{-3}, K = L = 10^4$, and $\beta = 0.1$. In our simulations we use values such as $n = 10^3 - 10^4, p = 10^{-2}, K = L = 10^2$, and $\beta = 0.1$.

3 The Linearized System

We start with a simplified model that ignores the nonlinearity of thresholds (a useful and familiar mathematical simplification from the theory of artificial neural networks). The assembly creation process is then captured by the following dynamical system, where z_j is the stimulus projection strength at neuron j, $x_j(t)$ the activation probability of neuron j at time t and $W_{ij}(t)$ is the strength of the synapse ij at time t:

$$\begin{aligned} x_j(t+1) &= z_j + (W^T x(t))_j \\ W_{ij}(t+1) &= W_{ij}(t) + \beta x_i(t) x_j(t+1) \end{aligned} \tag{1}$$

In addition, the pre-synaptic weights at each neuron are normalized to sum to 1 after each weight update. We assume that initially W is a random adjacency matrix in $G_{n,p}$, and that at time 0, the activations $x(0)$ are set to 1 for a random subset of K neurons and 0 for the rest.

▶ **Theorem 1.** *With high probability over W and $x(0)$, the dynamics (1) converge linearly to the following equilibrium ($ij \in E$ denotes a synapse from i to j):*

$$x_j^* = z_j + \frac{\sum_{i:ij\in E} (x_i^*)^2}{\sum_{i:ij\in E} x_i^*}.$$

This equation captures and verifies in a rather striking way the description of a similar phenomenon in [5], see Section 8: the probability that a neuron joins the assembly has two components, the first being the size of the stimulus projection on the neuron, and the second a function of the corresponding probabilities of (recursively) its presynaptic neurons – a function that is monotonically increasing in the region of interest (most neurons have near-zero probabilities, while the rest have comparable probabilities).

Proof. An equilibrium activation x^* must satisfy

$$x^* = z + W^T x^* \text{ and so } x^* = (I - W^T)^+ z$$

where A^+ is the pseudo-inverse of A. The equilibrium weight matrix satisfies:

$$W_{ij}^* = \frac{W_{ij}^* + \beta x_i^* x_j^*}{\sum_{l:lj\in E} W_{lj}^* + \beta \sum_{l:lj\in E} x_l^* x_j^*}.$$

Therefore, using the fact that we normalize the incoming synaptic weights of each node:

$$W_{ij}^*(1 + \beta(\sum_{l:lj\in E} x_l^*)x_j^*) = W_{ij}^* + \beta x_i^* x_j^*$$

[4] Or instead multiplied by $1 + \beta$, or in either case up to a saturation level B. Our results are robust with respect to these popular variants of Hebbian plasticity.

which implies

$$W^*_{ij} = \frac{x^*_i}{\sum_{l:lj\in E} x^*_l}$$

and therefore

$$x^*_j = z_j + \frac{\sum_{i:ij\in E}(x^*_i)^2}{\sum_{i:ij\in E} x^*_i}.$$

The mathematically demanding part is the proof of convergence; this follows from Lemma 2 below (whose proof can be found in the Appendix): the first part shows progress in each step at a rate depending on the spectral gap of $W(t)$, and the second part shows that weight updates cannot slow down the convergence. In addition, convergence implies *stability*: If at a later time the same input signal is presented, the same probabilities of formation will be effected. The "high probability" clause of the theorem refers to the fact that the following events are highly probable for random W and $x(0)$: (a) W is irreducible, and (b) at each time t, the matrix $W(t)$ has non-negligible spectral gap, and therefore the lemma applies. ◀

▶ **Lemma 2.** *Let W be an $n\times n$ nonnegative, irreducible matrix and $z \in \mathbf{R}^n_+$ be a nonnegative vector.*

1. *The iteration $x(t+1) = z + W^T x(t)$ with random $x(0)$ satisfies*

$$\frac{\|W^T x(t+1)\|^2}{\|x(t+1)\|_2^2} > \frac{\|W^T x(t)\|^2}{\|x(t)\|_2^2}.$$

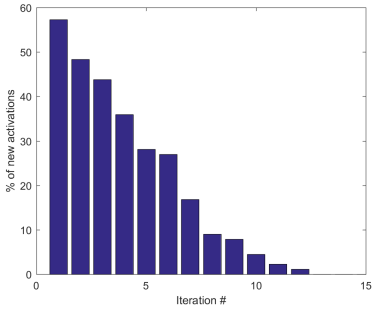
2. *Let the weight update rule (1) be applied repeatedly to synapse weights W for some $\beta > 0$ and current activation vector x. Then for each cell j, the vector w of incoming synaptic weights converges to a vector $\tilde{w}$ which satisfies*

$$\frac{\tilde{w}\cdot x}{\|\tilde{w}\|_2} \geq \frac{w\cdot x}{\|w\|_2}.$$

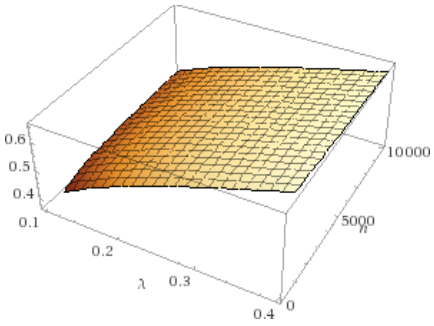
4 The Nonlinear System

Continuing to the full nonlinear model, our quantitative narrative of assembly formation (see Theorem 4.1 below) also recalls the key features in the description in [5]. Let $A(1), A(2), \ldots, A(t), \ldots$ denote the sets of K cells in the memory area firing at each discrete time step $t > 0$ during stimulus presentation. It is clear that $A(1)$ consists of the cells with the largest projection from the stimulus – we intuitively think of these cells as "born rich". At the next step, the theorem states that $A(2)$ contains a balanced mix of $A(1)$ cells, and cells that have strong *combined* projection from the stimulus *and* from $A(1)$. In experiments, the quantity λ capturing this balance is indeed reasonably far from 0 and 1 for the range of interest (see Figure 1b). Moreover, assuming powerful enough synaptic plasticity, subsequent sets $A(t)$ will converge rapidly to the final assembly A. Thus the theorem reasserts the interpretation of [5]. In the statement of the theorem, asymptotics are in terms of n, assuming that K is bounded from above and below by powers of n (e.g., $K = \sqrt{n}$).

▶ **Theorem 3.** *The following hold with high probability over random initial synaptic connections:*



(a) Fraction of new activations.



(b) Solutions to the β, λ equation.

Figure 1 Illustration of Theorem 3. The first plot is the relative size of the newly activated cells $A(t) \setminus A(t-1)$ in each iteration, with $n = 2000, K = L = 2\sqrt{n} = 89$.

1. *For all $t \geq 2$, we have*

$$|A(t) \cap A(1)| = (\lambda + o(1))K$$

for a constant $\lambda \in (0,1)$ depending on the synapse probability p, plasticity factor β, plasticity ceiling B and the ratio of assembly size to input size K/L.
2. *For large enough β and B, there is a $\bar{\lambda} < 1$ s.t. for all $t \geq 2$ we have*

$$|A(t) \setminus A(t-1)| \leq \left((1-\bar{\lambda})^t + o(1)\right) K.$$

Thus, the sequence of activated sets stabilizes rapidly, with the change to the previous set decaying geometrically with the number of steps. This can be seen in simulation, even for modest parameter values (see Figure 1a).

▶ **Lemma 4.** *Let $X \sim Bin(n,p)$. Then for $t > np$,*

$$\Pr(X \geq t) \leq \exp\left(-nH\left(p, \frac{t}{n}\right)\right)$$

where $H(p,q) = q\log(q/p) + (1-q)\log((1-q)/(1-p))$ is the entropy function.

For $p < 0.5$ and $np < t < 2np$, the above bound is at most $\exp(-(t-np)^2/np)$. We will use this in the proof of the main theorem.

Proof of Thm. 3. The set $A(1)$ consists of the K cells that receive the maximum total signal from the L stimulus cells. Since we model synaptic structure as a random graph with synapse

probability p, each cell j receives a signal $y_j = \sum_{i \in S} W_{ij}$ where S is the set of stimulus cells. This is the Bernoulli distribution $B(L, p)$, the sum of L independent Bernoulli random variables each with expectation p. The y_j's for different cells j are independent and thus the set $A(1)$ is exactly the K-cap of (K largest samples from) n independent copies of $B(L, p)$. This is the tail of the the Bernoulli $B(L, p)$ of probability K/n. A simple calculation using the Binomial tail bound (Lemma 4) gives us that the threshold for the K-cap is (very close) to

$$t_1 = pL + \sqrt{pL \ln(n/K)}$$

where n is the total number of cells in the MTL, and each cell in $A(1)$ receiving at least this much signal from the stimulus.

For the second step, the distribution of the signal to a cell depends on whether it is in $A(1)$ or not. A cell j *not* in $A(1)$, receives the signal of the input stimulus as well as the signal from cells in $A(1)$. We approximate this distribution by the Binomial $B(K + L, p)$, which ignores the conditioning that such a cell j was *not* in the K-cap of the initial Binomial; the latter conditioning can only reduce the probability that a cell not in $A(1)$ is a winner in the next round. For a cell j in $A(1)$, the signal from the external stimulus cells is fixed by the first step but amplified by a factor $(1 + \beta)$ due to plasticity; the signal from the K cells in $A(1)$ is random. So their signal comes from the distribution $(1 + \beta)t_1 + B(K, p)$. The threshold for the K-cap of this joint distribution is then close to

$$t_2 = (1 + \beta)t_1 + pK + \sqrt{pK \ln \frac{K}{\lambda K}} = p(K + L) + \sqrt{p(K + L) \ln \frac{n}{(1 - \lambda)K}}$$

where λ is the fraction of $A(2)$ that is also in $A(1)$. The equation above can be solved numerically for λ. For our range of interest, with $K = L = 2\sqrt{n}$, for λ in $[0.1, 0.4]$, the plasticity parameter β is in $[0.3, 0.6]$, and gets slightly smaller for larger graph size (see Figure 1b).

This establishes the intersection between $A(1)$ and $A(2)$ is at least a λ fraction with high probability. The first part of the theorem says that almost all of this intersection remains activated for all future time steps. To see this, note that after step 2, the weights from the input as well as from the all of $A(1)$ to cells in the intersection are increased again by a factor $1 + \beta$. These cells, which were already ahead, and are now further ahead. The rest of $A(1), A(2)$ are strictly inferior and the cells outside $A(1) \cup A(2)$ have gained no advantage at all, even ignoring the effect of the cap. The advantage of the intersection gets magnified with each iteration.

The general proof for both parts proceeds by induction on t. We claim inductively that, with high probability, any cell that is activated for a second time remains activated for all future steps. To see the inductive step, clearly such cells have an advantage over all other cells of factor of at least $(1 + \beta)$ for the signal coming from the external input cells, and for the signal coming from all such cells in the previous iteration (which are an increasing fraction of K, by the hypothesis). When such a cell is activated for the second time, it was already ahead of all other cells not in the activated set; this advantage is magnified by a factor of $1 + \beta$ for the signal from the input and from all such cells. Among the remaining cells, some will be activated for the second time and some for the first time. The relative fraction is bounded by $\bar{\lambda}$ via a calculation similar to the base case above – at each step the competition is between cells that have just been activated and received a $(1 + \beta)$ boost for the first time on part of their input signal (a diminishing fraction of K of such cells), and most of the n other cells that have not felt any plasticity yet. ◀

This result and its proof, as well as the equilibrium result for the linearized model, imply *consistency and stability* of the assemblies formed: If the same stimulus (or even a fairly similar stimulus, in some appropriate metric) is presented at a later time (even after certain limited changes in the weights of the circuit) then with high probability the same (more-or-less) cells will fire.

5 Association

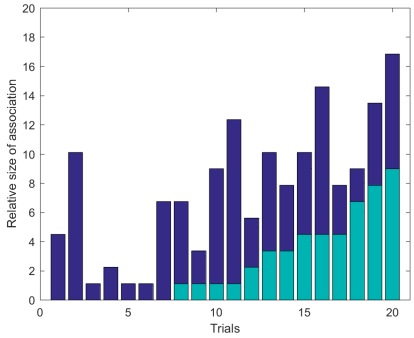
The experiment with the sequence of presentations A, B, A + B, A, B, described in the introduction shows that, after the joint presentation, the assemblies that respond to A and B "creep closer together," increasing their intersection to reflect the association between the two stimuli (see Figure 5.1). To understand and illustrate the underlying mechanism, we shall consider a stylized special case of assembly creation. In particular, we assume that plasticity is so intense that the assembly corresponding to stimulus A consists of only the cells receiving the K largest signal from stimulus A, and similarly for B. The advantage of this assumption is that it allows us to study the association phenomenon divorced from the subtleties of the proof of Theorem 4.1. Consider now the presentation of A + B; we can show the following (we assume for algebraic convenience that $K = L$):

▶ **Theorem 5.** *There is a $\mu > 0$ such that, with high probability, at least a μ fraction of the cells of the assembly for B will respond to the next presentation of A, and vice-versa.*

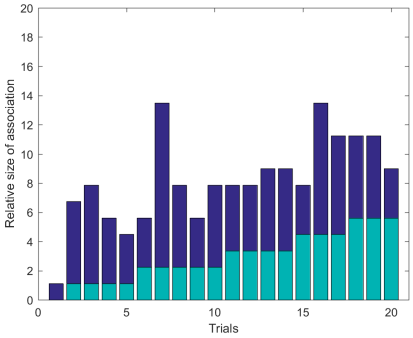
Proof. It is easy to see that, upon the presentation of A + B, the K cells that will fire consist of a fraction of the cells of assembly A and a fraction of the cells of assembly B, namely those that maximize the combined signal from A +B. The signal the cells of assembly A receives from stimulus A is very close to the K-cap $Lp(1+\sqrt{\frac{2\ln(n/K)}{Lp}})$, while from assembly B they only receive a binomial distribution with mean Lp (since those cells have not been selected for B). By symmetry, with high probability the number of cells from each of the two assemblies that fire will be very close to $\frac{K}{2}$. Let us call these sets $\frac{1}{2}$A and $\frac{1}{2}$B, respectively. We claim that all cells in $\frac{1}{2}$A receive a signal from stimulus A equal to $Lp(1+\sqrt{\frac{2\ln(n/K)}{Lp}})$ as before, plus a signal from stimulus B equal to $Lp(1+\sqrt{\frac{2\ln(n/K)}{Lp}}))$, since they were selected to be the cells in assembly A that are above the median with respect to received signal from B. Now notice that there are about pK^2 synapses between $\frac{1}{2}$A and $\frac{1}{2}$B. After the presentation of A + B, these synapses, as well as the synapses from stimulus A to $\frac{1}{2}$B, are boosted to plasticity saturation (since their endpoint cells spiked together for long enough).

