



Sum-of-Squares Lower Bounds for Densest k -Subgraph

Chris Jones

Bocconi University

Milan, Italy

chris.jones@unibocconi.it

Goutham Rajendran

Carnegie Mellon University

Pittsburgh, United States

gouthamr@cmu.edu

Aaron Potechin

University of Chicago

Chicago, United States

potechin@uchicago.edu

Jeff Xu

Carnegie Mellon University

Pittsburgh, United States

jeffxusichao@cmu.edu

ABSTRACT

Given a graph and an integer k , Densest k -Subgraph is the algorithmic task of finding the subgraph on k vertices with the maximum number of edges. This is a fundamental problem that has been subject to intense study for decades, with applications spanning a wide variety of fields. The state-of-the-art algorithm is an $O(n^{1/4+\varepsilon})$ -factor approximation (for any $\varepsilon > 0$) due to Bhaskara et al. [STOC '10]. Moreover, the so-called *log-density framework* predicts that this is optimal, i.e. it is impossible for an efficient algorithm to achieve an $O(n^{1/4-\varepsilon})$ -factor approximation. In the average case, Densest k -Subgraph is a prototypical noisy inference task which is conjectured to exhibit a *statistical-computational gap*.

In this work, we provide the strongest evidence yet of hardness for Densest k -Subgraph by showing matching lower bounds against the powerful Sum-of-Squares (SoS) algorithm, a meta-algorithm based on convex programming that achieves state-of-art algorithmic guarantees for many optimization and inference problems. For $k \leq n^{1/2}$, we obtain a degree n^δ SoS lower bound for the hard regime as predicted by the log-density framework.

To show this, we utilize the modern framework for proving SoS lower bounds on average-case problems pioneered by Barak et al. [FOCS '16]. A key issue is that small denser-than-average subgraphs in the input will greatly affect the value of the candidate pseudexpectation operator around the subgraph. To handle this challenge, we devise a novel matrix factorization scheme based on the *positive minimum vertex separator*. We then prove an intersection tradeoff lemma to show that the error terms when using this separator are indeed small.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

STOC '23, June 20–23, 2023, Orlando, FL, USA

© 2023 Copyright held by the owner/author(s). Publication rights licensed to ACM.

ACM ISBN 978-1-4503-9913-5/23/06...\$15.00

<https://doi.org/10.1145/3564246.3585221>

CCS CONCEPTS

• **Theory of computation** → **Semidefinite programming**; Proof complexity.

KEYWORDS

Densest k -Subgraph, Average-case complexity, Sum-of-Squares Lower Bounds

ACM Reference Format:

Chris Jones, Aaron Potechin, Goutham Rajendran, and Jeff Xu. 2023. Sum-of-Squares Lower Bounds for Densest k -Subgraph. In *Proceedings of the 55th Annual ACM Symposium on Theory of Computing (STOC '23)*, June 20–23, 2023, Orlando, FL, USA. ACM, New York, NY, USA, 12 pages. <https://doi.org/10.1145/3564246.3585221>

1 INTRODUCTION

In the Densest k -Subgraph problem, we are given an undirected graph G on n vertices and an integer k and we want to output the subgraph on k vertices with the most edges, or in other words, the subgraph on k vertices with the highest edge density. This is a natural generalization of the k -clique problem [65] and has been subject to a long line of work for decades [3, 4, 11, 18–20, 22, 23, 43–46, 50, 52, 53, 66, 67, 84, 99, 108]. This problem has been the subject of intense study partly because of its numerous connections to other problems and fields (e.g. [5, 30, 32, 33, 35, 36, 41, 54–56, 70, 71, 80, 82, 87, 94, 107, 110]). The best known approximation algorithm for this problem yields an approximation factor of $O(n^{1/4+\varepsilon})$ for any constant $\varepsilon > 0$, due to [19]. On the other hand, it is conjectured that no efficient algorithm can achieve an $O(n^{1/4-\varepsilon})$ approximation.

Densest k -Subgraph is a compelling problem because random instances (Erdős-Rényi graphs) are conjectured and widely believed to be the “hardest” instances for algorithms. In fact, the insight that “worst case is average case” was crucial to the aforementioned algorithm in [19]. Their idea of going from average-case instances to worst-case instances was generalized into the *log-density framework*, which has been further applied to various other problems [37, 38, 40]. Since an algorithm for random instances seems to be the crucial conceptual step needed to solve the problem on all instances, understanding these random instances is a pressing topic.

As stated in [19, 20, 23, 84], Densest k -Subgraph on a random graph is a landmark question in the field of average-case complexity. Moreover, the conjectured hardness of this problem on random instances (which is the focus of our work) has been used for applications in finance [8] and cryptography [7]. However, evidence of hardness for Densest k -Subgraph stands to be improved, both in the average-case and worst-case settings. For example, even in the worst-case setting, no work has been able to show that Densest k -Subgraph is hard to n^ϵ -approximate for a fixed $\epsilon > 0$ using any reasonable complexity-theoretic assumption (although some works come close, see Section 1.3). In the more interesting average-case setting of random graphs, relatively little progress has been made to justify hardness, let alone match the log-density framework.

In this work, we study the hardness of Densest k -Subgraph on random graphs through a generic, powerful algorithm for optimization known as the Sum-of-Squares (SoS) hierarchy [51, 79, 90, 93, 106]. The SoS hierarchy is a family of semidefinite programming relaxations for polynomial optimization problems which implements a certain type of “sum-of-squares reasoning”. Arguably at the center stage of average-case complexity in recent years, SoS has proven to be a highly effective tool for combinatorial and continuous optimization. First, it was shown that the SoS hierarchy is rich enough to capture the state-of-the-art convex relaxations for Sparsest Cut [9], Max-Cut [49], all Max k -CSPs [97], etc. Later on, the SoS hierarchy led to new algorithms for approximating CSPs [2, 12, 13] and breakthroughs in robust statistics [15, 16, 58, 61, 69, 75, 98], a highlight being the resolution of longstanding open problems in Gaussian mixture learning (over a decade of work culminating in [14, 83]). Finally, for a large class of problems, it has been shown that SoS algorithms are the most effective among all semidefinite programming relaxations [81]. Therefore, understanding the limits of SoS algorithms is an important research endeavour and lower bounds against SoS serve as strong evidence for algorithmic hardness [60, 62, 77].

In this paper, we prove that for $k \leq n^{\frac{1}{2}}$, SoS of degree n^δ does not offer any significant improvement in the conjectural hard regime of random instances for Densest k -Subgraph as predicted by the log-density framework. This settles the open questions raised in the works [20, 39, 101]. Considering that the algorithm of Bhaskara et al. [19] matching the log-density framework is captured by SoS, our lower bound completes the picture of the performance of SoS for Densest k -Subgraph for $k \leq n^{\frac{1}{2}}$. This gives solid evidence that the conjectured approximability thresholds for Densest k -Subgraph are correct.

1.1 Our Contributions

We will now describe our results on SoS lower bounds for Densest k -Subgraph that match the predictions of the log-density framework.

Consider the following hypothesis testing variant of the Densest k -Subgraph problem. For an integer n and a real $p \in [0, 1]$, let $\mathcal{G}_{n,p}$ denote the Erdős-Rényi random distribution where a graph on n vertices is sampled by choosing each edge to be present independently with probability p . For parameters $n, k \in \mathbb{N}$ and $p, q \in [0, 1]$, we are given a graph G sampled either from

- (1) The null distribution $\mathcal{G}_{n,p}$ or
- (2) The alternative distribution where we first sample $G \sim \mathcal{G}_{n,p}$, then a set $H \subseteq V(G)$ is chosen by including each vertex with probability $\frac{k}{n}$, and finally we replace H by a sample from $\mathcal{G}_{|H|,q}$.

and our goal is to correctly identify which distribution it came from, with non-negligible probability.

The hypothesis testing question is a “planted model” of Densest k -Subgraph which is conjectured to exhibit a statistics-computation gap [25, 26]. With high probability, for q slightly larger than p , the subgraph H in the alternative distribution is truly the densest subgraph of G with size k (hence the null and alternative distributions are statistically distinguishable), but it is conjecturally computationally impossible to distinguish the two cases (in the parameter regime below).