If stimulus A is presented next, all cells of assembly A will fire at the first step. At the second step, however, they will have new competition (which was missing at assembly creation time) from the cells in $\frac{1}{2}$B which have, during the presentation of A + B, acquired strong synapses from stimulus A and assembly A. As in our analysis in the proof of Theorem 4.1, the threshold equation for μ becomes, after simplification $\frac{1}{2}+\sqrt{2\ln(n/k)}+\sqrt{2\ln(\frac{1}{1-\mu})} = \sqrt{\frac{2}{3}\ln(\frac{1}{2\mu})}$. The third term of the lhs is negligible for small μ, and the rhs can take over the other two terms by selecting μ appropriately small. ◀

This argument gives $\mu < 1\%$, which is conservative: the fractions of the two assemblies that intersect after association seem to amount to several percent (see Figure 2a). A similar statement can also be shown in the absence of the high plasticity assumption, by focusing on $A(1) \cap A(2)$ (which we know is a constant fraction of K), instead of the whole assembly.



(a) Association fraction in the standard model $G_{n,p}$.



(b) And in the extended model $G^{++}_{n,p}$.

Figure 2 Illustration of association *à la* Ison et al [9].The intersection of the assemblies for stimuli A and B, before and after a joint A+B presentation, over 20 trials with $n = 2000$ cells and assembly/stimulus sizes $K, L = \lceil 2\sqrt{n} \rceil = 89$. Trials sorted by size of initial overlap.

6 The Model $G^{++}_{n,p}$ and the Birthday Paradox

As we have seen, the assemblies created by the process analyzed here have higher density than random sets of cells, essentially due to the second step of the stimulus presentation (Theorem 4.1(1)). This effect is enhanced considerably if we adopt a more sophisticated random graph model. Experiments in [14, 7] and elsewhere suggest that synaptic connectivity in brain areas including parts of the MTL is not uniformly random, but *biased* towards *reciprocity* and *triangle completion*. Reciprocity means that, even though the overall density of edges remains p, conditioned on synapse ij being present, the probability of synapse ji is larger than p, perhaps between 3 and 5 times larger. Triangle completion means that, conditioned on the existence of synapses ij and ik, the probability of synapse jk is similarly larger.

Here we shall ignore reciprocity bias, and adopt a limited form of triangle completion bias: Again, the overall density of edges is p, except that, conditioned on the synapses ij and ik being present, *where i is a sensory cell and j, k are memory cells,* the probability of synapse jk is γp for some $\gamma > 1$. The reason we ignore reciprocity bias is because it seems to have only a small effect on our present focus; the reason we restrict triangle completion bias in the bipartite graph between the sensory and memory areas is because this part of triangle completion matters most, and also because there are formal difficulties involved in the precise definition of a generative model of random non-bipartite directed graphs with triangle completion bias. We call the resulting random graph model $G^{++}_{n,p}$.

Within this model, and for some reasonably broad range of parameters, we can show that there is substantial enhancement of the density of the assemblies. The underlying mathematical reason is the *birthday paradox*: Upon presentation of a stimulus, memory cells receive a signal of strength Lp on average – that is, on the average they each have Lp presynaptic stimulus cells. Consider two cells i and j in the memory area, and call them *siblings* if they have a common presynaptic cell in the stimulus; the chance that this is the case is clearly p^2L. Suppose however that we have identified a subset S of memory cells whose signal is known to be of strength at least αLp for some $\alpha > 1$; for example, within the initial cells A_1 of the assembly (recall Theorem 4.1 (1)), $\alpha = 1 + \sqrt{\frac{2\ln(n/K)}{pK}}$. Then the chance that two cells in S are siblings is increased to $\alpha^2 p^2 L$. For parameter values $n = 10^7, L = K = 10^3, p = 10^{-2}$ (all very much within the range of interest), the probability

of two cells of S being siblings is increased from .1 to about .8. Since in $G^{++}_{n,p}$ two siblings have enhanced probability of synaptic connection, say $\gamma = 4$, it follows that the synaptic density within the region A_1 of an assembly in $G^{++}_{n,p}$ will be more than 3 times its value in $G_{n,p}$. The rest of the assembly will also have increased interconnections, because of a similar birthday paradox argument, but in addition for the reasons obtaining in the proof of Theorem 4.1(1). Adding plasticity to the picture, we conclude that *assembly creation selects a set of cells with denser and stronger synaptic connections than random,* and achieves this in three ways: Through the second step of the creation process (Theorem 4.1(1)); through plasticity; and through triangle completion and the birthday paradox in the $G^{++}_{n,p}$ model.

Our experiments show that in $G^{++}_{n,p}$ assembly formation converges faster than in $G_{n,p}$, while the association effect of the previous section does not change much.

7 The Densest K-Subgraph Problem

Finding a set of K nodes with maximum density is an intractable problem to solve exactly or approximately in general graphs. The mechanism for assembly creation proposed here can be abstracted as an approximation heuristic for $G_{n,p}$ graphs:

```
1 Select a set S of λK nodes at random
2 Let T be the (1−λ)K nodes with highest number of edges from S
3 Return S ∪ T; optimize λ ∈ (0, 1)
```

It does fairly well: The expected density of the result is about $p(1 + \sqrt{\frac{\ln \frac{n}{K}}{2Kp}})$, compared to density p of a random set. The expected maximum density of a K-node subgraph of $G_{n,p}$ (achievable through exponential exhaustive search) turns out to be – after a calculation paralleling that in [2] (see also [6]) – the solution to this equation $Kx \ln p = Kx \ln x - 2 \ln n$, which turns out to be $d_{\max} = \frac{2 \log n}{KW(\frac{2 \log n}{Kp})}$, where $W(\cdot)$ is the Lambert W function[5].

A competing algorithm is the *Cliques* algorithm: Let $c = \frac{\log n}{\log \frac{1}{p}}$, the maximum size of a clique that we know how to produce in $G_{n,p}$:

`Repeat` $\frac{K}{c}$ `times: create a clique of size` c.

The resulting density is $p + \frac{c}{K}$, which does not compare well with the present algorithm.

Another competitor is the *Greedy* algorithm:

`Repeat` $n - K$ `times: delete the lowest degree node.`

Greedy is hopelessly sequential (and thus irrelevant to our concerns here), and it is not known how to estimate its performance in $G_{n,p}$, but in experiments it does perform better than AssemblyCreation. Naturally, AssemblyCreation performs much better in $G^{++}_{n,p}$, arguably a more realistic model of synaptic connectivity.

8 A Distant Mirror: The Mouse Piriform Cortex

The memories in the human MTL discussed here are often termed "abstract," and not without justification. During sensory processing, the stimulus is coded, over several stages e.g. in the visual cortex, in a distributed way. This coding spatially reflects the perceived reality, in that features of the perceived reality (such as edges, frequencies, color, motion) are processed and coded by neural systems specializing in those features. After the conclusion of sensory processing, a process may be initiated, possibly mediated and supervised by some attention mechanism, that creates a new, sparse representation of the perceived item in

[5] Many thanks to Cris Moore for help in this calculation

the MTL, in which any links to the perceived world have been severed through random projection; this is the sense of abstraction imputed above.

A simple instance of this phenomenon has been identified recently in a rather unexpected place, the piriform cortex of the mouse [5]. Odorant molecules excite olfactory receptors in the animal's nose specializing in that molecule, and the axons of those excite in turn a small area (glomerulus) in the olfactory bulb; here again, each glomerulus specializes in one odor out of many hundreds. Next, the odorant's glomerulus projects strongly to the piriform cortex, creating a seemingly uniformly random – "abstract," disconnected from the outside world – sparse representation of the odorant. Here is the prescient interpretation in [5] of their experimental findings about this latter phenomenon:

1. *An odorant may [cause] a small subset of [...] neurons [in the piriform cortex to fire].*
2. *This small fraction of [...] cells would then generate sufficient recurrent excitation to recruit a larger population of neurons.*
3. *The strong feedback inhibition resulting from activation of this larger population of neurons would then suppress further spiking.*
4. *In the extreme, some cells could receive enough recurrent input to fire [...] without receiving [initial] input.*

This narrative was an important inspiration for the present work, and the mathematical analysis of our model (Theorems 3.1 and 4.1) recalls it with rather striking fidelity.

9 Discussion and Further Research

We provide rigorous proof that, in a strongly simplified mathematical model of the brain, an assembly can emerge in response to spiking stimulus cells, will be exceptionally dense in synapses (a nontrivial algorithmic feat in a random network), and will fire consistently on future presentations of the stimulus; furthermore, upon a joint representation of two established stimuli, the assemblies will adapt by increasing their intersection (as has been observed in experiments). Despite the restrictions of our model, our probabilistic approach is quite robust, and we expect that several extensions can be obtained with further calculation. One such model would include a more realistic model of inhibition through a Gaussian synaptic input whose mean increases with excitatory activity (and no fixed K). Another extension would be to show robustness of assembly formation to perturbation of the stimulus, initial random excitatory activity, and noise. Also, it would be interesting to compare our results with simulations of biologically more realistic models, and to test experimentally if indeed assemblies in brains are more densely connected than random.

It would be interesting to see if the kind of mechanism hypothesized in this paper is present in other cognitive functions besides long term memory. One such is that of *visual invariants*, the mysterious ability by humans to identify, e.g., various rotations, postures, zooms, and occlusions of a familiar face, or even identify those visual images with a voice waveform or a string of characters. Our experiments show that, if two stimuli are presented together repeatedly, then the corresponding assemblies keep coming closer and closer; eventually they may become indistinguishable, and one can wonder if this mechanism cannot be part of the neural basis of invariants.

More ambitiously, what if *two* stimuli – or existing assemblies, encoding let us say to two parts of a sentence – are projected to another brain area (the same way a single stimulus is projected in the basic mechanism of this paper)? The assembly thus formed can be thought of as encoding a *Merge* [1] of the other two, that is, the new root of a syntax tree. Also, a mechanism akin to our assembly creation called *assembly pointer* has been studied recently

through computational experiments [10]. An assembly pointer creates a copy of an extant assembly in a different brain area – perhaps a copy of the assembly for the lexical element "give" in another brain area where verbs get ready for syntax (see [16] for recent experimental evidence suggesting such activities in various areas of the cortex). It would be exciting to explore whether variants of the proposed mechanism can be the basis of beginning to understand how language is implemented in the Brain.

Acknowledgment. An inspiring discussion with Richard Axel on assemblies in the mouse's piriform cortex is gratefully acknowledged.

References

1 Robert C Berwick and Noam Chomsky. *Why only us: Language and evolution*. MIT Press, 2016.

2 Béla Bollobás. Random graphs. In *Modern Graph Theory*, pages 215–252. Springer, 1998.

3 G Buzsaki. Neural syntax: cell assemblies, synapsembles, and readers. *Neuron*, 68(3), 2010.

4 Uriel Feige, David Peleg, and Guy Kortsarz. The dense k-subgraph problem. *Algorithmica*, 29(3):410–421, 2001.

5 Kevin M Franks, Marco J Russo, Dara L Sosulski, Abigail A Mulligan, Steven A Siegelbaum, and Richard Axel. Recurrent circuitry dynamically shapes the activation of piriform cortex. *Neuron*, 72(1):49–56, 2011.

6 Alan Frieze and Michał Karoński. *Introduction to random graphs*. Cambridge University Press, 2015.

7 S Guzman, A J Schlö gl, M Frotscher, and P Jonas. Synaptic mechanisms of pattern completion in the hippocampal ca3 network. *Science*, 353(6304):1117–1123, 2016.

8 Donald Olding Hebb. *The organization of behavior: A neuropsychological theory*. Wiley, New York, 1949.

9 Matias J Ison, Rodrigo Quian Quiroga, and Itzhak Fried. Rapid encoding of new memories by individual neurons in the human brain. *Neuron*, 87(1):220–230, 2015.

10 Robert Legenstein, Christos H Papadimitriou, Santosh Vempala, and Wolfgang Maass. Assembly pointers for variable binding in networks of spiking neurons. *arXiv preprint arXiv:1611.03698*, 2016.

11 A. Litwin-Kumar and B. Doiron. Formation and maintenance of neuronal assemblies through synaptic plasticity. *Nature communications*, 5, 2014.

12 Rodrigo Quian Quiroga. Concept cells: the building blocks of declarative memory. *Nature Reviews Neurosci.*, 13(8):587–597, 2012.

13 Rodrigo Quian Quiroga. Neuronal codes for visual perception and memory. *Neuropsychologia*, 83:227–241, 2016.

14 S. Song, P. J. Sjöström, M. Reigl, S. Nelson, and D. B. Chklovskii. Highly nonrandom features of synaptic connectivity in local cortical circuits. *PLoS Biology*, 3(3):e68, 2005.

15 Leslie G. Valiant. A neuroidal architecture for cognitive computation. *J. ACM*, 47(5):854–882, 2000.

16 Emiliano Zaccarella, Lars Meyer, Michiru Makuuchi, and Angela D. Friederici. Building by syntax: The neural basis of minimal linguistic structures. *Cerebral Cortex*, 27(1):411–421, 2017. doi:10.1093/cercor/bhv234.

17 F. Zenke, E. J. Agnes, and W. Gerstner. Diverse synaptic plasticity mechanisms orchestrated to form and retrieve memories in spiking neural networks. *Nature communications*, 6, 2015.