Studying algorithms for this hypothesis testing variant was crucial to the log-density framework [19], which both generalizes an algorithm for the hypothesis testing variant into a worst-case algorithm, and predicts the relationships between n, k, p, q for which the hypothesis testing problem is hard. In particular, consider the setting

$$k = n^\alpha, \quad p = n^{-\beta}, \quad q = n^{-\gamma}$$

for constants $\alpha \in (0, 1/2]$, $\beta \in (0, 1)$, $\gamma \in (0, 1)$, a notation that we will use throughout this paper. According to the framework, it’s algorithmically hard to solve the problem if

$$\gamma > \alpha\beta$$

That is, in this regime, no polynomial-time algorithm can distinguish the two distributions with probability at least $2/3$ of success.¹

To state our result, we recall that the SoS hierarchy is a family of convex semidefinite programming relaxations parameterized by an integer D_{SoS} called the *degree* or *level* of SoS. The relaxation gets tighter as D_{SoS} increases but the runtime also increases at the rate² of approximately $n^{O(D_{\text{SoS}})}$ for degree D_{SoS} SoS. Thus, conceptually degree $O(1)$ corresponds to polynomial time algorithms and degree n^δ corresponds to subexponential time algorithms. In this work, we study the performance of degree $D_{\text{SoS}} = n^\delta$ Sum-of-Squares on the Densest k -Subgraph problem for a constant $\delta > 0$ and obtain strong lower bounds.

Because of the well-known duality between SoS programs and pseudo-expectation operators, to show a lower bound, it suffices to show a feasible pseudo-expectation operator $\tilde{\mathbb{E}}$ satisfying the constraints. For a formal definition of SoS, see Section 2.1. We are now ready to state our result.

THEOREM 1.1. *For all constants $\alpha \in (0, 1/2]$, $\beta \in (0, 1)$, $\gamma \in (0, 1)$ such that $\gamma > \alpha\beta$, there exists $\delta > 0$ such that with high probability over $G = (V, E) \sim \mathcal{G}_{n,p}$, there exists a degree n^δ pseudo-expectation operator $\tilde{\mathbb{E}}$ on SoS program variables $\{\mathbf{X}_u\}_{u \in V}$ such that*

¹When $\alpha > \frac{1}{2}$, i.e. $k = \omega(\sqrt{n})$, spectral algorithms beat the log-density threshold [19, 66]. Spectral algorithms are captured by degree-2 SoS. Various works have also studied other special settings (e.g. when $q = 1$, or when p, q are constants). See Section 1.3.

²In pathological cases, there may be issues with bit complexity [91, 100]

- (1) (Normalization) $\tilde{\mathbb{E}}[1] = 1 \pm o(1)$.
- (2) (Subgraph on k vertices) $\tilde{\mathbb{E}}[\sum_{v \in V} \mathbf{X}_v] = k(1 \pm o(1))$.
- (3) (Large density) $\tilde{\mathbb{E}}[\sum_{\{u,v\} \in E} \mathbf{X}_u \mathbf{X}_v] = \frac{k^2 q}{2}(1 \pm o(1))$
- (4) (Feasibility) The moment matrix $\mathbf{M}$ corresponding to $\tilde{\mathbb{E}}$ is positive semidefinite.

This in particular implies that, in the predicted hard regime of the log-density framework, SoS cannot be used to solve the Densest k -Subgraph problem as stated above. As discussed earlier, these SoS lower bounds offer strong evidence that for $k \leq \sqrt{n}$, it is unlikely that efficient algorithms can beat the predictions of the log-density framework for Densest k -Subgraph.

By setting $\alpha = 1/2$, $\beta = 1/2$ and $\gamma = 1/4 + \epsilon$, we obtain the following important corollary.

Corollary 1.2. *For any $\epsilon > 0$, there exists a constant $\delta > 0$ such that degree- n^δ Sum-of-Squares exhibits an integrality gap of $O(n^{1/4-\epsilon})$ for the Densest k -Subgraph problem.*

This corollary essentially matches the best known algorithmic guarantees for the Densest k -subgraph problem [19], namely an efficient $O(n^{1/4+\epsilon})$ -factor approximation algorithm, thereby completing the picture for Sum-of-Squares.

1.2 Our Approach

Since Sum-of-Squares is a convex program, in order to prove a lower bound, it suffices to construct a feasible point, i.e. a *pseudoexpectation operator* or *moment matrix*, which is a large nonlinear random matrix that depends on the input. We remark that in the literature, the pseudoexpectation operator formulation is often referred to as the dual semidefinite program, but in this work, we exclusively work with this formulation. At a high level, our proof leverages an existing strategy for proving lower bounds against the Sum-of-Squares algorithm on random inputs: use *pseudocalibration* [17] to construct a candidate moment matrix, then study the spectrum of the candidate matrix using *graph matrices* [1]. This strategy has been successfully applied in several contexts [17, 48, 64, 95], although in each case, including ours, significant additional insights have been required.

Given a random input graph, the first step is to construct the candidate pseudoexpectation operator or moment matrix. Pseudocalibration suggests a candidate matrix, which we can use here without further thinking. Recall that a semidefinite program optimizes over the cone of positive semi-definite (PSD) matrices; the main challenge is showing that the candidate moment matrix is feasible (PSD) with high probability over the random input.

The main issue we face is that matrix factorization strategies in prior works do not obviously lead to dominant PSD terms in our setting. There are several steps in the existing framework:

- (1) Express the candidate moment matrix Λ in the graph matrix (i.e., Fourier) basis;
- (2) Identify a class of spectrally dominant graph matrices in Λ which are together approximately PSD;

- (3) Perform an approximate PSD decomposition to create PSD terms plus additional error terms;
- (4) Show that all non-dominant terms and error terms can be charged to the dominant PSD terms, i.e. they are “negligible”.

For the purposes of the current discussion, it is enough to know that each graph matrix in step (1) measures how a fixed small subgraph, or *shape*, contributes to the candidate moment matrix, and furthermore that the spectral norm of a graph matrix can be read off of combinatorial properties of the small shape graph. It was shown in [64, 103] that the norm of a graph matrix is determined up to lower-order factors by the *Sparse Minimum-weight Vertex Separator* (SMVS) of the shape (Theorem 2.20). For intuition, shapes with smaller, denser separators have larger norms.

In order to identify the class of norm-dominant shapes in step (2), previous work decomposes shapes using their leftmost and rightmost Minimum Vertex Separator (in contrast to SMVS), yielding for each shape an approximately PSD term that spectrally dominates the original graph matrix. Using the norm bounds, combinatorial arguments about vertex separators are then employed to show that all deviation terms in step (4) are small.

Prior work has avoided using the SMVS as the decomposition criterion, using the Minimum Vertex Separator (MVS) instead. However, the SMVS is a necessity in our setting, because Densest k -Subgraph is sensitive to small, local structures in the input. To explain, for a fixed set of vertices U , if many vertices in U have a common exterior neighbor or are part of a denser-than-average subgraph, then this greatly increases the algorithm’s belief that U is part of the dense subgraph. Using the SMVS can be thought of as pinpointing, for each shape, the small dense subgraph which has the strongest effect on the graph matrix’s norm.

Interestingly, a decomposition based on SMVS poses new conceptual challenges. For one, the SMVS is highly sensitive to the instantiation of the graph sample (which is to be expected, since this is exactly how small subgraphs can have outsize effect). Surprisingly, the SMVS decomposition, without extra care, may rather lead to some supposedly “PSD” terms being negative instead. We address these technical challenges, alongside our solution using the *Positive Minimum-weight Vertex Separator* (see Section 3.1 for a technical overview) after providing the definitions needed for working with graph matrices.

Once we have properly identified the dominant PSD terms, what remains is to prove that the error terms in the decomposition are small using an *intersection tradeoff lemma*. This is also one of our novel contributions as it is significantly different from intersection lemmas in prior works. This combinatorial lemma is the most crucial part of the proof, as it ensures that the error terms in the approximate PSD decomposition have small enough norms.

It’s worth highlighting that the log-density criterion $\gamma > \alpha\beta$ occurs multiple times throughout our proof, which is fascinating to the authors. A partial explanation is that if we look at the contribution of each Fourier character, the quantity $\gamma - \alpha\beta$ measures the decay as the degree of the Fourier character increases, i.e. it’s the edge decay in a shape. Therefore, this has a dampening effect

on the higher Fourier levels in the decomposition. Such a Fourier decay is ubiquitous in the analysis of the low-degree likelihood ratio [60, 62, 78] and has been important in prior average-case SoS lower bounds [17, 48, 64, 96].

1.3 Related Work

Algorithms. Algorithms for the Densest k -Subgraph problem have been widely studied, e.g. [4, 11, 18, 19, 22, 23, 44–46, 50, 66, 85, 108, 109], and we do not attempt to give an overview of them (see e.g. [66] for a nice overview of some of them). For general graphs, the work [72] (which also introduced the problem) gave a polynomial time $\tilde{O}(n^{0.3885})$ -factor approximation algorithm. This was later improved to a $O(n^{1/3-\epsilon})$ -factor approximation (for a constant $\epsilon \approx 1/60$) in [45] and to a $O(n^{0.3159})$ -factor approximation in [50] respectively. The seminal work of [19], which also proposed the log-density framework improved this to give an algorithm that achieves a $n^{1/4+\epsilon}$ -factor approximation in time $n^{O(1/\epsilon)}$, for all constants $\epsilon > 0$. This is conjectured to be the best achievable by efficient algorithms.