10 Appendix

Proof of Lemma 2. Let $W = \sum_i \sigma_i u_i v_i^T$ be the SVD of W, and $v = \sum_i \alpha_i v_i$. Without loss of generality, assume $\sum_i \alpha_i^2 = 1$. Then for any integer k,

$$\|W^k v\|^2 = v^T (W^T)^k W^k v = \sum_i \alpha_i^2 \sigma_i^{2k} = \mathsf{E}(X^{2k})$$

where the random variable X is equal to σ_i with probability α_i^2. Then $x(t)$ is proportional to $(v + Wv + \dots W^t v)$ and the desired inequality can be stated as follows:

$$\frac{v^T(I + W + \dots + W^{t+1})^T W^T W (I + W + \dots + W^{t+1})v}{v^T(I + W + \dots + W^{t+1})^T (I + W + \dots + W^{t+1})v}$$
$$\geq \frac{v^T(I + W + \dots + W^{t})^T W^T W (I + W + \dots + W^{t})v}{v^T(I + W + \dots + W^{t})^T (I + W + \dots + W^{t})v}$$

which is equivalent to:

$$\mathsf{E}(X^2(1 + X + \dots + X^{t+1})^2)\mathsf{E}((1 + X + \dots + X^t)^2)$$
$$\geq \mathsf{E}(X^2(1 + X + \dots + X^{t})^2)\mathsf{E}((1 + X + \dots + X^{t+1})^2)$$

or

$$\mathsf{E}(\frac{X^2(1 - X^{t+2})^2}{(1 - X)^2})\mathsf{E}(\frac{(1 - X^{t+1})^2}{(1 - X)^2}) \geq \mathsf{E}(\frac{X^2(1 - X^{t+1})^2}{(1 - X)^2})\mathsf{E}(\frac{(1 - X^{t+2})^2}{(1 - X)^2}).$$

Define $f_1, f_2, g_1, g_2 : \mathbf{R}_+ \to \mathbf{R}_+$ to be each of the functions inside the expectations in the order above, so that the inequality is

$$\mathsf{E}(f_1(X))\mathsf{E}(f_2(X)) \geq \mathsf{E}(g_1(X))\mathsf{E}(g_2(X)).$$

Observe that for any X, we have

$$f_1(X) f_2(X) = g_1(X) g_2(X).$$

Moreover, for any X, Y, we claim that

$$f_1(X) f_2(Y) + f_1(Y) f_2(X) \geq g_1(X) g_2(Y) + g_1(Y) g_2(X).$$

For our choice of functions, this is

$$\frac{X^2(1 - X^{t+2})^2}{(1 - X)^2}\frac{(1 - Y^{t+1})^2}{(1 - Y)^2} + \frac{Y^2(1 - Y^{t+2})^2}{(1 - Y)^2}\frac{(1 - X^{t+1})^2}{(1 - X)^2}$$
$$\geq \frac{X^2(1 - X^{t+1})^2}{(1 - X)^2}\frac{(1 - Y^{t+2})}{(1 - Y)^2} - \frac{Y^2(1 - Y^{t+1})^2}{(1 - Y)^2}\frac{(1 - X^{t+2})^2}{(1 - X)^2}$$

which is equivalent to

$$X^2(1 - X^{t+2})^2(1 - Y^{t+1})^2 + Y^2(1 - Y^{t+2})^2(1 - X^{t+1})^2$$
$$\geq X^2(1 - X^{t+1})^2(1 - Y^{t+2})^2 + Y^2(1 - Y^{t+1})^2(1 - X^{t+2})^2$$

or

$$(X^2 - Y^2)\frac{(1 - X^{t+2})^2}{(1 - Y^{t+2})^2} \geq (X^2 - Y^2)\frac{(1 - X^{t+1})^2}{(1 - Y^{t+1})^2}$$

which is always true. Therefore, we have

$$\begin{aligned}
&\mathsf{E}(f_1(X))\mathsf{E}(f_2(X)) - \mathsf{E}(g_1(X))\mathsf{E}(g_2(X)) \\
&= \sum_i \alpha_i f_1(X_i) \sum_i \alpha_i f_2(X_i) - \sum_i \alpha_i g_1(X_i) \sum_i \alpha_i g_2(X_i) \\
&= \sum_{i<j} \alpha_i \alpha_j (f_1(X_i) f_2(X_j) + f_1(X_j) f_2(X_i) - g_1(X_i) g_2(X_j) - g_1(X_j) g_2(X_i)) \\
&\quad + \sum_{i=j} \alpha_i^2 (f_1(X_i) f_2(X_i) - g_1(X_i) g_2(X_i)) \\
&\geq 0.
\end{aligned}$$

Moreover this holds with strict inequality unless $X = Y$, i.e., two of the singular values of W are equal. Thus the rate of convergence is at least the minimum singular value gap of W.

For the second part, note that the vector w consists of only the nonzero synapses into some cell j, and so the update rule on each synapse can be written as $w_{ij} = w_{ij} + \beta_j x_i$ where $\beta_j = \beta x_j$. Treating w and x as indexed only by i, the cells with synapses to a fixed j, we write

$$\begin{aligned}
\left(\frac{\tilde{w} \cdot x}{\|\tilde{w}\|_2}\right)^2 &= \frac{((w + \beta_j x) \cdot x)^2}{\|w + \beta_j x\|_2^2} \\
&= \frac{(w \cdot x)^2 + \beta_j^2 \|x\|_2^4 + 2\beta_j (w \cdot x)\|x\|_2^2}{\|w\|_2^2 + \beta_j^2 \|x\|_2^2 + 2\beta_j (w \cdot x)}
\end{aligned}$$

and need to show that this is greater than

$$\frac{(w \cdot x)^2}{\|w\|_2^2}.$$

Comparing, the inequality becomes

$$((w \cdot x)^2 + \beta_j^2 \|x\|_2^4 + 2\beta_j (w \cdot x)\|x\|_2^2)\|w\|_2^2 > (w \cdot x)^2 (\|w\|_2^2 + \beta_j^2 \|x\|_2^2 + 2\beta_j (w \cdot x))$$

which is implied by the Cauchy-Schwartz inequality:

$$\|w\|_2^2 \|x\|_2^2 \geq (w \cdot x)^2$$

applied twice. ◀

Toward a Theory of Markov Influence Systems and their Renormalization*

Bernard Chazelle

Department of Computer Science, Princeton University, USA
chazelle@cs.princeton.edu

Abstract

Nonlinear Markov chains are probabilistic models commonly used in physics, biology, and the social sciences. In *Markov influence systems* (*MIS*), the transition probabilities of the chains change as a function of the current state distribution. This work introduces a renormalization framework for analyzing the dynamics of *MIS*. It comes in two independent parts: first, we generalize the standard classification of Markov chain states to the dynamic case by showing how to "parse" graph sequences. We then use this framework to carry out the bifurcation analysis of a few important *MIS* families. In particular, we show that irreducible *MIS* are almost always asymptotically periodic. We also give an example of "hyper-torpid" mixing, where a stationary distribution is reached in super-exponential time, a timescale that cannot be achieved by any Markov chain.

1998 ACM Subject Classification G.2.2 Graph Theory (F.2.2)

Keywords and phrases Markov influence systems, nonlinear Markov chains, dynamical systems, renormalization, graph sequence parsing

Digital Object Identifier 10.4230/LIPIcs.ITCS.2018.58

1 Introduction

Nonlinear Markov chains are popular probabilistic models in the natural and social sciences. They are commonly used in interacting particle systems, epidemic models, replicator dynamics, mean-field games, etc. [8, 12, 13, 15, 18]. They differ from the linear kind by allowing transition probabilities to vary as a function of the current state distribution.[1] For example, a traffic network might update its topology and edge transition rates adaptively to alleviate congestion. The traditional formulation of these models comes from physics and relies on the classic tools of the trade: stochastic differential calculus, McKean interpretations, Feynman-Kac models, Fokker-Planck PDEs, etc. [3, 5, 13, 18]. These techniques assume all sorts of symmetries that are typically absent from the "mesoscopic" scales of natural algorithms. They also tend to operate at the thermodynamic limit, which rules out genuine agent-based modeling. Our goal is to initiate a theory of discrete-time Markov chains whose topologies vary as a function of the current probability distribution. Of course, the entire theory of finite Markov chains

* The Research was sponsored by the Army Research Office and the Defense Advanced Research Projects Agency and was accomplished under Grant Number W911NF-17-1-0078. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Office, the Defense Advanced Research Projects Agency, or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation herein.

[1] The systems are Markovian in that the future depends only on the present: in this case the current state distribution rather than the single state presently visited.

should be recoverable as a special case. Our contribution comes in two parts (of independent interest), which we discuss informally in this introduction.

Renormalization

The term refers to a wide-reaching approach to complex systems that originated in quantum mechanics and later expanded to statistical mechanics and dynamics. Whether in its exact or coarse-grained form, the basic idea is intuitively appealing: break down a complex system into a hierarchy of simpler parts. The concept seems so simple—isn't it what divide-and-conquer is all about?—one can easily be deceived and miss the point. When we slap a dynamics on top of the system (think of interacting particles moving about) then the hierarchy itself creates its own dynamics between the layers. This new "renormalized" dynamics can be entirely different from the original one. Crucially, it can be both easier to analyze and more readily expressive of global properties. For example, second-order phase transitions in the Ising model correspond to fixed points of the renormalized dynamics.[2]

What is the relation to Markov chains? You may have noticed how texts on the subject often dispatch absorbing chains quickly before announcing that from then on all chains will be assumed to be irreducible (and then, usually a few pages later, ergodic). This is renormalization at work! Indeed, although rarely so stated, the standard classification of the states of a Markov chain is a prime example of exact renormalization. Recall that the main idea is to express the chain as an acyclic directed graph, its *condensation*, whose vertices correspond to the strongly connected components. This creates a two-level hierarchy: a tree with a root (the condensation) and its children (the strongly connected components). Now get the chain going and watch what happens at the root: the probability mass flows entirely into the sinks of the condensation. Check the leaves of the tree for a detailed understanding of the motion. The renormalized dynamics (visible only in the condensation) has an attracting manifold that tells much of the story. If the story lacks excitement it is partly because the hierarchy is flattish: only two levels. Time-varying Markov chains, as we shall soon see, do not suffer from that problem.

Consider an infinite sequence $(g_k)_{k>0}$ of digraphs over the same set of vertices. A *temporal* random walk is defined in the obvious way by picking a starting vertex in g_1, moving to a random neighbor, and then repeating this step in g_2, g_3, etc [6, 7, 14, 19]. The walk is called temporal because it traverses one edge from g_t at time $t = 1, 2, \ldots$. How might one go about classifying the states of this "dynamic" Markov chain? Repeating the condensation decomposition at each step makes no sense, as it carries zero information about the temporal walks. The key insight is to monitor when and where temporal walks are *extended*. The *cumulant* graph collects all extensions and, when this process stalls, reboots the process while triggering a deepening of the hierarchy. To streamline this process, we define a grammar with which we can parse the sequence $(g_k)_{k>0}$. The (exact) renormalization framework introduced in this work operates along two tracks: time and network. The first track summarizes the past to anticipate the future while the second one clusters the graphs hierarchically. The method is very general and we expect it to be used elsewhere.

[2] The idea is very powerful: Ken Wilson won the 1982 Nobel prize in physics and Artur Avila the 2014 Fields medal for their (very different) breakthroughs in the use of renormalization: finding new critical exponents; proving the weak mixing of interval exchange transformations, etc.

Markov influence systems

All finite Markov chains oscillate periodically or mix to a stationary distribution. The key fact about their dynamics is that the timescales never exceed a single exponential in the number of states. Allowing the transition probabilities to fluctuate over time at random does not change that basic fact [1, 9, 10]. Markov influence systems are an entirely different beast. Postponing formal definitions, let us think of an *MIS* for now as a dynamical system defined by iterating the map $f : \mathbf{x}^\top \mapsto \mathbf{x}^\top S(\mathbf{x})$, where $\mathbf{x}$ is a probability distribution and $S(\mathbf{x})$ is a stochastic matrix that is piecewise-constant as a function of $\mathbf{x}$. We assume that the discontinuities are linear (ie, hyperplanes). The assumption is not restrictive in any meaningful sense: we explain why with a simple example.

Consider a random variable ξ over the distribution $\mathbf{x}$ and fix two stochastic matrices A and B. Define $S(\mathbf{x}) = A$ (resp. B) if $\mathrm{var}_{\mathbf{x}}\, \xi > 1$ (resp. else); in other words, the Markov chain picks one of two stochastic matrices at each step depending on the variance of ξ with respect to the current state distribution $\mathbf{x}$. This clearly violates our assumption because the discontinuity is quadratic in $\mathbf{x}$; hence nonlinear. This is not an issue because we can linearize the variance: here, we begin with the identity $\mathrm{var}_{\mathbf{x}}\, \xi = \sum_{i,j} (\xi_i - \xi_j)^2 x_i x_j$ and the fact that $\mathbf{y} := (x_i x_j)_{i,j}$ is a probability distribution. We form the Kronecker square $T(\mathbf{x}) = S(\mathbf{x}) \otimes S(\mathbf{x})$ and lift the system into the $(n^2 - 1)$-dimensional unit simplex to get a brand-new *MIS* defined by the map $\mathbf{y} \mapsto T(\mathbf{y})$. We now have linear discontinuities. This same type of tensor lift can be used to linearize any algebraic constraints.[3] Using ideas from [4], one can go much further than that and base the step-by-step Markov chain selection on the outcome of any first-order logical formula we may fancy (with the x_i's as free variables).[4] What all of this shows is that the assumption of linear discontinuities is not restrictive.

We prove that irreducible *MIS* are almost always asymptotically periodic. (This assumes that $S(\mathbf{x})$ forms an irreducible chain for each $\mathbf{x}$.) We extend this result to larger families of Markov influence systems. We also give an example of "hyper-torpid" mixing: an *MIS* that converges to a stationary distribution in time equal to a tower-of-twos in the size of the chain. The emergence of timescales far beyond the reach of standard Markov chains is a distinctive feature of Markov influence systems. We note that the long-time horizon analysis of general systems is still open.

Some intuition

Is there a quick, intuitive explanation why the analysis of *MIS* should require all of that renormalization machinery? Of course, perhaps it does not and future work will show how to bypass it. But the specific challenges raised by the model are easy to state. The first hurdle is that Markov influence systems are not globally contractive. Worse, the eigenspaces over which they are not may be constantly changing over time. It is this spectral incoherence that renormalization attempts to "tame." To see why this has a strong graph-theoretic flavor, consider the fact that the stationary distributions may change at each time step and so can their number. The key insight is that these changes can be read off the topology of the graph: for example, the number of sinks in the condensation is precisely equal to the dimension of the principal eigenspace. Renormalization can thus be seen as an attempt to

[3] This requires making the polynomials over x_i homogeneous, which we can do by using the identity $\sum_i x_i = 1$.

[4] The key fact behind this result is that the first-order theory of the reals is decidable by quantifier elimination. This allows us to pick the next stochastic matrix at each time step on the basis of the truth value of a Boolean logic formula with arbitrarily many quantifiers. See [4] for details.

restore coherence to an ever-changing spectral landscape via a dynamic hierarchy of graphs, subgraphs, and homomorphs.

The bifurcation analysis at the heart of our analysis of Markov influence systems follows an approach commonly used in dynamics [2, 16, 21]: the idea is develop a notion of "general position" in order to bound the growth rate of the induced symbolic dynamics. The root of the problem is a clash between order and randomness. (This is the same conflict that arises between entropy and energy in statistical mechanics.) All Markov chains are attracted to a limit cycle (ie, order). Changing the chain at each step introduces pseudorandomness into the process (ie, disorder) and the question is to know which one of the two "forces" of order or disorder will prevail. The conflict is mediated by introducing a perturbation parameter and locating its critical values. We show that, in this case, the critical region forms a Cantor set of Hausdorff dimension strictly less than 1.