Lower bounds for Densest k -Subgraph. Because of its conceptual significance and wide applicability, studying lower bounds against the Densest k -Subgraph problem is an important research endeavour. We give a non-exhaustive list of such prior works below.

- (1) Conditional hardness: While it's well known that Densest k -Subgraph is NP-hard to solve exactly, to the best of our knowledge, NP-hardness of even constant factor approximation is unknown. That said, there are various other conditional hardness results assuming more than $P \neq NP$, e.g. [3, 23, 43, 67, 84, 99]. We highlight the influential work of Manurangsi [84], who assuming the Exponential Time Hypothesis showed almost-polynomial factor hardness for this problem. See the same paper for a more detailed list of other conditional hardness results. It's worth noting that none of these results achieve polynomial factor hardness. These approaches argue that Densest k -Subgraph is hard by reduction. One source of difficulty is that reductions are not as successful for average-case problems, since a reduction tends to distort the input distribution and produce somewhat pathological outputs. Proving hardness of Densest k -Subgraph may be possible using a reduction to a novel non-random instance, but, if it is true that random (or sufficiently pseudorandom) graphs are the *only* hard instances of Densest k -Subgraph, then a stronger theory of average-case reductions may be a prerequisite. Some recent works make exciting progress on realizing average-case reductions [21, 25, 27, 57].

The remaining lower bounds, including ours, are unconditional results that do not rely on any conjectures.

- (2) Sherali-Adams hardness: An integrality gap of $n^{\alpha(1-\alpha)-o(1)}$ was shown for the degree- $\tilde{O}(\log n)$ Sherali-Adams hierarchy (which is a family of linear programming relaxations) in [20, 39]. Our result is stronger than these Sherali-Adams lower bounds in three important ways. First, we consider SoS rather than Sherali-Adams. The SoS hierarchy captures

the Sherali-Adams hierarchy and is known to be much stronger in many cases (e.g., see [29, 31, 42, 68, 74] in conjunction with [10, 49]) so our result implies their result. Second, we obtain an n^δ degree lower bound as opposed to an $\tilde{O}(\log n)$ degree lower bound. Finally, while these Sherali-Adams lower bounds are for the particular setting where $\beta = \alpha$ (the setting that maximizes the integrality gap for a fixed α), our lower bounds work for the entire range of parameters α, β, γ .

- (3) SoS hardness: Worst-case SoS lower bounds have been exhibited in [20, 40, 85] obtained by reducing from Max k -CSP hardness results, within the SoS framework as pioneered by [111]. However, these SoS lower bounds were not optimal even for worst-case instances, since they didn't match known algorithmic guarantees (to be more precise, they showed an $n^{1/4-O(\epsilon)}$ -factor lower bound for degree n^ϵ SoS, whereas $n^{1/4-O(\epsilon)}$ -factor hardness is conjectured). Our work on the other hand studies average-case instances (as opposed to worst-case) and matches the guarantees of known algorithms. Therefore, we significantly improve these prior hardness results and close the gap. Moreover, our results can be reduced à la [111] to show SoS hardness for other problems such as Densest k -Subhypergraph [101, Theorem 3.17] and also potentially Minimum p -Union [101].

Average-case Sum-of-Squares lower bounds. Sum-of-Squares lower bounds for average-case problems have proliferated in the last decade, for example, Planted Clique [17, 59, 88], Sherrington-Kirkpatrick Hamiltonian [48, 76, 89], Sparse and Tensor PCA [60, 95, 96] and Max k -CSPs [73]. Most of these works have been in the colloquial “dense” regime where the random inputs are sampled from $\mathcal{G}_{n,1/2}$ or the standard normal distribution $\mathcal{N}(0, 1)$. Recently, average-case SoS lower bounds have been shown for the sparse setting, i.e. inputs sampled from $\mathcal{G}_{n,p}$ where $p = o(1)$, for the problem of Maximum Independent Set [64, 103]. The common thread underlying recent SoS lower bounds, including ours, is spectral analysis of large random matrices. See the works [63, 95, 102] for additional background and intuition on the matrix analysis framework used in these lower bounds.

The low-degree likelihood ratio hypothesis. We add that similar predictions as the log-density framework for the threshold of algorithmic distinguishability may possibly be obtained by analyzing the *low-degree likelihood ratio* [60, 62, 78]. The low-degree likelihood ratio is used in the context of noisy statistical inference problems to predict, among other things, the existence of statistical-computational gaps, i.e. when the signal (the planted dense subgraph) is information-theoretically detectable (and hence recoverable by a brute-force search), but is not detectable by efficient algorithms. In the same context, the low-degree likelihood ratio is used to predict the distinguishing power of low-degree polynomial algorithms. In [105], they analyze the low-degree likelihood ratio for certain parameter regimes of Densest k -Subgraph, but their results do not seem to recover the predictions of the log-density framework precisely. Our Proposition 2.31 can be interpreted as

showing that the low-degree likelihood ratio is $1+o(1)$ in the entire hard regime for the log-density framework.

Planted Dense Subgraph and Planted Clique conjectures. In our work, we have focused on the regime $\alpha \in (0, 1/2]$, $\beta, \gamma \in (0, 1)$. Other instantiations of these parameters have also been subject to intense study in recent years and various conjectures predicting the limits of efficient algorithms have been proposed, broadly referred to as the Planted Dense Subgraph conjecture or in the case $\gamma = 0$, the Planted Clique conjecture. Furthermore, assuming these conjectures, inapproximability results have been derived for various problems such as Sparse PCA, Stochastic Block model, Biclustering, etc. See e.g. [24, 27, 28, 34, 52, 86, 95, 96] and references therein. Densest k -Subgraph lies at the heart of many of these reductions, therefore it's plausible that our hardness result can be exploited to derive better inapproximability results for various other problems, which we leave for future work.

1.4 Organization of the Paper

The rest of the paper is organized as follows. We start with a brief overview in Section 2 of graph matrices, which are at the heart of our spectral analysis, using it to construct our candidate moment matrix following the pseudo-calibration framework in Section 2.5. With the matrix in hand, we then delve into the extensive PSDness analysis that forms the bulk of work. We motivate and discuss our conceptually novel PMVS decomposition in Section 3. We defer the formal details and other technical verifications to the full version of the paper.

2 PRELIMINARIES

2.1 The Sum-of-Squares Algorithm

We now formally describe the Sum-of-Squares hierarchy. For a detailed treatment and survey of SoS, see e.g. [47, 62, 63, 98, 104].

SoS is used to check feasibility of a system of polynomials. Given a graph $G = (V, E)$, the simplest polynomial formulation for the existence of a subgraph with k vertices and m edges encodes the 0/1 indicator of the subgraph:

$$\begin{aligned} \text{Variables: } & \mathbf{X}_v, \forall v \in V \\ \text{Constraints: } & \\ & \sum_{v \in V} \mathbf{X}_v = k \quad (\text{Vertex count}) \\ & \sum_{\{u,v\} \in E} \mathbf{X}_u \mathbf{X}_v = m \quad (\text{Edge count}) \\ & \mathbf{X}_v^2 = \mathbf{X}_v \quad \forall v \in V \quad (\text{Boolean}) \end{aligned}$$

The sum-of-squares algorithm is parameterized by the degree $D_{\text{SoS}} \in \mathbb{N}$. The nontrivial levels of the algorithm are $D_{\text{SoS}} = 2, 4, 6, 8, \dots$

Definition 2.1 (Pseudoexpectation). *Given a set of variables $\mathbf{X}_1, \dots, \mathbf{X}_n$, a degree- D_{SoS} pseudoexpectation operator is a linear functional $\tilde{\mathbb{E}} : \mathbb{R}^{\leq D_{\text{SoS}}}[\mathbf{X}_1, \dots, \mathbf{X}_n] \rightarrow \mathbb{R}$ such that $\tilde{\mathbb{E}}[1] = 1$.*

Here $\mathbb{R}^{\leq D_{\text{SoS}}}[\mathbf{X}_1, \dots, \mathbf{X}_n]$ denotes the set of polynomials with degree at most D_{SoS} .

Definition 2.2 (Satisfying an equality constraint). *A degree- D_{SoS} pseudoexpectation operator $\tilde{\mathbb{E}}$ satisfies a polynomial constraint “ $f(\mathbf{X}) = 0$ ” if $\tilde{\mathbb{E}}[f(\mathbf{X})p(\mathbf{X})] = 0$ for all polynomials $p(\mathbf{X})$ of degree at most D_{SoS} .*

Definition 2.3 (SoS-feasible). *A degree- D_{SoS} pseudoexpectation operator $\tilde{\mathbb{E}}$ is SoS-feasible if for every polynomial $p \in \mathbb{R}^{\leq D_{\text{SoS}}/2}$, $\tilde{\mathbb{E}}[p(\mathbf{X})^2] \geq 0$.*

Definition 2.4 (Sum-of-squares algorithm). *Given a system of polynomial constraints $\{f_i(\mathbf{X}) = 0\}$ in n variables $\mathbf{X}_1, \dots, \mathbf{X}_n$, the degree- D_{SoS} Sum-of-Squares algorithm checks for the existence of an SoS-feasible degree- D_{SoS} pseudoexpectation operator $\tilde{\mathbb{E}}$ that satisfies the constraints. If $\tilde{\mathbb{E}}$ exists, the algorithm outputs “may be feasible”, otherwise it outputs “infeasible”. This can be done algorithmically by solving a semidefinite program of size $n^{O(D_{\text{SoS}})}$ that searches for a feasible moment matrix (Definition 2.7).*

If no pseudoexpectation operator exists, then SoS successfully refutes the polynomial system (i.e., it proves that there is no dense subgraph in the input). On the other hand, if a pseudoexpectation operator exists, SoS cannot rule out that the polynomial system is feasible (the pseudoexpectation operator fools SoS, but it may or may not correspond to a true distribution on feasible points). A lower bound against SoS consists of a feasible pseudoexpectation operator in the case when the system is actually infeasible.

2.2 Moment Matrices

Analysis of the SoS algorithm on an n -variable polynomial system is typically accomplished by formulating it in terms of large matrices indexed by subsets of $[n]$, known as *moment matrices*.

Definition 2.5 (Matrix index). *Let $\mathcal{I}$ be the set of ordered subsets of $[n]$ of size at most $D_{\text{SoS}}/2$.*

Remark 2.6. *Another reasonable definition of $\mathcal{I}$ uses subsets of $[n]$ and not ordered subsets. For technical reasons, we include an ordering.*

The degree- D_{SoS} sum-of-squares algorithm can be equivalently formulated in terms of $\mathbb{R}^{\mathcal{I} \times \mathcal{I}}$ matrices, which are called *moment matrices*.

Definition 2.7 (Moment matrix). *The moment matrix $\Lambda = \Lambda(\tilde{\mathbb{E}})$ associated to a degree- D_{SoS} pseudoexpectation $\tilde{\mathbb{E}}$ is an $\mathcal{I}$ -by- $\mathcal{I}$ matrix defined as*

$$\Lambda[I, J] := \tilde{\mathbb{E}}[\mathbf{X}^I \cdot \mathbf{X}^J].$$

Fact 2.8. *$\tilde{\mathbb{E}}$ is SoS-feasible if and only if $\Lambda(\tilde{\mathbb{E}}) \succeq 0$.*

Definition 2.9 (SoS-symmetric). *A matrix $\Lambda \in \mathbb{R}^{\mathcal{I} \times \mathcal{I}}$ is SoS-symmetric if $\Lambda[I, J]$ depends only on the disjoint union $I \sqcup J$ as an unordered multiset. Along with the additional constraint $\Lambda[\emptyset, \emptyset] = 1$, this characterizes $\Lambda \in \mathbb{R}^{\mathcal{I} \times \mathcal{I}}$ which are moment matrices of degree- D_{SoS} pseudoexpectation operators.*

In the presence of Boolean constraints “ $\mathbf{X}_i^2 = \mathbf{X}_i$ ”, a moment matrix satisfies these constraints if and only if $\Lambda[I, J]$ depends only on the union $I \cup J$ as an unordered set (ignore duplicates).

2.3 p -biased Fourier Analysis and Graph Matrices

We are interested in matrices which depend on a random graph $G \sim \mathcal{G}_{n,p}$. To analyze these as functions of G , we encode G via its edge indicator vector in $\{0, 1\}^{\binom{n}{2}}$ and perform p -biased Fourier analysis.

Definition 2.10 (Fourier character). χ denotes the p -biased Fourier character,

$$\chi(0) = -\sqrt{\frac{p}{1-p}}, \quad \chi(1) = \sqrt{\frac{1-p}{p}}. \quad (1)$$

For H a subset or multi-subset of $\binom{[n]}{2}$, let $\chi_H(G) := \prod_{e \in H} \chi(G_e)$.

Definition 2.11 (Ribbon). A ribbon is a tuple $R = (A_R, B_R, E(R))$ where $A_R, B_R \in \mathcal{I}$ and $E(R) \subseteq \binom{[n]}{2}$. The corresponding matrix $\mathbf{M}_R \in \mathbb{R}^{\mathcal{I} \times \mathcal{I}}$ is:

$$\mathbf{M}_R[I, J] = \begin{cases} \chi_{E(R)}(G) & I = A_R, J = B_R \\ 0 & \text{otherwise.} \end{cases}$$

The ribbon matrices $\mathbf{M}_R$ are mean-zero, orthonormal under the expectation of the Frobenius inner product on matrices, and form a basis for all $\mathbb{R}^{\mathcal{I} \times \mathcal{I}}$ -valued functions of G . They are the natural Fourier basis for random matrices that depend on G .

In the matrices that we study, the coefficient on a ribbon will not depend on the particular labels of the ribbon's vertices, but only on the graphical structure of the ribbon. This graphical structure is called the *shape*.

Definition 2.12 (Shape). A shape α is an equivalence class of ribbons under relabeling of the vertices (equivalently, permutation by S_n). Each shape is associated with a representative graph $(U_\alpha, V_\alpha, E(\alpha))$. We let $V(\alpha) := U_\alpha \cup V_\alpha \cup V(E(\alpha))$.

We use the convention of Greek letters such as α, γ, τ for shapes and Latin letters R, L, T for ribbons.

Definition 2.13 (Embedding). Given a shape α and an injective function $\varphi : V(\alpha) \rightarrow [n]$, we let $\varphi(\alpha)$ be the ribbon obtained by labeling α in the natural way (preserving the order on U_α and V_α).

A ribbon R has shape α if and only if R can be obtained by an embedding of $V(\alpha)$ into $[n]$. Note that different embeddings may produce the same ribbon.

Definition 2.14 (Graph matrix). Given a shape α , the graph matrix $\mathbf{M}_\alpha$ is

$$\mathbf{M}_\alpha = \sum_{\text{injective } \varphi: V(\alpha) \rightarrow [n]} \mathbf{M}_{\varphi(\alpha)}.$$

The entries of a graph matrix are degree- $|E(\alpha)|$ monomials in the variables G_e , therefore we think of graph matrices as low-degree polynomial random matrices in G . We call them “nonlinear” to distinguish them from the degree-1 case, which is well-studied (being essentially the adjacency matrix of G).

Definition 2.15 (Diagonal). A ribbon or shape α is diagonal if $V(\alpha) = U_\alpha = V_\alpha$.

A diagonal shape is only nonzero on the diagonal entries of the matrix in the block corresponding to U_α . Note that there are additional shapes which have the same support, namely shapes which potentially have additional edges and vertices outside of $U_\alpha = V_\alpha$. The diagonal shapes as we have defined them are the most important contributors to the diagonal entries of the matrix.

Definition 2.16 (Transpose). The transpose of a ribbon or shape swaps A_R, B_R or U_α, V_α respectively. This has the effect of transposing the matrix for the ribbon/shape.

2.4 Norm Bounds

Definition 2.17 (Weight of a set). For a graph S , let $w(S) = |V(S)| - \log_n(1/p)|E(S)|$.

Definition 2.18 (Vertex separator). A vertex separator of two sets A, B in a graph G is a set $S \subseteq V(G)$ such that all paths from A to B pass through S .

Definition 2.19 (Sparse minimum vertex separator (SMVS)). Given a ribbon or shape α , a sparse minimum vertex separator (SMVS) is a minimizer of $w(S)$ over $S \subseteq V(\alpha)$ which separate U_α and V_α .

THEOREM 2.20 (NORM BOUND, INFORMAL [64, 103]). With high probability, for all proper shapes α :

$$\|\mathbf{M}_\alpha\| \leq \tilde{O}\left(n^{\frac{|V(\alpha)| - |w(S_{\min})|}{2}}\right)$$

where $S_{\min}$ is the SMVS of α .

2.5 Pseudocalibration

Pseudocalibration is a heuristic used to construct candidate pseudoexpectation operators $\tilde{\mathbb{E}}$ for SoS lower bounds, introduced in the context of SoS lower bounds for Planted Clique [17]. See e.g. [17, 48, 98] for a formal description.