Previous work

There is a growing body of literature on dynamic graphs [3, 14, 17, 19] and their random walks [1, 6, 7, 8, 12, 9, 10, 15]. By contrast, as mentioned earlier, most of the research on nonlinear Markov chains has been done within the framework of stochastic differential calculus. The closest analog to the *MIS* model are the diffusive influence systems we introduced in [4]. The relation is interesting. Random walks and diffusion are dual processes that coincide only when the underlying operator is self-adjoint (which is not the case here). As a rule of thumb, diffusion is easier to analyze because even in a changing medium the constant function is always a principal eigenfunction. As a result, a diffusion model can converge to a fixed point while its dual Markov process does not. The reason the dynamics is so different is that, as has long been known [20], multiplying stochastic matrices from the right is harder than from the left.[5] Our renormalization scheme is new, but the idea of parsing graph sequences is not. We introduced it in [4] as a way of tracking the flow of information across changing graphs. The parsing method we discuss here is entirely different, however: being *topological* rather than *informational*, it is vastly more general and, we believe, likely to be useful in other applications of dynamic networks.

2 How to Parse a Graph Sequence

Throughout this work, a *digraph* refers to a directed graph with vertices in $[n]$ and a self-loop at each vertex.[6] We denote digraphs by lower-case letters (g, h, etc) and use boldface symbols for sequences. A *digraph sequence* $\mathbf{g} = (g_k)_{k>0}$ is an ordered, finite or infinite, list of digraphs over the vertex set $[n]$. The digraph $g_i \times g_j$ consists of all the edges (x, y) such that there exist at least an edge (x, z) in g_i and another one (z, y) in g_j. The operation $\times$ is associative.[7] We define the *cumulant* $\prod_{\leq k} \mathbf{g} = g_1 \times \cdots \times g_k$ and write $\prod \mathbf{g} = g_1 \times g_2 \times \cdots$ for finite $\mathbf{g}$. The cumulant indicates all the pairs of vertices that can be joined by a temporal walk of a given length. We need additional terminology:

[5] For example, consider the product AB of two stochastic matrices, where $\text{rank}(B) = 1$. We have $AB = B$ whereas BA can be any old stochastic matrix of rank 1.
[6] The graphs and digraphs (words we use interchangeably) have no multiple edges and $[n] := \{1, \ldots, n\}$.
[7] The sign is meant to highlight the connection with the multiplication of incidence matrices.

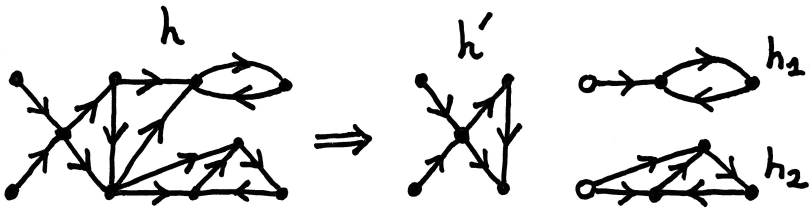


Figure 1 The decomposition of h into its stem h' and its petals h_1, h_2.

- *Transitive front of g*: An edge (x, y) of a digraph g is *leading* if there is u such that (u, x) is an edge of g but (u, y) is not.[8] The non-leading edges form a subgraph $tf(g)$, called the transitive front of g. We omit the (easy) proof that the transitive front is indeed transitive, ie, if (x, y) and (y, z) are edges of $tf(g)$ then so is (x, z). Given two graphs g, h over the same vertex set, we write $g \preceq h$ if all the edges of g are in h (with strict inclusion denoted by the symbol $\prec$). Because of the self-loops, $g, h \preceq g \times h$. We easily check that the transitive front of g is the (unique) densest graph h such that $g \times h = g$.
- *Subgraphs and contractions*: Given two digraphs g, h with vertex sets $V_g \supseteq V_h$, we denote by $g_{|\,h}$ the subgraph of g induced by V_h. Pick $U \subseteq V_h$ and contract all these vertices into a single one. By abuse of notation, we still designate by $g_{|\,h}$ the graph derived from g by first taking the subgraph induced by V_h and then contracting the vertices of U; note that the notation $g_{|\,(V_h,U)}$ would be more accurate but it will not be needed. Given a sequence $\mathbf{g} = (g_k)_{k>0}$, we use the shorthand $\mathbf{g}_{|\,h}$ for $(g_{k\,|\,h})_{k>0}$. Finally, $\mathbb{K}$ denotes the set of all complete digraphs (of any size) with self-loops, while $\mathbb{K}\otimes 1$ consists of the complete digraphs with an extra vertex pointing to all the others unidirectionally.[9]
- *Stem decomposition of h*: The strongly connected components of a graph h form, by contraction, an acyclic digraph called its *condensation*. Let $V_1, \ldots, V_\ell$ be the vertex sets from $[n]$ corresponding to the ℓ sinks of the condensation.[10] The remaining vertices of h induce a subgraph h' called the *stem* of h. For each $i \in [\ell]$, the *petal* h_i is the subgraph induced by V_i if no vertex outside V_i links to it; else h_i is the subgraph induced by V_i and h', with all the vertices of h' subsequently contracted into a single vertex and the multiple edges removed (fig.1).

The parser

The parse tree of a (finite or infinite) graph sequence $\mathbf{g}$ is a rooted tree whose leaves are associated with $g_1, g_2, \ldots$ from left to right; each internal node assigns a syntactical label to the subsequence $g_i, \ldots, g_j$ formed by the leaves of its subtree. The purpose of the parse tree is to monitor the formation of new temporal walks as time progresses. How to do that begins with the observation that the cumulant $\prod_{\leq k} \mathbf{g}$ is monotonically nondecreasing.[11] If the increase was strict at each step then the parse tree would be trivial: each graph of $\mathbf{g}$ would appear as a separate leaf with the root as its parent. Of course, the increase cannot

[8] For example, $tf(x \to y \to z)$ is the graph over x, y, z with the single edge $x \to y$ (and the three self-loops.) If g is transitive, then $tf(g) = g$. The transitive front of a directed cycle has no edges besides the self-loops.
[9] For example, ignoring self-loops, $\{(x, y), (y, x)\} \in \mathbb{K}$ and $\{(x, y), (y, x), (z, x), (z, y)\} \in \mathbb{K}\otimes 1$.
[10] These are the vertices with no outgoing edges: there is at least one of them; hence $\ell > 0$.
[11] All references to graph ordering are relative to $\preceq$.

go on forever. How to deal with time intervals within which the cumulant is "stuck" is the whole point of parsing: Short answer: proceed recursively. The grammar consists of only two pairs of productions, (1a,1b) and (2a,2b).

1. TIME RENORMALIZATION Let m be the smallest index k at which $\prod_{\leq k} \mathbf{g}$ achieves its maximal value; write $\mathbf{g}_l = (g_k)_{k<m}$, $\mathbf{g}_r = (g_k)_{k>m}$, and $h = tf(\prod_{\leq m} \mathbf{g})$. The two productions below cluster time into the relevant intervals.
   a. *Transitivization.* Using a parenthesis system to express the parse tree, the first production supplies the root with at most three children:

$$\mathbf{g} \longrightarrow \big(\, \mathbf{g}_l \,\big)\, g_m \,\big(\, \mathbf{g}_r \triangle h \,\big)\,, \tag{1a}$$

   where h is transitive, and $\mathbf{g}_l$ or $\mathbf{g}_r$ (or both) may be the empty sequence $\varnothing$. If $\mathbf{g}_l \neq \varnothing$, then $\prod \mathbf{g}_l \prec \prod_{\leq m} \mathbf{g} = \prod \mathbf{g}$. The right sibling of $(\mathbf{g}_l)$ is the terminal symbol g_m (a leaf of the parse tree) followed by $\mathbf{g}_r \triangle h$. The annotation $\triangle h$ indicates that $\prod \mathbf{g}_r \preceq h$ and that h will "guide" the parsing of $\mathbf{g}_r$.[12] Observe that h is available when needed but not earlier. This ensures that the parsing is of the *LR type*, meaning that it can be carried out bottom-up in a single left-to-right scan.[13]
   b. *Cumulant completion.* We parse $\mathbf{g} \triangle h$ in the special case where h is in $\mathbb{K}$ or $\mathbb{K}\otimes 1$. Recall that the notation $\triangle$ implies that $\prod \mathbf{g} \preceq h$. Partition the sequence $\mathbf{g}$ into minimal subsequences $\mathbf{g}_k\, g_{m_k}$ such that $\prod \mathbf{g}_k \prec h = \big(\prod \mathbf{g}_k\big) \times g_{m_k}$:

$$\mathbf{g} \triangle h \longrightarrow \big(\mathbf{g}_1\big)\, g_{m_1} \,\big(\mathbf{g}_2\big)\, g_{m_2} \cdots \tag{1b}$$

   The list on the right-hand side could be finite or infinite; if finite, it could be missing the final g_{m_k}. This production is the one doing the heavy lifting in that it establishes a bridge between renormalization and Lyapunov exponents.
2. NETWORK RENORMALIZATION Two productions parse the rightmost term in (1a) by recursively breaking down the graph into clusters. This is done either by carving out subgraphs or taking homomorphs. In both cases, it is assumed that $\prod \mathbf{g} \preceq h$ and that h is transitive but *not* in $\mathbb{K}$ or $\mathbb{K}\otimes 1$.
   a. *Decoupling.* If the number of connected components $h_1, \ldots, h_k$ of h exceeds one, then[14]

$$\mathbf{g} \triangle h \longrightarrow \big(\, \mathbf{g}_{\,|\,h_1} \triangle h_1 \,\big) \parallel \cdots \parallel \big(\, \mathbf{g}_{\,|\,h_k} \triangle h_k \,\big)\,. \tag{2a}$$

   In terms of the parse tree, the node has k children that model processes operating in parallel. Intuitively, the production decouples the system into the subsystems formed by the components. As we show below, this does not always imply the independence of the respective dynamics, however.
   b. *One-way coupling.* If the undirected version of h has a single connected component, we use its stem decomposition $h', h_1, \ldots, h_\ell$ to cluster the digraphs of $\mathbf{g}$:

$$\mathbf{g} \triangle h \longrightarrow \big(\, \mathbf{g}_{\,|\,h'} \triangle h' \,\big) \;\Vdash\; \Big\{\, \big(\, \mathbf{g}_{\,|\,h_1} \triangle h_1 \,\big) \parallel \cdots \parallel \big(\, \mathbf{g}_{\,|\,h_\ell} \triangle h_\ell \,\big) \,\Big\}\,. \tag{2b}$$

[12] By definition of g_m, no temporal walk from $\mathbf{g}_r$ can extend one from $(g_k)_{k\leq m}$. This shows that $\big(\prod_{\leq m} \mathbf{g}\big) \prod \mathbf{g}_r = \prod_{\leq m} \mathbf{g}$; hence $\prod \mathbf{g}_r \preceq h$.

[13] This is usually a requirement for the bifurcation analysis. If not for that, we could use the simpler production $\mathbf{g} \to \mathbf{g} \triangle h$, instead.

[14] This refers to the subgraphs of h induced by each one of the vertex subsets of the connected components of the undirected version of h.

Since h is neither in $\mathbb{K}$ nor in $\mathbb{K}\otimes 1$, its stem and petals both exist (with $\ell > 0$). The assumed transitivity of h implies that each $h_i \in \mathbb{K}\otimes 1$. We iterate the production if h' is neither in $\mathbb{K}$ nor in $\mathbb{K}\otimes 1$. System-wise, the symbol $\Vdash$ indicates the direction of the information flow. None flows into $\mathbf{g}_{\mid h'} \triangle h'$, so its dynamics is decoupled from the rest. Such decoupling does not hold for the petals, so it is one-way. This allows us to renormalize the stem into a single vertex for the purposes of the petals: the common 1 in all the instances of $\mathbb{K}\otimes 1$. In terms of the parse tree, the nodes has $\ell + 1$ children that operate in parallel, with the last ℓ of them collecting information from the first one.

Network renormalization exploits the fact that the information flowing across the system might get stuck in portions of the graph for some period of time: we cluster the graph when and where this happens. Sometimes only time renormalization is possible. Consider the infinite sequence $(g_k)_{k>0}$, where $g_k = h_{k \,(\mathrm{mod}\, n)}$ and, for $k = 1, \ldots, n$, h_k consists of the graph at the vertices and edges from k to all $n-1$ other vertices: the cumulant never ceases to grow until it reaches $\mathbb{K}$, at which point the process repeats itself; the parsing involves n applications of (1a) with $\mathbf{g}_r = \varnothing$, followed by infinitely many calls to (1b). There is no network renormalization. Quite the opposite, the case of an infinite single-graph sequence features abundant network renormalization (fig.2).

The depth of the parse tree

It is easily verified that cumulants $\prod \mathbf{g}$ lose at least one edge from parent to child, which puts an obvious bound of n^2 on the maximal height of the parse tree.[15] This quadratic bound is tight. Indeed, consider the sequence (g_k), where $g_{k+1} = h_{k \,(\mathrm{mod}\, n-1)}$ for $0 \le k \le (n-2)(n-1)$, and (besides self-loops) h_k consists of the single edge $(k+2, k+1)$ for $k = 0, \ldots, n-2$. The j-th copy of h_k adds to the cumulant the new edge $(k+j+1, k+1)$, which creates, in total, a quadratic number of increments. The bounded depth implies that the parse tree for an infinite sequence includes exactly one node with an infinite number of children. That node is expressed by a production of type (1b).

Undirected graphs

Note that the cumulant of a sequence of undirected graphs might itself be directed.[16] Suppose that an undirected edge e of the digraph g does not extend any edge of g into a new temporal 2-edge walk.[17] Obviously, e must already be in the digraph; furthermore, any vertex linking to one of its endpoints must link to the other one as well. This implies that the *undirected transitive front* of a directed graph (say, a cumulant) consists of disjoint cliques of undirected edges. This simplifies the parsing since the condensation is trivial and the parsing tree has no nodes of type (2b). The complexity of the parse tree can still be as high as quadratic, however. To see why, consider the following recursive construction: given a clique C_k over k vertices at time t, attach to it, at time $t+1$, a two-edge path $(x, y), (y, z)$, say, at x. The cumulant gains the edge (y, z) as well as all k edges joining y to the clique. At time $t+2, \ldots, t+k+1$, visit each one of these k edges by using single-edge graphs g_i. Each such step will see the addition of a new edge to the cumulant, until it becomes a clique C_{k+2}.

[15] This differs from the linear-depth trees derived from the flow tracker [4].
[16] The product $(x \leftrightarrow y \leftrightarrow z) \times (x \quad y \leftrightarrow z)$ has a directed edge from x to z but not from z to x.
[17] An edge is undirected if both of its directed versions are present in the graph.