The pseudocalibrated operator $\tilde{\mathbb{E}}[\mathbf{X}^I]$ is defined using the Fourier coefficients of the corresponding function $\mathbf{X}^I(H)$ evaluated on the planted distribution. First we need to compute these Fourier coefficients. A similar computation was performed by [39] to exhibit integrality gaps for the Sherali-Adams hierarchy.

Lemma 2.21. Let $\mathbf{X}^I(H)$ be the 0/1 indicator function for I being in the planted solution i.e. $I \subseteq H$. Then, for all $I \subseteq [n]$ and $\alpha \subseteq \binom{[n]}{2}$,

$$\mathbb{E}_{(G,H) \sim \mathcal{D}_{pl}}[\mathbf{X}^I(H) \cdot \chi_\alpha(\tilde{G})] = \left(\frac{k}{n}\right)^{|V(\alpha) \cup I|} \left(\frac{q-p}{\sqrt{p(1-p)}}\right)^{|E(\alpha)|}$$

PROOF. First observe that if any vertex of $V(\alpha) \cup I$ is outside H , then the expectation is 0. This is because either I is outside H , in which case $\mathbf{X}^I(H) = 0$, or an edge of α is outside H , in which case the expectation of this Fourier character is 0. Now, each vertex of $V(\alpha) \cup I$ is in H independently with probability $\frac{k}{n}$. Conditioned on this event happening, each edge independently evaluates to

$$\mathbb{E}_{e \sim \text{Bernoulli}(q)} \chi(e) = q \cdot \chi(1) + (1-q) \cdot \chi(0) = \frac{q-p}{\sqrt{p(1-p)}}.$$

Putting these together gives the result. $\blacksquare$

For technical convenience, we slightly modify the parameters β and γ so that $\frac{p}{1-p} = n^{-\beta}$ and $\frac{q-p}{1-p} = n^{-\gamma}$.

Pseudocalibration suggests transferring the low-degree Fourier coefficients from the planted distribution. As long as the size of the Fourier coefficients is larger than the SoS degree, i.e. $D_V \gg D_{\text{SoS}}$, then SoS should not notice that we are using a truncation.

Definition 2.22 (Additional parameters). *Let $\varepsilon = \gamma - \alpha\beta$ be the slack in the log-density equations.*

Let $D_V = C \cdot D_{\text{SoS}} \log n$ where constant $C = C(\varepsilon)$ is sufficiently large.

Definition 2.23 ($\mathcal{S}$). *Let $\mathcal{S}$ be the set of (proper) ribbons R such that:*

- (i) (**Degree bound**) $|A_R|, |B_R| \leq D_{\text{SoS}}/2$
- (ii) (**Size bound**) $|V(R)| \leq D_V$

We will sometimes use $\alpha \in \mathcal{S}$ as the set of shapes with the same properties, following the convention of using Latin letters for ribbons and Greek letters for shapes.

Definition 2.24 ($\mathbf{M}$). *Define the pseudocalibrated candidate moment matrix*

$$\mathbf{M} = \sum_{R \in \mathcal{S}} \left(\frac{k}{n} \right)^{|V(R)|} \left(\frac{q-p}{\sqrt{p(1-p)}} \right)^{|E(R)|} \mathbf{M}_R$$

For the purposes of analyzing the spectrum of $\mathbf{M}$ in later sections, it is more convenient to rescale the entries so that $\tilde{\mathbb{E}}[\mathbf{X}^I]$ has order 1 for all $I \subseteq [n]$.

Definition 2.25 (λ_α). *Given a shape or ribbon α , let*

$$\begin{aligned} \lambda_\alpha &= \left(\frac{k}{n} \right)^{|V(\alpha)| - \frac{|U_\alpha| + |V_\alpha|}{2}} \left(\frac{q-p}{\sqrt{p(1-p)}} \right)^{|E(\alpha)|} \\ &= n^{(\alpha-1) \left(|V(\alpha)| - \frac{|U_\alpha| + |V_\alpha|}{2} \right) + \left(\frac{\beta}{2} - \gamma \right) |E(\alpha)|} \end{aligned}$$

Definition 2.26 (properly composable, informal). *We say that a sequence of ribbons $R_1, \dots, R_k$ is properly composable if for all $j \in [k-1]$, $B_{R_j} = A_{R_{j+1}}$ and these ribbons have no other vertices in common.*

Lemma 2.27. *If R, S are properly composable ribbons, then $\lambda_{R \circ S} = \lambda_R \lambda_S$.*

Definition 2.28 (Λ). *Define $\Lambda = \sum_{R \in \mathcal{S}} \lambda_R \mathbf{M}_R$.*

Lemma 2.29. $\mathbf{M} \geq 0$ if and only if $\Lambda \geq 0$.

PROOF. We have $\mathbf{M} = \mathbf{DAD}$ where $\mathbf{D}$ is a diagonal matrix with positive entries $D[I, I] = \left(\frac{k}{n} \right)^{\frac{|I|}{2}}$. Hence $x^\top \mathbf{M} x \geq 0$ for all $x \in \mathbb{R}^I$ if and only if $x^\top \Lambda x \geq 0$ for all $x \in \mathbb{R}^I$. ■

Lemma 2.30. $\mathbf{M}$ is SoS-symmetric and satisfies the constraints " $\mathbf{X}_i^2 = \mathbf{X}_i$ ".

Proposition 2.31. *With high probability, we have $\tilde{\mathbb{E}}[1] = 1 \pm o(1)$.*

Remark 2.32. *While the definition of $\tilde{\mathbb{E}}$ requires that $\tilde{\mathbb{E}}[1] = 1$, this can be achieved by taking the pseudo-expectation values given by pseudo-calibration and dividing them by $\tilde{\mathbb{E}}[1]$.*

Proposition 2.33. *With high probability,*

$$\sum_{i=1}^n \tilde{\mathbb{E}}[\mathbf{X}_i] \geq (1 \pm o_n(1))k,$$

and

$$\sum_{(i,j) \in E(G)} \tilde{\mathbb{E}}[\mathbf{X}_i \mathbf{X}_j] \geq (1 \pm o_n(1)) \frac{k^2 q}{2}.$$

3 POSITIVE MINIMUM VERTEX SEPARATOR DECOMPOSITION

3.1 Motivation for the Positive Minimum Vertex Separator

After pseudocalibration, to complete the proof of [Theorem 1.1](#), we need to show that the rescaled candidate moment matrix is PSD with high probability,

$$\Lambda = \sum_{\alpha \in \mathcal{S}} \lambda_\alpha \cdot \mathbf{M}_\alpha \geq 0.$$

For each graph matrix $\lambda_\alpha \mathbf{M}_\alpha$ in Λ , we want to find an (approximately) PSD term which spectrally dominates it. Previous work led to the following idea: for each shape α , we can split it across the leftmost and rightmost minimum vertex separators so that α is decomposed into three parts,

$$\alpha = \sigma \circ \tau \circ \sigma'^\top.$$

Then the target spectral upper bound is given by

$$\lambda_\sigma^2 \mathbf{M}_{\sigma \circ \sigma^\top} + \lambda_{\sigma'}^2 \mathbf{M}_{\sigma' \circ \sigma'^\top}.$$

This is approximately PSD since $\mathbf{M}_{\sigma \circ \sigma^\top} \approx \mathbf{M}_\sigma \mathbf{M}_\sigma^\top \geq 0$. To make this strategy work, we need to prove that the middle shape $\mathbf{M}_\tau$ is spectrally dominated by the corresponding identity via combinatorial charging. In previous work, it has been essentially possible to charge all middle shapes to the identity matrix, but this breaks down in the setting of Densest k -Subgraph. In the baby case, this is evident in our calculation for $\sum_{(u,v) \in E(G)} \tilde{\mathbb{E}}[\mathbf{X}_u \mathbf{X}_v]$ in [Proposition 2.33](#), where the dominant term is no longer the trivial shape but instead the shape with an edge in between.

A second, related issue is the presence of edges inside the separator. Concretely, say that $(U_\tau, E(U_\tau))$ and $(V_\tau, E(V_\tau))$ are the leftmost/rightmost SMVS of a middle shape τ , and we hope to charge τ to the diagonal matrix corresponding to the leftmost/rightmost SMVS. Concretely, letting U_τ also denote the diagonal shape with edges $E(U_\tau)$, we want to charge

$$\lambda_\tau (\mathbf{M}_\tau + \mathbf{M}_\tau^\top) \leq \lambda_{U_\tau} \mathbf{M}_{U_\tau} + \lambda_{V_\tau} \mathbf{M}_{V_\tau}.$$

However, this strategy crucially requires that $\lambda_{U_\tau} \cdot \mathbf{M}_{U_\tau}$ and $\lambda_{V_\tau} \cdot \mathbf{M}_{V_\tau}$ are PSD by themselves in order to conclude that the result is PSD. Since λ_α is non-negative, this boils down to the PSD-ness of the diagonal shape $(U_\tau, E(U_\tau))$ for the SMVS. This latter matrix is easy to verify as the non-zero diagonal entries are given by, for a ribbon R of the corresponding shape U_τ ,

$$\chi_{E(R)}(G) = \prod_{e \in E(R)} \chi_e(G)$$

and recall that we are working on the p -biased Fourier basis,

$$\chi_e(1) = \sqrt{\frac{1-p}{p}}, \quad \chi_e(0) = -\sqrt{\frac{p}{1-p}}$$

At this point, we observe that the instantiation of the SMVS edges $E(R)$ plays a crucial role as they determine whether our candidate "PSD" mass is truly positive. If all edges of $E(R)$ are present in G , then the diagonal entry is positive,

$$\prod_{e \in E(R)} \chi_e(G) = \sqrt{\frac{1-p}{p}}^{|E(R)|} \geq 0.$$

On the other hand, if an edge is missing, then positivity is not guaranteed. Ignoring this bad case for now, we have the following sufficient criterion for finding a PSD dominant term. If T is a ribbon of shape τ , and R is the restricted ribbon to U_τ , then if $E(R) \subseteq E(G)$, we must charge $\lambda_\tau \mathbf{M}_T$ to $\lambda_{U_\tau} \mathbf{M}_R$.

When an edge is missing inside the SMVS, then we need to look harder. Despite the candidate PSD term not being truly positive, it is not yet time to panic. In this case, (1) a missing edge scales down the matrix, in line with the intuition that subgraphs with edges present are the highest-norm terms, therefore (2) we look in the remainder of the shape for the new SMVS, to determine the new matrix norm. This creates a recursive process, and when all edges inside the candidate SMVS are actually present in the graph, we terminate, calling this the *Positive Minimum-weight Vertex Separator* (PMVS).

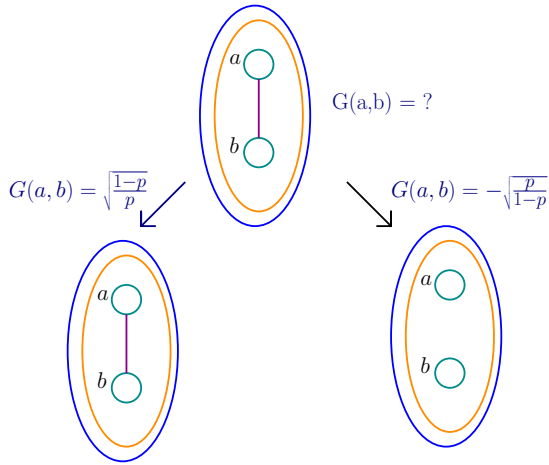


Figure 1: PMVS Search

Let us give an example. The graph matrix at the top of Fig. 1 appears on the diagonal of our moment matrix. In this example shape, the only vertex separator is the entire shape, and so the SMVS contains the edge $G(a, b)$. We check whether or not the edge appears in the graph. In the "yes" outcome on the left, we have a PSD matrix whose (a, b) -th diagonal entry is $\mathbf{1}_{(a,b) \in E(G)} \sqrt{\frac{1-p}{p}}$. In the "no" outcome on the right, the (a, b) -th diagonal entry is $-\mathbf{1}_{(a,b) \notin E(G)} \sqrt{\frac{p}{1-p}}$, which is negative and therefore the matrix is not PSD. This matrix

comes with a small coefficient of approximately $\sqrt{p}$ and hence it can be charged to the corresponding identity matrix, whose (a, b) -th diagonal entry is just 1. In this example, the recursion terminates after just one step, but in larger shapes, we would need to find the new SMVS for the case on the right.

As described above, the recursion outputs the minimizer of a certain weight function. However, we need to slightly modify the recursive process described above so that it always "moves left", in order for the crucial Remark 3.7 to hold.

3.2 PMVS Subroutine

We make the following extended definition of the *Positive Minimum Vertex Separator* (PMVS) of a ribbon R .

Definition 3.1 (Left and right indicators). *We say that a ribbon R has left indicators if R has edge indicators for every edge $e \in E(A_R)$. Similarly, we say that a ribbon R has right indicators if R has edge indicators for every edge $e \in E(B_R)$.*

Our goal is to have composable triples of ribbons R_1, R_2, R_3 with the following properties:

Definition 3.2 (Ribbons with PMVS identified). *A composable triple of ribbons R_1, R_2, R_3 has PMVS identified if:*

- (i) R_1 is a left ribbon and R_3 is a right ribbon.
- (ii) R_1, R_2, R_3 are properly composable.
- (iii) R_1 has right indicators, R_2 has both left and right indicators, and R_3 has left indicators.
- (iv) The edges and edge indicators agree on $B_{R_1} = A_{R_2}$ and $B_{R_2} = A_{R_3}$.
- (v) R_1, R_2, R_3 have no other edge indicators.

When these properties hold, we say that the left PMVS is A_{R_2} and the right PMVS is B_{R_2} .

Remark 3.3. *The left and right PMVS may not have the same size or weight. In fact, they may not even be an SMVS of R_2 . We will bound the difference between the PMVS and the SMVS.*

At the beginning, we take each ribbon R and decompose it into ribbons R_1, R_2, R_3 based on the leftmost and rightmost SMVS. This gives us a composable triple of ribbons R_1, R_2, R_3 such that

- (i) R_1 is a left ribbon, R_2 is a middle ribbon, and R_3 is a right ribbon.
- (ii) R_1, R_2, R_3 are properly composable.
- (iv) The edges agree on $B_{R_1} = A_{R_2}$ and $B_{R_2} = A_{R_3}$.
- (v) R_1, R_2 , and R_3 have no edge indicators.

Remark 3.4. *The ribbon encoded by the triple R_1, R_2, R_3 is $(R_1 \setminus E(B_{R_1})) \circ R_2 \circ (R_3 \setminus E(A_{R_3}))$ rather than $R_1 \circ R_2 \circ R_3$ because edges inside $B_{R_1} = A_{R_2}$ should not be duplicated.*

In order to satisfy the condition that R_1 has right indicators, R_2 has both left and right indicators, and R_3 has left indicators, we repeat the following sequence of operations as many times as needed.

- (1) **Adding left and right indicators operation:** To add indicators to $B_{R_1} = A_{R_2}$ and $B_{R_2} = A_{R_3}$, we replace each edge $e \in E(A_{R_2}) \cup E(B_{R_2})$ that does not yet have an indicator using³ the equation $\chi_e = \frac{1}{1-p} \mathbf{1}_{e \in E(G)} \chi_e - \sqrt{\frac{p}{1-p}}$. This leads to two possible new ribbons which have different edge structure, one with e still present and the other with e removed.
- (2) **PMVS operation:** After adding the edge indicators to $B_{R_1} = A_{R_2}$ and $B_{R_2} = A_{R_3}$, we check if R_1 is still a left ribbon and R_3 is still a right ribbon. If so, we stop and exit the loop. If not, we let A' be the leftmost SMVS separating A_{R_1} from B_{R_1} and we let B' be the rightmost SMVS separating A_{R_3} from B_{R_3} . We then replace R_1, R_2 , and R_3 with the ribbons R'_1, R'_2 , and R'_3 where
 - (a) R'_1 is the part of R_1 between A_{R_1} and A' .
 - (b) R'_2 is the composition of the part of $R_1 \setminus E(B_{R_1})$ between A' and B_{R_1} , R_2 , and the part of $R_3 \setminus E(A_{R_3})$ between A_{R_3} and B' .
 - (c) R'_3 is the part of R_3 between B' and B_{R_3} .
- (3) **Removing middle edge indicators operation:** If R_2 has one or more edge indicators which are now outside of A_{R_2} and B_{R_2} , we re-convert them back into Fourier characters using the equation $\frac{1}{1-p} \mathbf{1}_{e \in E(G)} \chi_e = \sqrt{\frac{p}{1-p}} + \chi_e$.

We call this repeated sequence of operations the **Finding PMVS subroutine**, which takes a triple of composable ribbons R_1, R_2, R_3 which have all the needed properties except having left and right indicators (some but not all indicators may be present) and gives us a triple of composable ribbons with all of the needed properties.

Remark 3.5. Note that each triple R_1, R_2, R_3 leads to many triples R'_1, R'_2, R'_3 depending on which summand is taken in each equation. The recursion proceeds on every term except for the one in which every χ_e is replaced by $\frac{1}{1-p} \mathbf{1}_{e \in E(G)} \chi_e$.

Remark 3.6. At first glance, checking whether or not edges inside A_{R_2} and B_{R_2} are present leads to a complicated dependence on the input graph G . In order to mathematically express the recursion in a G -independent way, we formally use the edge indicator function to express the two cases.