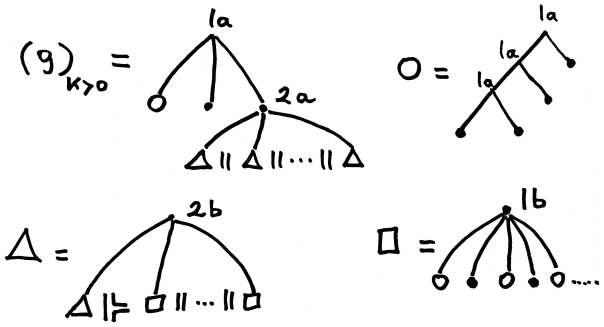


Figure 2 The parse tree of an infinite sequence consisting of the same graph g.

(Note that visiting (y, z) would ruin the whole construction.) The quadratic lower bound on the tree depth follows immediately.

Backward parsing

The sequence of graphs leads to products where each new graph is multiplied to the right, as would happen in a time-varying Markov chain. Algebraically, the matrices are multiplied from left to right. In diffusive systems (eg, multiagent agreement systems, Hegselmann-Krause models, Deffuant systems, voter models), however, matrices are multiplied from right to left. Although the dynamics can be quite different, the same parsing algorithm can be used. Given a sequence $\mathbf{g} = (g_k)_{k>0}$, its *backward parsing* is formed by applying the parser to the sequence $\overleftarrow{\mathbf{g}} = (h_k)_{k>0}$, where h_k is derived from g_k by reversing the direction of every edge, ie, (x, y) becomes (y, x). Once the parse tree for $\overleftarrow{\mathbf{g}}$ has been built, we simply restore each edge to its proper direction to produce the *backward parse tree* of $\mathbf{g}$.

3 The Markov Influence Model

Let $\mathbb{S}^{n-1}$ (or $\mathbb{S}$ when the dimension is understood) be the standard simplex $\left\{ \mathbf{x} \in \mathbb{R}^n \mid \mathbf{x} \geq \mathbf{0}\,,\ \|\mathbf{x}\|_1 = 1 \right\}$ and let $\mathcal{S}$ denote set of all n-by-n rational stochastic matrices. A *Markov influence system* (*MIS*) is a discrete-time dynamical system with phase space $\mathbb{S}$, which is defined by the map $f\colon \mathbf{x} \mapsto f(\mathbf{x}) := \mathbf{x}^\top S(\mathbf{x})$, where S is a function $\mathbb{R}^n \mapsto \mathcal{S}^n$ that is constant over the pieces of a finite polyhedral partition[18] $\mathcal{P} = \{P_k\}$ of $\mathbb{R}^n$. We define the digraph $g(\mathbf{x})$ (and its corresponding Markov chain) formed by the positive entries of $S(\mathbf{x})$. To avoid irrelevant technicalities, we assume the presence of self-loops in $g(\mathbf{x})$, ie, $S(\mathbf{x})_{ii} > 0$. In this way, any orbit of an *MIS* corresponds to a lazy, time-varying random walk with transitions defined endogenously.[19] We recall some basic terminology. The *orbit* of $\mathbf{x} \in \mathbb{S}$ is the infinite sequence $(f^t(\mathbf{x}))_{t\geq 0}$ and its *itinerary* is the corresponding sequence of cells P_k visited in the process. The orbit is *periodic* if $f^t(\mathbf{x}) = f^s(\mathbf{x})$ for any $s = t$ modulo a fixed integer. It is asymptotically periodic if it gets arbitrarily close to a periodic orbit over time.

For convenience, we assume a representation of the discontinuities induced by $\mathcal{P}$ as hyperplanes H_i of the form $\mathbf{a}_i^\top \mathbf{x} = 1 + \delta$, where $\delta \in \frac{1}{2}[-1, 1]$ (for concreteness). Note that the polyhedral partition is invariant up to scaling for all values of the bifurcation parameter,

[18] How f is defined on the discontinuities of the partition is immaterial.
[19] As discussed in the introduction, to access the full power of first-order logic in the stepwise choice of digraphs requires nonlinear partitions, which can be handled by a suitable tensor lift.

so the *MIS* remains well-defined as we vary δ. The parameter δ is necessary for the analysis: indeed, as we explain below in Section 5, chaos cannot be avoided without it. The *coefficient of ergodicity* $\tau(M)$ of a matrix M is defined as half the maximum ℓ_1-distance between any two of its rows [20]. It is submultiplicative for stochastic matrices, a direct consequence of the identity

$$\tau(M) = \max\Big\{\, \|\mathbf{x}^\top M\|_1 : \mathbf{x}^\top \mathbf{1} = 0 \;\text{ and }\; \|\mathbf{x}\|_1 = 1 \,\Big\}.$$

Given $\Omega \subset \mathbb{R}$, let L^t_Ω denote the set of t-long prefixes of any itinerary for any starting position $\mathbf{x} \in \mathbb{S}$ and any $\delta \in \Omega$. We define the *ergodic renormalizer* $\eta = \eta(\Omega)$ as the smallest integer such that, for any $t \geq \eta$ and any matrix sequence $S_1, \ldots, S_t$ matching an element of L^t_Ω, the product $S_1 \cdots S_t$ is primitive (ie, some high enough power is a positive matrix) and its coefficient of ergodicity is less than $1/2$. We assume in this section that $\eta = \eta(\mathbb{R}) < \infty$ and discuss in §4 how to remove this assumption via renormalization. Let D be the union of the hyperplanes H_i from $\mathcal{P}$ in $\mathbb{R}^n$ (where δ is understood). We define $Z_t = \bigcup_{0 \leq k \leq t} f^{-k}(D)$ and $Z = \bigcup_{t \geq 0} Z_t$. Remarkably, for almost all δ, Z_t becomes strictly equal to Z in a finite number of steps.

▶ **Lemma 1.** *Given any $\varepsilon > 0$, there exists an integer $\nu \leq 2^{\eta^{O(1)}} |\log \varepsilon|$ and a finite union K of intervals of total length less than ε such that $Z = Z_\nu$ for any $\delta \notin K$.*

▶ **Corollary 2.** *For δ almost everywhere,*[20] *every orbit is asymptotically periodic.*

Proof. The equality $Z = Z_\nu$ implies the eventual periodicity of the symbolic dynamics. The period cannot exceed the number of connected components in the complement of Z. Once an itinerary becomes periodic at time t_o with period σ, the map f^t can be expressed locally by matrix powers. Indeed, divide $t - t_o$ by σ and let q be the quotient and r the remainder; then, locally, $f^t = g^q \circ f^{t_o + r}$, where g is specified by a stochastic matrix with a positive diagonal, which implies convergence to a periodic point at an exponential rate. Apply Lemma 1 repeatedly, with $\varepsilon = 2^{-l}$ for $l = 1, 2, \ldots$ and denote by K_l be the corresponding union of "forbidden" intervals. Define $K^l = \bigcup_{j \geq l} K_j$ and $K^\infty = \bigcap_{l > 0} K^l$: $\text{Leb}(K^l) < 2^{1-l}$; hence $\text{Leb}(K^\infty) = 0$. The lemma follows from the fact that any δ outside of K^∞ lies outside of K^l for some $l > 0$. ◀

The corollary states that the set of "nonperiodic" values of δ has measure zero in parameter space. Our result is actually stronger than that. We prove that the nonperiodic set can be covered by a Cantor set of Hausdorff dimension strictly less than 1. The remainder of this section is devoted to a proof of Lemma 1.

3.1 Shift spaces and growth rates

The *growth exponent* of a language is defined as $\lim_{n \to \infty} \frac{1}{n} \max_{k \leq n} \log N(k)$, where $N(k)$ is the number of words of length k; for example, the growth exponent of $\{0,1\}^*$ is 1. The language consisting of all the itineraries of a Markov influence system forms a *shift space* and its growth exponent is the *topological entropy* of its symbolic dynamics [21].[21] It can be strictly positive, which is a sign of chaos. We show that, for a typical system, it is zero, the key fact driving periodicity. Let $M_1, \ldots, M_T$ be n-by-n matrices from a fixed set $\mathcal{M}$ of

[20] Meaning everywhere in $\frac{1}{2}[-1, 1]$ outside a set of Lebesgue measure zero.
[21] Which should not be confused with the topological entropy of the *MIS* itself.

primitive stochastic rational matrices with positive diagonals, and assume that $\tau(M) < 1/2$ for $M \in \mathcal{M}$; hence $\tau(M_1 \cdots M_k) < 2^{-k}$. Because each product $M_1 \cdots M_k$ is a primitive matrix, it can be expressed as $\mathbf{1}\boldsymbol{\pi}_k^\top + Q_k$ (by Perron-Frobenius), where $\boldsymbol{\pi}_k$ is its (unique) stationary distribution.[22] If $\boldsymbol{\pi}$ is a stationary distribution for a stochastic matrix S, then its j-th row $\boldsymbol{s}_j$ satisfies $\boldsymbol{s}_j - \boldsymbol{\pi}^\top = \boldsymbol{s}_j - \boldsymbol{\pi}^\top S = \sum_i \boldsymbol{\pi}_i(\boldsymbol{s}_j - \boldsymbol{s}_i)$; hence, by the triangular inequality, $\|\boldsymbol{s}_j - \boldsymbol{\pi}\|_1 \le \sum_i \pi_i \|\boldsymbol{s}_j - \boldsymbol{s}_i\|_1 \le 2\tau(S)$. This implies that

$$\begin{cases} M_1 \cdots M_k = \mathbf{1}\boldsymbol{\pi}_k^\top + Q_k \\ \|Q_k\|_\infty \le 2\tau(M_1 \cdots M_k) < 2^{1-k}. \end{cases} \tag{1}$$

Property U

Fix a vector $\mathbf{a} \in \mathbb{Q}^n$, and denote by $M^{(\theta)}$ the n-by-m matrix with the m column vectors $M_1 \cdots M_{k_i}\,\mathbf{a}$, where $\theta = (k_1, \ldots, k_m)$ is an increasing integer sequence of nonnegative integers in $[T]$. We say that property **U** holds if there exists a vector $\mathbf{u}$ such that $\mathbf{1}^\top\mathbf{u} = 1$ and $\mathbf{x}^\top M^{(\theta)}\mathbf{u}$ does not depend on the variable $\mathbf{x} \in \mathbb{S}$.[23] Intuitively, property **U** is a quantifier elimination device for expressing "general position" for *MIS*. To see the connection, consider a simple statement such as "the three points $\big(x, x^2\big)$, $\big(x+1, (x+1)^2\big)$, and $\big(x+2, (x+2)^2\big)$ cannot be collinear for any value of x." This can be expressed by saying that a certain determinant polynomial in x is constant. Likewise, the vector $\mathbf{u}$ manufactures a quantity, $\mathbf{x}^\top M^{(\theta)}\mathbf{u}$, that "eliminates" the variable $\mathbf{x}$. Note that some condition on $\mathbf{u}$ is obviously needed since we could pick $\mathbf{u} = \mathbf{0}$. We explain below why $\mathbf{1}^\top\mathbf{u} = 1$ is the right condition.

▶ **Lemma 3.** *There exists a constant $b > 0$ (linear in n) such that, given any integer $T > 0$ and any increasing sequence θ in $[T]$ of length at least $T^{1-\alpha}/\alpha$, property* **U** *holds, where $\alpha := \mu^{-b}$ and μ is the number of bits needed to encode any entry of M_k for any $k \in [T]$.*

Proof. By choosing b large enough, we can automatically ensure that T is as big as we want.[24] The proof is a mixture of algebraic and combinatorial arguments. We begin with a Ramsey-like statement about stochastic matrices.

▶ **Lemma 4.** *There is a constant $d > 0$ such that, if the sequence θ contains $j_0, \ldots, j_n$ with $j_i \ge d\mu j_{i-1}$ for each $i \in [n]$, then property* **U** *holds.*

Proof. By (1), $\|Q_k\mathbf{a}\|_\infty < c_0 2^{-k}$ for constant $c_0 > 0$. Note that Q_k has rational entries over $O(\mu k)$ bits (with the constant factor depending on n). We write $M^{(\theta)} = \mathbf{1}\mathbf{a}^\top\Pi^{(\theta)} + Q^{(\theta)}$, where $\Pi^{(\theta)}$ and $Q^{(\theta)}$ are the n-by-m matrices formed by the m column vectors $\boldsymbol{\pi}_{k_i}$ and $Q_{k_i}\mathbf{a}$, respectively, for $i \in [m]$; recall that $\theta = (k_1, \ldots, k_m)$. The key fact is that the dependency on $\mathbf{x} \in \mathbb{S}$ is confined to the term $Q^{(\theta)}$: indeed,

$$\mathbf{x}^\top M^{(\theta)}\mathbf{u} = \mathbf{a}^\top\Pi^{(\theta)}\mathbf{u} + \mathbf{x}^\top Q^{(\theta)}\mathbf{u}. \tag{2}$$

This shows that, in order to satisfy property **U**, it is enough to ensure that $Q^{(\theta)}\mathbf{u} = 0$ has a solution such that $\mathbf{1}^\top\mathbf{u} = 1$. Let $\sigma = (j_0, \ldots, j_{n-1})$. If $Q^{(\sigma)}$ is nonsingular then, because

[22] Positive diagonals play a key role here because primitiveness is not closed under multiplication: for example, $\left(\begin{smallmatrix}1&1\\1&0\end{smallmatrix}\right)$ and $\left(\begin{smallmatrix}0&1\\1&1\end{smallmatrix}\right)$ are both primitive but their product is not.

[23] Because $\mathbf{x}$ is a probability distribution, property **U** does not imply $\mathbf{x}^\top M^{(\theta)}\mathbf{u} = 0$; for example, we have $\mathbf{x}^\top\big(\mathbf{1}\mathbf{1}^\top\big)\mathbf{u} = 1$, for $\mathbf{u} = \frac{1}{n}\mathbf{1}$.