3.3 Intersection Term Operation

Once we have these triples of ribbons R_1, R_2, R_3 , we can apply an approximate factorization across the PMVS. When we do this, we will obtain error terms which can be described by triples of ribbons R_1, R_2 , and R_3 which have at least one non-trivial intersection (they are not properly composable) but satisfy the other four properties in [Definition 3.2](#). We handle this as follows.

- (1) **Intersection term decomposition operation:** Let A' be the leftmost SMVS between A_{R_1} and $V_{\text{intersected}}(R_1) \cup B_{R_1}$ and let B' be the rightmost SMVS between $V_{\text{intersected}}(R_3) \cup$

A_{R_3} and B_{R_3} . We now replace R_1, R_2 , and R_3 with the ribbons R'_1, R'_2 , and R'_3 where

- (a) R'_1 is the part of R_1 between A_{R_1} and A' .
- (b) To obtain R'_2 , we improperly compose the part of $R_1 \setminus E(B_{R_1})$ between A' and B_{R_1} , R_2 , and the part of $R_3 \setminus E(A_{R_3})$ between A_{R_3} and B' . We then linearize the multi-edges, replacing $\chi_e^k = c_0 + c_1 \chi_e$ using the appropriate coefficients c_0, c_1 .

In the edge case that a multi-edge also has an edge indicator (for example, because an edge inside A_{R_2} intersects with an edge from R_3), we instead use the equation

$$\mathbf{1}_e \chi_e^k = \left(\sqrt{\frac{1-p}{p}} \right)^{k-1} \mathbf{1}_e \chi_e.$$

- (c) R'_3 is the part of R_3 between B' and B_{R_3} .
- (2) We apply the **Removing middle edge indicators operation** to R_2 .

The ribbon R'_2 is defined to “grow” R_2 so that it includes the intersections. After these steps, we are in essentially the same situation as we started. More precisely, we have a triple of ribbons R_1, R_2, R_3 such that

- (i) R_1 is a left ribbon and R_3 is a right ribbon.
- (ii) R_1, R_2, R_3 are properly composable.
- (iv) The edges and edge indicators agree on $B_{R_1} = A_{R_2}$ and $B_{R_2} = A_{R_1}$.
- (v) R_1, R_2, R_3 have no edge indicators outside of $B_{R_1} = A_{R_2}$ and $B_{R_2} = A_{R_3}$.

At this point, we can repeat the operations, applying the **Finding PMVS subroutine** to identify a new PMVS, approximately factoring, then decomposing intersection terms, as many times as needed.

3.4 Summary of the Operations and Overall Decomposition

We now summarize our procedure.

Finding PMVS subroutine: repeat the following until convergence,

- (1) Apply the **Adding left and right indicators operation** to add indicators to $B_{R_1} = A_{R_2}$ and $B_{R_2} = A_{R_3}$.
- (2) Apply the **PMVS operation** to ensure that R_1 is a left ribbon and R_3 is a right ribbon. If no change is made to R_1 or R_3 , then we have identified the PMVS.
- (3) Apply the **Removing middle edge indicators operation** to R_2 to ensure that R_2 has no middle indicators.

Overall decomposition procedure:

- (1) We start with triples of composable ribbons R_1, R_2, R_3 which have all the needed properties except having left and right indicators.
- (2) We apply the **Finding PMVS subroutine**.
- (3) Recursive factorization: We apply the following procedure repeatedly until there are no more error terms.
 1. We approximate the sum over the composable triples of ribbons R_1, R_2, R_3 by enlarging the sum to include all left

³The high-level overview of the PMVS alluded to the slightly different formula $\chi_e = \mathbf{1}_{e \in E(G)} \chi_e + \mathbf{1}_{e \notin E(G)} \chi_e$. These are morally equivalent, but the formula here is simpler to analyze.

ribbons R_1 and right ribbons R_3 (not necessarily properly composable with R_2 or with each other). This yields a matrix $\mathbf{LQ}_i\mathbf{L}^\top$ where $\mathbf{L}$ sums over left ribbons and $\mathbf{Q}_i$ sums over the ribbons R_2 on the i th iteration of the loop. We then move to the triples of ribbons R_1, R_2, R_3 for the intersection error terms, if any.⁴

2. We apply the **Intersection term decomposition operation** to obtain a triple of ribbons R_1, R_2, R_3 which are properly composable.
3. We apply the **Removing middle edge indicators operation** to R_2 to ensure that R_2 has no middle indicators.
4. We apply the **Finding PMVS subroutine**.

Remark 3.7. As with previous SoS lower bounds using graph matrices, a key observation is that the PMVS operation and the intersection term decomposition operation are unaffected by replacing R'_1 with a different left ribbon R''_1 or replacing R'_3 with a different right ribbon R''_3 as long as $B_{R''_1} = B_{R'_1}$ and $A_{R''_3} = A_{R'_3}$. This ensures that all left ribbons R'_1 and right ribbons R'_3 appear in the matrices $\mathbf{L}$ and $\mathbf{L}^\top$.

Carrying out this process, the overall decomposition of the moment matrix is then

$$\Lambda = \mathbf{L} \left(\sum_{i=0}^{2D_V} \mathbf{Q}_i \right) \mathbf{L}^\top + \text{truncation error}.$$

Therefore, the main requirement for $\Lambda \geq 0$ is to show that $\sum_{i=0}^{2D_V} \mathbf{Q}_i \geq 0$. We will show that the norm-dominant terms are the diagonal shapes (Definition 2.15). By virtue of the PMVS factorization, these shapes are PSD, as we can easily check.

Lemma 3.8. If R_1, R_2, R_3 are ribbons with PMVS identified, such that R_2 is diagonal, then $\lambda_{R_2} \mathbf{M}_{R_2} \geq 0$.

PROOF. $\lambda_{R_2} \geq 0$ and R_2 is diagonal with one nonzero entry, so we need that the entry is nonnegative. Since R_2 has edge indicators, the entry is

$$\prod_{e \in E(R_2)} \mathbf{1}_{e \in E(G)} \chi_e(G).$$

Any time the entry is nonzero, its value is

$$\chi(1)^{|E(R_2)|} = \left(\sqrt{\frac{1-p}{p}} \right)^{|E(R_2)|} \geq 0. \quad \blacksquare$$

4 CONCLUSION

In this work, we showed Sum-of-Squares lower bounds for Densest k -Subgraph. Our results lend strength to the conjecture that Densest k -Subgraph is truly a hard problem in the predicted “hard” parameter regime. Our results are in line with the log-density framework for Densest- k -Subgraph, complementing the extraordinary work of [19] from over a decade ago.

⁴There are also additional error terms for the truncation error, as the maximum size of the left ribbons R_1, R_3 will be slightly smaller for intersection terms. This must be handled separately.

Our work provides a formal lower bound against a concrete class of algorithms for Densest k -Subgraph. For the optimistic algorithm designer that wishes to solve Densest k -Subgraph, what kind of algorithms could circumvent our lower bound? First, one could try to modify the constraints or objective of the semi-definite program. For example, “mismatching” the size of the hidden subgraph may be helpful for the related Planted Clique problem [6]. Our proof does not formally rule out non-standard SDP-based algorithms, although we believe it is likely that our proof could be modified into a lower bound against other SDPs. Second, algebraic approaches based on finite fields, Gaussian elimination, or lattice-based methods are not captured by Sum-of-Squares reasoning [112]. However, these techniques typically require a rigid “noise-free” structure in the problem which isn’t present in Densest k -Subgraph, so such an algorithm would be unexpected.

There are some technical limitations to our work, which are also present in almost all existing SoS lower bounds. Technical improvements such as improving the SoS degree from n^ϵ to $\tilde{\Omega}(k)$, or tightening the slack $\gamma - \alpha\beta$ seem out of reach for our current techniques. We could also consider the closely related planted model where the size of the planted subgraph is not approximately but exactly k . Our analysis doesn’t go through immediately in this setting for technical reasons, which is also the case in most existing SoS lower bounds. With additional work, this might be overcome, as Pang [92] did for Planted Clique. That said, we believe that the behavior of SoS is qualitatively the same.

ACKNOWLEDGMENTS

CJ was supported in part by the ERC under the European Union’s Horizon 2020 research and innovation programme (grant agreement No. 834861); AP was supported in part by NSF grant CCF-2008920; GR was supported in part by NSF grants CCF-1816372 and CCF-2008920; JX was supported in part by NSF CAREER Award #2047933 and CyLab Presidential Fellowship. We thank Madhur Tulsiani for useful discussions. Most of the work for this project was completed while CJ and GR were PhD students at the University of Chicago.