[24] All the constants in this work may depend on the input parameters such as n, $\mathcal{P}$, etc. Dependency on other parameters is indicated by a subscript.

each one of its entries is a rational over $O(\mu j_{n-1})$ bits, we have $|\det Q^{(\sigma)}| \geq c_1^{\mu j_{n-1}}$, for constant $c_1 > 0$. Let R be the $(n+1)$-by-$(n+1)$ matrix derived from $Q^{(\sigma)}$ by adding the column $Q^{j_n}\mathbf{a}$ to its right and then adding a row of ones at the bottom. If R is nonsingular, then $R\,\mathbf{u} = (0, \ldots, 0, 1)^\top$ has a (unique) solution in $\mathbf{u}$ and property $\mathbf{U}$ holds (after padding $\mathbf{u}$ with zeroes). Otherwise, we expand the determinant of R along the last column. Suppose that $\det Q^{(\sigma)} \neq 0$. By Hadamard's inequality, all the cofactors are at most a constant $c_2 > 0$ in absolute value; hence, for d large enough,

$$0 = |\det R| \geq |\det Q^{(\sigma)}| - nc_2 \|Q_{j_n}\mathbf{a}\|_\infty \geq c_1^{\mu j_{n-1}} - nc_2c_0 2^{-j_n} > 0.$$

This contradiction implies that $Q^{(\sigma)}$ is singular, so (at least) one of its rows can be expressed as a linear combination of the others. We form the n-by-n matrix R' by removing that row from R, together with the last column, and setting $u_{j_n} = 0$ to rewrite $Q^{(\theta)}\mathbf{u} = 0$ as $R'\mathbf{u}' = (0, \ldots, 0, 1)^\top$, where $\mathbf{u}'$ is the restriction of $\mathbf{u}'$ to the columns indexed by R'. Having reduced the dimension of the system by one variable, we can proceed inductively in the same way; either we terminate with the discovery of a solution or the induction runs its course until $n = 1$ and the corresponding 1-by-1 matrix is null, so that the solution 1 works. Note that $\mathbf{u}$ has rational coordinates over $O(\mu T)$ bits. ◀

Let $N(T)$ be the largest sequence θ in $[T]$ such that property $\mathbf{U}$ does not hold. Divide $[T]$ into bins $[(d\mu)^k, (d\mu)^{k+1} - 1]$ for $k \geq 0$. By Lemma 4, the sequence θ can intersect at most $2n$ of them, so, if $T > t_0$, for some large enough $t_0 = (d\mu)^{O(n)}$, there is at least one empty interval in T of length $T/(d\mu)^{2n+3}$. This gives us the recurrence $N(T) \leq T$ for $T \leq t_0$ and $N(T) \leq N(T_1) + N(T_2)$, where $T_1 + T_2 \leq \beta T$, for a positive constant $\beta = 1 - (d\mu)^{-2n-3}$. The recursion to the right of the empty interval, say, $N(T_2)$, warrants a brief discussion. The issue is that the proof of Lemma 4 relies crucially on the property that Q_k has rational entries over $O(\mu k)$ bits—this is needed to lower-bound $|\det Q^{(\sigma)}|$ when it is not 0. But this is not true any more, because, after the recursion, the columns of the matrix $M^{(\theta)}$ are of the form $M_1 \cdots M_k\,\mathbf{a}$, for $T_1 + L < k \leq T$, where L is the length of the empty interval and $T = T_1 + T_2 + L$. Left as such, the matrices use too many bits for the recursion to go through. To overcome this obstacle, we observe that the recursively transformed $M^{(\theta)}$ can be factored as AB, where $A = M_1 \cdots M_{T_1+L}$ and B consists of the column vectors $M_{T_1+L+1} \cdots M_k\,\mathbf{a}$. The key observation now is that, if $\mathbf{x}^\top B\,\mathbf{u}$ does not depend on $\mathbf{x}$, then neither does $\mathbf{x}^\top M^{(\theta)}\,\mathbf{u}$, since it can be written as $\mathbf{y}^\top B\,\mathbf{u}$ where $\mathbf{y} = A^\top\mathbf{x} \in \mathbb{S}$. In this way, we can enforce property $\mathbf{U}$ while having restored the proper encoding length for the entries of $M^{(\theta)}$.

Plugging in the ansatz $N(T) = t_0 T^\gamma$, for some unknown positive $\gamma < 1$, we find by Jensen's inequality that, for all $T > 0$, $N(T) \leq t_0(T_1^\gamma + T_2^\gamma) \leq t_0 2^{1-\gamma}\beta^\gamma T^\gamma$. For the ansatz to hold true, we need to ensure that $2^{1-\gamma}\beta^\gamma \leq 1$. Setting $\gamma = 1/(1 - \log\beta) < 1$ completes the proof of Lemma 3. ◀

Define $\phi^k(\mathbf{x}) = \mathbf{x}^\top M_1 \cdots M_k$ and let $h_\delta : \mathbf{a}^\top\mathbf{x} = 1 + \delta$ be some hyperplane in $\mathbb{R}^n$. We consider a set of canonical intervals of length ρ (or less): $\mathcal{D}_\rho = \big\{\, [k\rho, (k+1)\rho] \cap \varepsilon\,\mathbb{I} \,|\, k \in \mathbb{Z} \,\big\}$, where $\mathbb{I} := [-1, 1]$ and $\varepsilon > 0$ is the parameter of Lemma 1. Roughly, the "general position" lemma below says that, for most δ, the ϕ^k-images of any ρ-wide cube centered in the simplex $\mathbb{S}^{n-1}$ cannot near-collide with the hyperplane $\mathbf{a}^\top\mathbf{x} = 1 + \delta$ for most values of $k \leq T$. This may sound seriously counterintuitive. After all, if the stochastic matrices M_i are the identity, the images do not move, so if the initial cube collide then all of the images will! The point is that M_i is primitive so it cannot be the identity. The low coefficients of ergodicity will also play a key role. Notation: α refers to its use in Lemma 3.

▶ **Lemma 5.** *For any real $\rho > 0$ and any integer $T > 0$, there exists $U \subseteq \mathcal{D}_\rho$ of size $c_T = 2^{O(\mu T)}$, where c_T is independent of ρ, such that, for any $\Delta \in \mathcal{D}_\rho \backslash U$ and $\mathbf{x} \in \mathbb{S}$, there*

are at most $T^{1-\alpha}/\alpha$ *integers* $k \leq T$ *such that* $\phi^k(X) \cap h_\Delta \neq \emptyset$*, where* $X = \mathbf{x} + \rho\mathbb{I}^n$ *and* $h_\Delta := \bigcup_{\delta\in\Delta} h_\delta$*.*

Proof. In what follows, $b_0, b_1, \ldots$ refer to suitably large positive constants. We assume the existence of more than $T^{1-\alpha}/\alpha$ integers $k \leq T$ such that $\phi^k(X) \cap h_\Delta \neq \emptyset$ and draw the consequences: in particular, we infer certain linear constraints on δ; by negating them, we define the forbidden set U and ensure the conclusion of the lemma. Let $k_1 < \cdots < k_m$ be the integers in question, where $m > T^{1-\alpha}/\alpha$. For each $i \in [m]$, there exists $\mathbf{x}(i) \in X$ such that $|\,\mathbf{x}(i)^\top M_1 \cdots M_{k_i}\mathbf{a} - 1 - \delta\,| \leq \rho$. By the stochasticity of the matrices, $|\,\big(\mathbf{x}(i) - \mathbf{x}\big)^\top M_1 \cdots M_{k_i}\mathbf{a}\,| \leq b_0\,\rho$; hence $|\,\mathbf{x}^\top M_1 \cdots M_{k_i}\mathbf{a} - 1 - \delta\,| \leq (b_0+1)\rho$. By Lemma 3, there is a rational vector $\mathbf{u}$ such that $\mathbf{1}^\top\mathbf{u} = 1$ and $\mathbf{x}^\top M^{(\theta)}\mathbf{u} = \psi(M^{(\theta)}, \mathbf{a})$ does not depend on the variable $\mathbf{x} \in \mathbb{S}$; on the other hand, $|\,\mathbf{x}^\top M^{(\theta)}\mathbf{u} - (1+\delta)\,| \leq b_1\rho$. Two quick remarks: (i) the term $1+\delta$ is derived from $(1+\delta)\mathbf{1}^\top\mathbf{u} = 1+\delta$; (ii) $b_1 \leq (b_0+1)\|\mathbf{u}\|_1$, where $\mathbf{u}$ is a rational over $O(\mu T)$ bits. We invalidate the condition on $k_1, \ldots, k_m$ by keeping δ outside the interval $\psi(M^{(\theta)}, \mathbf{a}) - 1 + b_1\rho\mathbb{I}$, which rules out at most $2(b_1+1)$ intervals from $\mathcal{D}_\rho$. Repeating this for all sequences $(k_1, \ldots, k_m)$ raises the number of forbidden intervals, ie, the size of U, to $c_T = 2^{O(\mu T)}$. ◀

Topological entropy

We identify the family $\mathcal{M}$ with the set of all matrices of the form $S_1 \cdots S_k$ for $\eta \leq k \leq 3\eta$. By definition of the ergodic renormalizer $\eta = \eta(\Omega)$ (for a set Ω that will be specified later), any $M \in \mathcal{M}$ is primitive and $\tau(M) < 1/2$; furthermore, both μ and $\log|\mathcal{M}|$ are in $O(\eta)$. Our next result implies a bound of $\lim_{T\to\infty} T^{-\eta^{-O(1)}} = 0$ on the topological entropy of the shift space of itineraries.

▶ **Lemma 6.** *For any real* $\rho > 0$ *and any integer* $T > 0$*, there exists* $t_\rho = O(\eta|\log\rho|)$ *and* $V \subseteq \mathcal{D}_\rho$ *of size* $d_T = 2^{O(T)}$ *such that, for any* $\Delta \in \mathcal{D}_\rho \setminus V$*, any integer* $t \geq t_\rho$*, and any* $\sigma \in L^t_\Delta$*,* $\log\big|\,\{\,\sigma' \,|\, \sigma\cdot\sigma' \in L^{t+T}_\Delta\,\}\,\big| \leq \eta^{b+1}T^{1-\eta^{-b}}$*, for constant* $b > 0$*.*

Proof. In the lemma, t_ρ (resp. d_T) is independent of T (resp. ρ). The main point is that the exponent of T is bounded away from 1. We define the set V as the union of the sets U formed by applying Lemma 5 to each one of the hyperplanes h_δ involved in $\mathcal{P}$ and every possible sequence of T matrices in $\mathcal{M}$. This increases c_T to $2^{O(\eta T)}$. Fix $\Delta \in \mathcal{D}_\rho \setminus V$ and consider the (lifted) phase space $\mathbb{S}\times\Delta$ for the dynamical system induced by the map $f_\uparrow : \big(\mathbf{x}, \delta\big) \mapsto \big(\,\mathbf{x}^\top S(\mathbf{x}), \delta\,\big)$. The system is piecewise-linear with respect to the polyhedral partition $\mathcal{P}_\uparrow$ of $\mathbb{R}^{n+1}$ formed by treating δ as a variable in h_δ. Let Υ_t be a continuity piece for $f^t_\uparrow$, ie, a maximal region of $\mathbb{S}\times\Delta$ over which the t-th iterate of $f_\uparrow$ is linear. By the argument leading to (1), therefore, any matrix sequence $S_1, \ldots, S_t$ matching an element of L^t_Δ is such that $S_1 \cdots S_t = \mathbf{1}\boldsymbol{\pi}^\top + Q$, where $\|Q\|_\infty < 2^{2-t/\eta}$; hence there exists $t_\rho = O(\eta|\log\rho|)$ such that, for any $t \geq t_\rho$, $f^t_\uparrow(\Upsilon_t) \subseteq (\mathbf{x} + \rho\mathbb{I}^n)\times\Delta$, for some $\mathbf{x} = \mathbf{x}(t, \Upsilon_t) \in \mathbb{S}$.

Consider a nested sequence $\Upsilon_1 \supseteq \Upsilon_2 \supseteq \cdots$.[25] We say there is a *split* at k if $\Upsilon_{k+1} \subset \Upsilon_k$, and we show that, given any $t \geq t_\rho$, there are only $O(\eta T^{1-\alpha}/\alpha)$ splits between t and $t+\eta T$, where $\alpha = \eta^{-b}$, for constant b (see Lemma 3 for definitions). We may confine our attention to splits caused by the same hyperplane h_δ (since $\mathcal{P}$ features only a constant number of them). Arguing by contradiction, we assume the presence of at least $6\eta T^{1-\alpha}/\alpha$ splits, which implies

[25] Note that Υ_1 is a cell of $\mathcal{P}_\uparrow$, $f^k_\uparrow(\Upsilon_{k+1}) \subseteq f^k_\uparrow(\Upsilon_k)$, and S_l is the stochastic matrix used to map $f^{l-1}_\uparrow(\Upsilon_l)$ to $f^l_\uparrow(\Upsilon_l)$ (ignoring the dimension δ).

that at least $N := 2T^{1-\alpha}/\alpha$ of those splits occur for values of k at least 2η apart. This is best seen by binning $[t+1, t+\eta T]$ into T intervals of length η and observing that at least $3N$ intervals must feature splits. In fact, this proves the existence of N splits at positions separated by a least two consecutive bins. Next, we use the same binning to produce the matrices $M_1, \ldots, M_T$, where $M_j = S_{t+1+(j-1)\eta} \cdots S_{t+j\eta}$.

Suppose that all of the N splits occur for values k of the form $t + j\eta$. In this case, a straightforward application of Lemma 5 is possible: we set $X \times \Delta = f^t_\uparrow(\Upsilon_t)$ and note that the functions ϕ^k are all products of matrices from the family $\mathcal{M}$, which happen to be η-long products. The number of splits, $2T^{1-\alpha}/\alpha$, exceeds the number allowed by the lemma and we have a contradiction. If the splits do not fall neatly at endpoints of the bins, we use the fact that $\mathcal{M}$ includes matrix products of any length between η and 3η. This allows us to reconfigure the bins so as to form a sequence $M_1, \ldots, M_T$ with the splits occurring at the endpoints: for each split, merge its own bin with the one to its left and the one to its right (neither of which contains a split) and use the split's position to subdivide the resulting interval into two new bins; we leave all the other bins alone.[26] This leads to the same contradiction, which implies the existence of fewer than $O(\eta T^{1-\alpha}/\alpha)$ splits at $k \in [t, t+\eta T]$; hence the same bound on the number of strict inclusions in the nested sequence $\Upsilon_t \supseteq \cdots \supseteq \Upsilon_{t+\eta T}$. The set of all such sequences forms a tree of depth ηT, where each node has at most a constant number of children and any path from the root has $O(\eta T^{1-\alpha}/\alpha)$ nodes with more than one child. Rescaling T to ηT and raising b completes the proof. ◀

3.2 Proof of Lemma 1

We show that the nonperiodic δ intervals can be covered by a Cantor set of Hausdorff dimension less than one. All the parameters below refer to Lemma 6 and are set in this order: $T(\eta)$, $\rho(T, \varepsilon)$, and $\nu(T, \rho, \varepsilon)$. The details follow. Let δ, Δ such that $\delta \in \Delta \in \mathcal{D}_\rho \backslash V$. Given a continuity piece $C^t \subseteq \mathbb{S}$ for f^t, the $(t+T)$-th iterate of f induces a partition of C^t into a finite number of continuity pieces $C^t_1, \ldots, C^t_m$, so we can define $\lambda_{t,T}(C^t) = \sum_i \operatorname{diam}_{\ell_\infty} f^{t+T}(C^t_i)$. As was observed in the proof of Lemma 6, $\operatorname{diam}_{\ell_\infty} f^{t+T}(C^t_i) = O(2^{-T/\eta} \operatorname{diam}_{\ell_\infty} f^t(C^t))$. That same lemma shows that if we pick $T = 2^{\eta^{2b}}$, for b large enough then, for any $t \geq t_\rho$,

$$\lambda_{t,T}(C^t) = \sum_{i=1}^m \operatorname{diam}_{\ell_\infty} f^{t+T}(C^t_i) \leq b 2^{\eta^{b+1} T^{1-\eta^{-b}}} 2^{-T/\eta} \operatorname{diam}_{\ell_\infty} f^t(C^t) \leq \tfrac{1}{2} \operatorname{diam}_{\ell_\infty} f^t(C^t). \quad (3)$$

Next we set $\rho = \varepsilon/(2d_T)$ so that the intervals of V cover a length of at most $\varepsilon/2$. This gives us an extra length of $\varepsilon/2$ worth of forbidden intervals at our disposal. For any $t = t_\rho + kT$ large enough, $f^t(\mathbb{S})$ is the union of (possibly overlapping) convex bodies $K_1, \ldots, K_p$. A key observation is that we can prevent any K_i from splitting at time $t + kT$ by keeping δ outside an interval of length $\operatorname{diam}_{\ell_\infty} K_i$ for each discontinuity of f. By iterating (3), we find that $\lambda_{t_\rho, kT}(C^{t_\rho}) \leq 2^{-k}$. We expand V by adding these intervals, which expands the total length covered by $2^{O(t_\rho)-k}$. To keep this expansion, as stated earlier, below $\varepsilon/2$, we set $k = O(t_\rho) + |\log \varepsilon|$. It follows that $Z_\nu = Z_{\nu+1}$ for $\nu + 1 = t_\rho + kT$, and hence $Z_t = Z$ for any $t \geq \nu$.[27] In view of $T = 2^{\eta^{2b}}$, $d_T = 2^{O(T)}$, $\rho = \varepsilon/(2d_T)$, and $t_\rho = O(\eta |\log \rho|)$, we

[26] We note the possibility of an inconsequential decrease in T caused by the merges. Also, we can now see clearly why Lemma 5 is stated in terms of the slab h_Δ and not the hyperplane h_δ. This allows us to express splitting caused by the hyperplane $\mathbf{a}^\top \mathbf{x} = 1 + \delta$ in lifted space $\mathbb{R}^{n+1}$.

[27] No point $\mathbf{x}$ is such that (a) $f^{\nu+1}(\mathbf{x})$ is in D (the union of the discontinuities) but $f^\nu(\mathbf{x})$ is not. To see why this implies that $Z_{t+1} = Z_t$ for any $t > \nu$, and hence $Z = Z_\nu$, suppose that $Z_{t+1} \supset Z_t$, ie,

observe that $|\log \rho| = |\log \varepsilon| + O(T)$, and both t_ρ and k are in $O(\eta|\log \varepsilon| + \eta T)$; hence

$$\nu = t_\rho + kT = O\big(\eta|\log \varepsilon|T + \eta T^2\big) = 2^{\eta^{O(1)}}|\log \varepsilon|,$$

which proves Lemma 1.

4 Applications

We show how the two sets of techniques developed above, renormalization and bifurcation analysis, allows us to resolve a few important families of *MIS*.

4.1 Irreducible systems

A Markov influence system is called *irreducible* if the Markov chain $g(\mathbf{x})$ is irreducible for all $\mathbf{x} \in \mathbb{S}$; with our self-loop assumption, this also means ergodic. All the digraphs $g(\mathbf{x})$ of an irreducible *MIS* are strongly connected; therefore, in the first instantiation of production (1a) $\mathbf{g} \to \big(\,\mathbf{g}_l\,\big)\, g_m\, \big(\,\mathbf{g}_r \triangle h\,\big)$, the right-hand side expands into

$$\mathbf{g} \longrightarrow ((((g_1)g_2)\cdots)g_{m-1})g_m\, \big(\,\mathbf{g}_r \triangle h\,\big), \tag{4}$$

with $h \in \mathbb{K}$ and $m < n$. In other words, every step sees growth in the cumulant until it is in $\mathbb{K}$ (the family of all complete digraphs). To see why, assume by contradiction that $\prod_{j<k} g_j = \prod_{j\le k} g_j$ for $k < m$. If so, then g_k is a subgraph of $tf(\prod_{j<k} g_j)$. Because the latter is transitive and it has in g_k a strongly connected subgraph that spans all the vertices, it must belong to $\mathbb{K}$; hence $k = m$, which contradicts our assumption. Since the last cumulant is in $\mathbb{K}$, the parsing of $\mathbf{g}_r \triangle h$ in (4) proceeds via (1b); hence

$$\begin{aligned}\mathbf{g} \longrightarrow\ & ((((g_1)g_2)\cdots)g_{m_1-1})\, g_{m_1} \\ & \Big(((((g_{m_1+1})g_{m_1+2})\cdots)g_{m_2-1})g_{m_2}\, ((((g_{m_2+1})g_{m_2+2})\cdots)g_{m_3-1})g_{m_3}\ \cdots\Big).\end{aligned} \tag{5}$$

It follows that $\eta(\mathbb{R})$ is polynomial in n. By Lemma 1 and Corollary 2, this shows that irreducible Markov influence systems are typically asymptotically periodic. We strengthen this result in our next application.

4.2 Weakly irreducible systems

We now assume a fixed partition of the vertices such that each digraph $g(\mathbf{x})$ consists of disjoint strongly connected graphs defined over the subsets $V_1, \ldots, V_l$ of the partition. Irreducible systems correspond to the case $l = 1$. What makes weak irreducibility interesting is that the systems are not simply the union of independent irreducible systems. Indeed, note that communication flows among states in two ways: (i) directly, vertices collect information from neighbors to update their states; and (ii) indirectly, via the polyhedral partition $\mathcal{P}$, the sequence of graphs for V_i may be determined by the current states within V_j. In the extreme case, we can have the co-evolution of two systems V_1 and V_2, each one depending entirely on the other one yet with no links between the two of them. If the two subsystems were

that $f^{t+1}(\mathbf{y})$ is in D but $f^t(\mathbf{y})$ is not, for $\mathbf{y} \in \mathbb{S}$; then (a) holds for $\mathbf{x} = f^{t-\nu}(\mathbf{y})$, a contradiction. This shows that the continuity pieces for f^ν are the same as for any $f^{\nu+k}$, which implies that the f-image of any such piece must fall entirely inside a single one of them. The eventual periodicity of the itinerary follows.

independent, their joint dynamics could be expressed as a direct sum and resolved separately. This cannot be done, in general, and the bifurcation analysis requires some modifications.

As before, the right-hand side of production (1a) expands into (4). The difference is that h is now a collection of disjoint complete digraphs $h_1, \ldots, h_l$, one for each V_i. This gives us an opportunity to use network renormalization (2a) to derive

$$\mathbf{g} \triangle h \longrightarrow \left(\mathbf{g}_{\mid h_1} \triangle h_1\right) \parallel \cdots \parallel \left(\mathbf{g}_{\mid h_l} \triangle h_l\right).$$

Each $\mathbf{g}_{\mid h_i} \triangle h_i$ is parsed as in (1b) into $\left(\mathbf{g}^1_{\mid h_i}\right) g^{m_1}_{\mid h_i} \left(\mathbf{g}^2_{\mid h_i}\right) g^{m_2}_{\mid h_i} \cdots$ (with indices moved up for clarity). For the bifurcation analysis, Lemma 4 relies on the rank-l expansion

$$\mathbf{x}^\top M^{(\theta)} = \sum_{i=1}^{l} \Big(\sum_{j \in V_i} x_j \Big) \mathbf{a}^\top \Pi_i^{(\theta)} + \mathbf{x}^\top Q^{(\theta)},$$

where (i) $M_1 \cdots M_k = \sum_{i=1}^{l} \mathbf{1}_{|V_i} \boldsymbol{\pi}_{i,k}^\top + Q_k$; (ii) all vectors $\mathbf{1}_{|V_i}$ and $\boldsymbol{\pi}_{i,k}$ have support in V_i; (iii) Q_k is block-diagonal and $\|Q_k\|_\infty < 2^{1-k}$; (iv) $\Pi_i^{(\theta)}$ and $Q^{(\theta)}$ are formed, respectively, by the column vectors $\boldsymbol{\pi}_{i,k_j}$ and $Q_{k_j}\,\mathbf{a}$ for $j \in [m]$, with $\theta = (k_1, \ldots, k_m)$. Property **U** no longer holds, however (see §3.1): indeed, if $l > 1$, it is no longer true that $\mathbf{x}^\top M^{(\theta)} \mathbf{u}$ is independent of the variable $\mathbf{x} \in \mathbb{S}$. The dependency is confined to the sums $s_i := \sum_{j \in V_i} x_j$ for $i \in [l]$. The key observation is that these sums are time-invariant. We fix them once and for all and redefine the phase space as the invariant manifold $\prod_{i=1}^{l} \big(s_i\, \mathbb{S}^{|V_i|-1}\big)$, which induces a foliation of the original simplex $\mathbb{S}^{n-1}$ via $\mathbb{S}^{l-1}$. The rest of the proof mimics the irreducible case, whose conclusion therefore still applies.

4.3 Condensation systems

We now assume that the condensations of the $g(\mathbf{x})$ all share the same transitive reduction. In other words, the condensations may change with time but all of them feature the same pairs of path-connected vertices.[28] Past the first n steps, a temporal walk will have been set up joining all pairs of path-connected vertices. We ignore the possibility of a call to (2a), which would be handled as in the weakly irreducible case. The parse tree features a node labeled (2b), where the cumulant h is the common transitive closure of any $g(\mathbf{x})$:

$$\mathbf{g} \triangle h \longrightarrow \left(\mathbf{g}_{\mid h'} \triangle h'\right) \Vdash \left\{ \left(\mathbf{g}_{\mid h_1} \triangle h_1\right) \parallel \cdots \parallel \left(\mathbf{g}_{\mid h_l} \triangle h_l\right) \right\}.$$

It helps to think of the right-hand side of $\Vdash$ as the absorbing states of a time-varying Markov chain. Every n steps, another temporal walk is established to match any path in h. Let $\sum_{h'} x_i$ be the sum of the probabilities at the nodes of the stem h'. It is easy to see that, after every interval of n steps, $\sum_{h'} x_i$ is multiplied by less than $1 - O(1)^{-n} < 1/c$ for constant $c > 1$. We are now ready to reduce the problem to the case of weakly irreducible systems. Indeed, the sums s_i approach a fixed value with an additive error rate of $c^{-t/n}$, which is fast enough to keep the previous analysis valid. We omit the technical details, which simply recycle the reasoning from the previous sections.

The term "typically" below means almost everywhere, ie, for δ in a subset of $\frac{1}{2}[-1, 1]$ of full Lebesgue measure 1. Recall that the result below applies, de facto, to irreducible and weakly irreducible systems.

[28] It is is implicit that the vertices of the condensations must match the same set of vertices. Note that the condensations have the same transitive closure and that any system with a time-invariant condensation forms a condensation *MIS*.

▶ **Theorem 7.** *Typically, every orbit of a condensation Markov influence system is asymptotically periodic.*

5 Hyper-Torpid Mixing and Chaos

Among the *MIS* that converge to a single stationary distribution, we show that the mixing time can be super-exponential. The creation of new timescales is what most distinguishes *MIS* from standard Markov chains. As we mentioned earlier, the system can be chaotic. We prove all of these claims below.

5.1 A super-exponential mixer

How can reaching a fixed point distribution take so long? Before we answer this question formally, we provide a bit of intuition. Imagine having three unit-volume water reservoirs A, B, C alongside a clock that rings at noon and 1pm every day. Initially, the clock is at 2pm and A is full while B and C are empty. Reservoir A transfers half of its contents to B and repeats this each hour until the clock rings noon. At this point, reservoir A empties into C the little water that it has left. Next, the clock now rings 1pm and B empties its contents into A. At 2pm, we resume what we did the day before at the same hour, ie, A transfers half of its water contents to B, etc. This goes on until some day, at 1pm, reservoir C finds its more than half full. (Note that the water level of C rises by about 10^{-3} every day.) At this point, both B and C transfer all their water back to A, so that at 2pm on that day, we are back to square 1. The original 12-step clock has been extended into a new clock of period roughly 1,000. The proof below shows how to simulate this iterative process with an *MIS*.

▶ **Theorem 8.** *There exist Markov influence systems that mix to a stationary distribution in time equal to a tower-of-twos of height linear in the number of states.*

Proof. We construct an *MIS* with a periodic orbit of length equal to a power-of-twos. It is easy to turn it into an orbit with a fixed-point attractor that reaches a stationary distribution and we omit this part of the discussion. Assume, by induction, that we have a Markov influence system M cycling through states $1, \ldots, p$, for $p > 1$. We build another one with period c^p, for fixed $c > 1$, by adding a "gadget" to it consisting of a three-vertex graph $1, 2, 3$ with probability distribution $(x, y, z) \in \mathbb{S}$. We initialize the system by placing M in state 1 (ie, 2pm in our clock example) and setting $x = 1$. The dynamic graph is specified by these rules:

1. Suppose that $z > 1/2$. If M is in state p, then the graph has the edges $(2, 1)$ and $(3, 1)$; both are given probability 1 (so that nodes 2, 3 have no self-loops). If M is in any other state, the graph has only three self-loops, each one assigned probability 1.
2. Suppose that $z \leq 1/2$.
   a. If M is in state $1, \ldots, p-2$, then the graph has the edge $(1, 2)$, which is assigned probability $1/2$, as is the self-loop at 1.
   b. If M is in state $p-1$, then the graph has the edge $(1, 3)$, which is assigned probability 1 (hence no self-loop at 1).
   c. If M is in state p, then the graph has the edge $(2, 1)$, which is assigned probability 1 (hence no self-loop at 2).

Suppose that M is in state 1 and that $y = 0$ and $z \leq 1/2$. When M reaches state $p-1$, the probability vector is $\big((1-z)2^{2-p}, (1-z)(1-2^{2-p}), z\big)$. At the next step, M is in state p and the vector becomes $\big(0, (1-z)(1-2^{2-p}), z+(1-z)2^{2-p}\big)$. If the last coordinate is still at most $1/2$, then M moves to state 1 and the vector becomes $\big((1-z)(1-2^{2-p}), 0, z+(1-z)2^{2-p}\big)$. The key observation is that z increases by $(1-z)2^{2-p}$, which is between 2^{1-p} and 2^{2-p} as

long as $z \le 1/2$. Since z begins at 0, it will cross the threshold $1/2$ after on the order of $p2^p$ steps. Transfers of mass to vertex 3 only happens when M is in state $p-1$, so at the next step, M is in state p and $z > 1/2$. The system moves to state 1 and restores its initial vector $(1,0,0)$: the cycle is closed. The construction on top of M adds three new vertices so we can push this recursion roughly $n/3$ times to produce a Markov influence system that is periodic with a period of length equal to a tower-of-twos of height roughly $n/3$. We tie up the loose ends now:

- The construction needs to recognize two consecutive states of M: they are labeled $p-1$ and p in our description but, by symmetry, they could be any other consecutive pair. We give each one of these two states their own distinct polyhedral cell. The obvious choice is the unique state satisfying $z > 1/2$, which is followed by the only state such that $x \ge 1$.
- The basis case of our inductive construction consists of a two-vertex system of constant period k with initial distribution $(1,0)$. If $x > 2^{1-k}$, the graph has an edge from 1 to 2 assigned probability $1/2$; else an edge from 2 to 1 given probability 1 to reset the system.
- The construction assumes probabilities summing up to 1 within each of $\lfloor (n-2)/3 \rfloor + 1$ gadgets, which is clearly wrong. Being piecewise-linear, however, the system suggests an easy fix: we divide the probability weights equally among each gadget and adjust the linear discontinuities appropriately. ◀

5.2 Chaos

We give a simple 5-state construction with chaotic symbolic dynamics:

$$A = \frac{1}{3}\begin{pmatrix} 2 & 1 & 0 & 0 & 0 \\ 0 & 1 & 2 & 0 & 0 \\ 0 & 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 2 & 1 \\ 0 & 0 & 0 & 0 & 3 \end{pmatrix} \quad \text{if } x_1 + x_2 > x_4 \qquad \text{and} \qquad B = \frac{1}{3}\begin{pmatrix} 1 & 0 & 2 & 0 & 0 \\ 1 & 2 & 0 & 0 & 0 \\ 0 & 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 2 & 1 \\ 0 & 0 & 0 & 0 & 3 \end{pmatrix} \quad \text{else,}$$

for $\mathbf{x} \in \mathbb{S}^4$. We focus our attention on $\Sigma = \left\{ (x_1, x_2, x_4) \,|\, 0 < x_1 \le x_4/2 \text{ and } x_4/2 \le x_2 < x_4 \right\}$, and easily check that it is an invariant manifold. At time 0, we fix $x_4 = 1/4$ and $x_5 = 0$; at all times, of course, $x_3 = 1 - x_1 - x_2 - x_4 - x_5$. The variable $y := (2x_2 - x_4)/(2x_1 - x_4)$ is always nonpositive over Σ.[29] It evolves as follows:

$$y \leftarrow \begin{cases} \frac{1}{2}(y+1) & \text{if } y < -1 \\ \frac{2y}{y+1} & \text{if } -1 \le y \le 0. \end{cases}$$

Writing $z = (y+1)/(y-1)$, we note that $-1 \le z < 1$ and it evolves according to $z \mapsto 2z+1$ if $z \le 0$, and $z \mapsto 2z-1$ otherwise, a map that conjugates with the baker's map and is well known to be chaotic [11].

Acknowledgments. I wish to thank Maria Chudnovsky and Ramon van Handel for helpful comments.

References

1 C. Avin, M. Koucký, and Z. Lotker. How to explore a fast-changing world (cover time of a simple random walk on evolving graphs). *Proc. 35th ICALP*, pages 121–132, 2008.

[29] We used a different but similar construction in [4].

2 H. Bruin and J.H.B. Deane. Piecewise contractions are asymptotically periodic. *Proc. American Mathematical Society*, 137(4):1389–1395, 2009.
3 C. Castellano, S. Fortunato, and V. Loreto. Statistical physics of social dynamics. *Rev. Mod. Phys.*, 81:591–646, 2009.
4 B. Chazelle. Diffusive influence systems. *SIAM J. Comput.*, 44:1403–1442, 2015.
5 B. Chazelle, Q. Jiu, Q. Li, and C. Wang. Well-posedness of the limiting equation of a noisy consensus model in opinion dynamics. *J. Differential Equations*, 263:365–397, 2017.
6 A. Condon and D. Hernek. Random walks on colored graphs. *Random Structures and Algorithms*, 5:285–303, 1994.
7 A. Condon and R.J. Lipton. On the complexity of space bounded interactive proofs. *Proc. 30th IEEE Symp. on Foundations of Computer Science*, pages 462–267, 1989.
8 D. Coppersmith and P. Diaconis. Random walk with reinforcement. *Unpublished manuscript*, 1986.
9 O. Denysyuk and L. Rodrigues. Random walks on directed dynamic graphs. *Proc. 2nd International Workshop on Dynamic Networks: Algorithms and Security (DYNAS'10), Bordeaux, France (Also arXiv:1101.5944 (2011))*, 2010.
10 O. Denysyuk and L. Rodrigues. Random walks on evolving graphs with recurring topologies. *Proc. 28th International Symposium on Distributed Computing (DISC)*, 2014.
11 R.L. Devaney. An Introduction to Chaotic Dynamical Systems (2nd ed.). *Westview Press*, 2003.
12 P. Diaconis. Recent progress on de Finetti's notions of exchangeability. *Bayesian Statistics*, 3, 1988.
13 T.D. Frank. Strongly nonlinear stochastic processes in physics and the life sciences. *ISRN Mathematical Physics*, Article ID 149169, 2013.
14 P. Holme and J. Saramäki. Temporal networks. *Physics Reports*, 519:97–125, 2012.
15 G. Iacobelli and D.R. Figueiredo. Edge-attractor random walks on dynamic networks. *J. Complex Networks*, 5:84–110, 2017.
16 A. Katok and B. Hasselblatt. Introduction to the Modern Theory of Dynamical Systems. *Cambridge University Press*, 1996.
17 D. Kempe, J. Kleinberg, and A. Kumar. Connectivity and inference problems for temporal networks. *J. Computer and System Sciences*, 64:820–842, 2002.
18 V.N. Kolokoltsov. Nonlinear Markov Processes and Kinetic Equations. *Cambridge Tracks in Mathematics (182)*, Cambridge Univ. Press, 2010.
19 M. Othon. An introduction to temporal graphs: an algorithmic perspective. *Internet Mathematics*, 12, 2016.
20 E. Seneta. Non-Negative Matrices and Markov Chains. *Springer, 2nd ed.*, 2006.
21 S. Sternberg. Dynamical Systems. *Dover Books on Mathematics*, 2010.

Learning Dynamics and the Co-Evolution of Competing Sexual Species*

Georgios Piliouras[†1] **and Leonard J. Schulman**[‡2]

1 **Singapore University of Technology and Design, Singapore**
georgios@sutd.edu.sg
2 **California Institute of Technology, Pasadena, USA**
schulman@caltech.edu

Abstract

We analyze a stylized model of co-evolution between any two purely competing species (e.g., host and parasite), both sexually reproducing. Similarly to a recent model of Livnat *et al.* [11] the fitness of an individual depends on whether the truth assignments on n variables that reproduce through recombination satisfy a particular Boolean function. Whereas in the original model a satisfying assignment always confers a small evolutionary advantage, in our model the two species are in an evolutionary race with the parasite enjoying the advantage if the value of its Boolean function matches its host, and the host wishing to mismatch its parasite. Surprisingly, this model makes a simple and robust behavioral prediction. The typical system behavior is *periodic*. These cycles stay bounded away from the boundary and thus, *learning-dynamics competition between sexual species can provide an explanation for genetic diversity.* This explanation is due solely to the natural selection process. No mutations, environmental changes, etc., need be invoked.

The game played at the gene level may have many Nash equilibria with widely diverse fitness levels. Nevertheless, sexual evolution leads to gene coordination that implements an optimal strategy, i.e., an optimal population mixture, at the species level. Namely, the play of the many "selfish genes" implements a time-averaged correlated equilibrium where the average fitness of each species is exactly equal to its value in the two species zero-sum competition.

Our analysis combines tools from game theory, dynamical systems and Boolean functions to establish a novel class of conservative dynamical systems.

1998 ACM Subject Classification J.3 Life and Medical Sciences, J.4 Social and Behavioral Sciences

Keywords and phrases Dynamical Systems, Potential Game, Team Zero-Sum Game, Boolean Functions, Replicator Dynamics

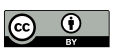
Digital Object Identifier 10.4230/LIPIcs.ITCS.2018.59

An exciting recent line of work in the theory of computation has focused on the algorithmic power of the evolutionary process (Valiant [22], Livnat et al. [11, 10]). The latter two papers identified as interesting the case of a sexually reproducing, haploidal, and panmictic species, evolving in a fixed environment according to variants of Multiplicative Weights Update

* A full version of the paper is available at [17], https://arxiv.org/abs/1711.06879.

† Supported in part by SUTD grant SRG ESD 2015 097, MOE AcRF Tier 2 Grant 2016-T2-1-170 and a NRF Fellowship. Part of the work was completed while GP was a CMI Wally Baer and Jeri Weiss postdoctoral scholar at Caltech. Part of the work was completed while GP was a Simons Institute research fellow.

‡ Supported in part by NSF grants 1319745 and 1618795, and, while in residence at the Israel Institute for Advanced Studies, by a EURIAS Senior Fellowship co-funded by the Marie Skłodowska-Curie Actions under the 7th Framework Programme.



dynamics [4, 5] – which are typically referred to as "replicator dynamics" in the evolutionary dynamics literature [23]. Curiously, however, Mehta et al. [12] made the discovery that these dynamics lead in the long run (in almost all cases) to a genetic monoculture. This rather contradicts the evidence of natural diversity around us.

Several plausible explanations exist for this discrepancy, including: (a) mutations [13], (b) speciation (*e.g.*, the Bateson-Dobzhansky-Muller model) [9], (c) the mathematical assumptions are too far from reality, (d) "in the long run" is longer than geologic time. There is, however, a long-standing argument, that there is another (and perhaps more important) factor driving diversity; to our knowledge this case was first compellingly laid out by Ehrlich and Raven in 1964 [6]: "It is apparent that reciprocal selective responses have been greatly underrated as a factor in the origination of organic diversity." (Already Darwin noted the significance of co-evolution, e.g., between orchids and moths that feed on their nectar; but the proposed implication for diversity seems to have come later.) In the ensuing decades this idea played a role in the *Red Queen Hypothesis* [21] and was advanced as an explanation of an advantage of sexual over asexual reproduction [1].

Apart from empirical study (e.g., [3, 20, 16, 2]), the dynamics of co-evolution have also been studied mathematically, but primarily (explicitly or implicitly) for asexual reproduction – dynamics in which the abundance of a genome changes over time in proportion to its fitness (possibly with mutations), as in the work of Eigen, Schuster and others [7, 8, 15, 18, 19]. The case of sexual reproduction, however, is quite different. There is a good mathematical model for these dynamics, called the "weak selection" model [14], but effects of co-evolution are not yet understood in this model.

We study a specific class of systems in this model, and provide a quantitative study of the evolutionary dynamics of sexual species in highly competitive ("zero sum") interactions. This study supports the thesis of Ehrlich and Raven, that competition drives diversity, in a strong form: not only does a genetic monoculture not take over, but in fact the entropy of the species' genomes is bounded away from 0 for all time. Thus we support a rationale for ecosystem diversity without invoking mutation, speciation or environmental change.

References

1 G. Bell. *The Masterpiece Of Nature: The Evolution and Genetics of Sexuality*. Univ. of California Press, 1982.

2 C. W. Benkman, T. L. Parchman, A. Favis, and A. M. Siepielski. Reciprocal selection causes a coevolutionary arms race between crossbills and lodgepole pine. *The American Naturalist*, 162(2):182–194, 2003.

3 E. D. Brodie Jr., B. J. Ridenhour, and E. D. Brodie III. The evolutionary response of predators to dangerous prey: Hotspots and coldspots in the geographic mosaic of coevolution between garter snakes and newts. *Evolution*, 56(10):2067–2082, 10 2002.

4 E. Chastain, A. Livnat, C. Papadimitriou, and U. Vazirani. Multiplicative updates in coordination games and the theory of evolution. In *Proceedings of the 4th Conference on Innovations in Theoretical Computer Science*, ITCS '13, pages 57–58, New York, NY, USA, 2013. ACM.

5 Erick Chastain, Adi Livnat, Christos Papadimitriou, and Umesh Vazirani. Algorithms, games, and evolution. *Proceedings of the National Academy of Sciences*, 111(29):10620–10623, 2014.

6 P. R. Ehrlich and P. H. Raven. Butterflies and plants: A study in coevolution. *Evolution*, 18(4):586–608, Dec. 1964.

7 M. Eigen. Selforganization of matter and the evolution of biological macromolecules. *Naturwissenschaften*, 58(10):465–523, 1971.

8 M. Eigen and P. Schuster. *The Hypercycle: A Principle of Natural Self-Organization*. Springer-Verlag, 1979.

9 S. Gavrilets. *Fitness Landscapes and the Origin of Species*. Princeton University Press, 2004.

10 A. Livnat and C. Papadimitriou. Sex as an algorithm: the theory of evolution under the lens of computation. *Communications of the ACM (CACM)*, 59:84–93, November 2016.

11 A. Livnat, C. Papadimitriou, A. Rubinstein, A. Wan, and G. Valiant. Satisfiability and evolution. In *FOCS*, 2014.

12 R. Mehta, I. Panageas, and G. Piliouras. Natural selection as an inhibitor of genetic diversity. In *ITCS*, 2015.

13 R. Mehta, I. Panageas, G. Piliouras, P. Tetali, and V. V. Vazirani. Mutation, sexual reproduction and survival in dynamic environments. In *ITCS*, 2017.

14 T. Nagylaki. The evolution of multilocus systems under weak selection. *Genetics*, 134(2):627–47, 1993.

15 M. A. Nowak and H. Ohtsuki. Prevolutionary dynamics and the origin of evolution. *Proceedings of the National Academy of Sciences*, 105(39):14924–14927, 2008.

16 O. Pellmyr. Yuccas, yucca moths, and coevolution: A review. *Annals of the Missouri Botanical Garden*, 90(1):35–55, 2003.

17 G. Piliouras and L. J. Schulman. Learning dynamics and the co-evolution of competing sexual species. *Arxiv preprint*, 2017. URL: `https://arxiv.org/abs/1711.06879`.

18 J. N. Thompson. *The Coevolutionary Process*. U Chicago Press, 1994.

19 J. N. Thompson. *The Geographic Mosaic of Coevolution*. U Chicago Press, 2005.

20 J. N. Thompson and B. M. Cunningham. Geographic structure and dynamics of coevolutionary selection. *Nature*, 417:735–738, 2002. `doi:10.1038/nature00810`.

21 L. Van Valen. A new evolutionary law. *Evolutionary Theory*, 1:1–30, 1973.

22 L. Valiant. *Probably Approximately Correct: Nature's Algorithms for Learning and Prospering in a Complex World*. Basic Books, 2013.

23 J. W. Weibull. *Evolutionary Game Theory*. MIT Press; Cambridge, MA, 1995.