REFERENCES

- [1] Kwangjun Ahn, Dhruv Medarametla, and Aaron Potechin. 2020. Graph Matrices: Norm Bounds and Applications. *arXiv preprint arXiv:1604.03423* (2020).
- [2] Vedat Levi Alev, Fernando Granha Jeronimo, and Madhur Tulsiani. 2019. Approximating Constraint Satisfaction Problems on High-Dimensional Expanders. (2019). Manuscript.
- [3] Noga Alon, Sanjeev Arora, Rajsekar Manokaran, Dana Moshkovitz, and Omri Weinstein. 2011. Inapproximability of densest k -subgraph from average case hardness, 2011. *Manuscript* 6 (2011).
- [4] Brendan PW Ames. 2015. Guaranteed recovery of planted cliques and dense subgraphs by convex relaxation. *Journal of Optimization Theory and Applications* 167, 2 (2015), 653–675.
- [5] Reid Andersen and Kumar Chellapilla. 2009. Finding dense subgraphs with size bounds. In *International workshop on algorithms and models for the web-graph*. Springer, 25–37.
- [6] Maria Chiara Angelini, Paolo Fachin, and Simone de Feo. 2021. Mismatching as a tool to enhance algorithmic performances of Monte Carlo methods for the planted clique model. *Journal of Statistical Mechanics: Theory and Experiment* 2021, 11 (2021), 113406.
- [7] Benny Applebaum, Boaz Barak, and Avi Wigderson. 2010. Public-key cryptography from different assumptions. In *Proceedings of the forty-second ACM symposium on Theory of computing*. 171–180.
- [8] Sanjeev Arora, Boaz Barak, Markus Brunnnermeier, Rong Ge, et al. 2010. Computational complexity and information asymmetry in financial products. In *ICS*.

- 49–65.
- [9] Sanjeev Arora, Satish Rao, and Umesh Vazirani. 2004. Expander flows and a $\sqrt{\log n}$ -approximation to sparsest cut.
 - [10] Sanjeev Arora, Satish Rao, and Umesh Vazirani. 2009. Expander flows, geometric embeddings and graph partitioning. *Journal of the ACM (JACM)* 56, 2 (2009), 1–37.
 - [11] Yuichi Asahiro, Refael Hassin, and Kazuo Iwama. 2002. Complexity of finding dense subgraphs. *Discrete Applied Mathematics* 121, 1–3 (2002), 15–26.
 - [12] Mitali Bafna, Boaz Barak, Pravesh K Kothari, Tselil Schramm, and David Steurer. 2021. Playing unique games on certified small-set expanders. In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*. 1629–1642.
 - [13] Mitali Bafna, Max Hopkins, Tali Kaufman, and Shachar Lovett. 2022. High Dimensional Expanders: Eigenstripping, Pseudorandomness, and Unique Games. In *Proceedings of the 2022 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*. SIAM, 1069–1128.
 - [14] Ainesh Bakshi, Ilias Diakonikolas, He Jia, Daniel M Kane, Pravesh K Kothari, and Santosh S Vempala. 2020. Robustly Learning Mixtures of k Arbitrary Gaussians. *arXiv preprint arXiv:2012.02119* (2020).
 - [15] Ainesh Bakshi and Pravesh K Kothari. 2020. List-Decodable Subspace Recovery: Dimension Independent Error in Polynomial Time. *arXiv preprint arXiv:2002.05139* (2020).
 - [16] Ainesh Bakshi and Adarsh Prasad. 2021. Robust linear regression: Optimal rates in polynomial time. In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*. 102–115.
 - [17] B. Barak, S. B. Hopkins, J. Kellner, P. Kothari, A. Moitra, and A. Potechin. 2016. A Nearly Tight Sum-of-Squares Lower Bound for the Planted Clique Problem. 428–437.
 - [18] Siddharth Barman. 2015. Approximating nash equilibria and dense bipartite subgraphs via an approximate version of caratheodory’s theorem. In *Proceedings of the forty-seventh annual ACM symposium on Theory of computing*. 361–369.
 - [19] Aditya Bhaskara, Moses Charikar, Eden Chlamtac, Uriel Feige, and Aravindan Vijayaraghavan. 2010. Detecting High Log-densities – An $O(n^{1/4})$ Approximation for Densest k -Subgraph.
 - [20] Aditya Bhaskara, Moses Charikar, Venkatesan Guruswami, Aravindan Vijayaraghavan, and Yuan Zhou. 2012. Polynomial integrality gaps for strong sdg relaxations of densest k -subgraph. In *Proceedings of the twenty-third annual ACM-SIAM symposium on Discrete Algorithms*. SIAM, 388–405.
 - [21] Enric Boix-Adserà, Matthew Brennan, and Guy Bresler. 2021. The Average-Case Complexity of Counting Cliques in Erdős–Rényi Hypergraphs. *SIAM J. Comput.* (2021), FOCs19–39.
 - [22] Polina Bombina and Brendan Ames. 2020. Convex optimization for the densest subgraph and densest submatrix problems. In *SN Operations Research Forum*, Vol. 1. Springer, 1–24.
 - [23] Mark Braverman, Young Kun Ko, Aviad Rubinfeld, and Omri Weinstein. 2017. ETH hardness for densest- k -subgraph with perfect completeness. In *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms*. SIAM, 1326–1341.
 - [24] Matthew Brennan and Guy Bresler. 2019. Optimal average-case reductions to sparse pca: From weak assumptions to strong hardness. *arXiv preprint arXiv:1902.07380* (2019).
 - [25] Matthew Brennan and Guy Bresler. 2020. Reducibility and statistical-computational gaps from secret leakage. In *Conference on Learning Theory*. PMLR, 648–847.
 - [26] Matthew Brennan, Guy Bresler, Samuel B Hopkins, Jerry Li, and Tselil Schramm. 2020. Statistical query algorithms and low-degree tests are almost equivalent. *arXiv preprint arXiv:2009.06107* (2020).
 - [27] Matthew Brennan, Guy Bresler, and Wasim Huleihel. 2018. Reducibility and computational lower bounds for problems with planted sparse structure. In *Conference on Learning Theory*. PMLR, 48–166.
 - [28] Matthew Brennan, Guy Bresler, and Wasim Huleihel. 2019. Universality of computational lower bounds for submatrix detection. In *Conference on Learning Theory*. PMLR, 417–468.
 - [29] Siu On Chan, James R Lee, Prasad Raghavendra, and David Steurer. 2016. Approximate constraint satisfaction requires large LP relaxations. *Journal of the ACM (JACM)* 63, 4 (2016), 1–22.
 - [30] Moses Charikar, Mohammad Taghi Hajiaghayi, and Howard Karloff. 2011. Improved approximation algorithms for label cover problems. *Algorithmica* 61, 1 (2011), 190–206.
 - [31] Moses Charikar, Konstantin Makarychev, and Yuri Makarychev. 2009. Integrality gaps for Sherali-Adams relaxations. In *Proceedings of the forty-first annual ACM symposium on Theory of computing*. 283–292.
 - [32] Chandra Chekuri and Shi Li. 2015. A note on the hardness of approximating the k -way Hypergraph Cut problem. *Manuscript*, <http://chekuri.cs.illinois.edu/papers/hypergraph-kcut.pdf> (2015).
 - [33] Wei Chen, Fu Li, Tian Lin, and Aviad Rubinfeld. 2015. Combining traditional marketing and viral marketing with amphibious influence maximization. In *Proceedings of the Sixteenth ACM Conference on Economics and Computation*. 779–796.
 - [34] Yudong Chen and Jiaming Xu. 2014. Statistical-computational tradeoffs in planted problems and submatrix localization with a growing number of clusters and submatrices. *arXiv preprint arXiv:1402.1267* (2014).
 - [35] Stephen R Chestnut and Rico Zenklusen. 2017. Hardness and approximation for network flow interdiction. *Networks* 69, 4 (2017), 378–387.
 - [36] Eden Chlamtác, Michael Dinitz, Christian Konrad, Guy Kortsarz, and George Rabanca. 2018. The densest k -subhypergraph problem. *SIAM Journal on Discrete Mathematics* 32, 2 (2018), 1458–1477.
 - [37] Eden Chlamtác, Michael Dinitz, and Robert Krauthgamer. 2012. Everywhere-sparse spanners via dense subgraphs. In *2012 IEEE 53rd Annual Symposium on Foundations of Computer Science*. IEEE, 758–767.
 - [38] Eden Chlamtác, Michael Dinitz, and Yuri Makarychev. 2017. Minimizing the union: Tight approximations for small set bipartite vertex expansion. In *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms*. SIAM, 881–899.
 - [39] Eden Chlamtác and Pasin Manurangsi. 2018. Sherali-adams integrality gaps matching the log-density threshold. *arXiv preprint arXiv:1804.07842* (2018).
 - [40] Eden Chlamtác, Pasin Manurangsi, Dana Moshkovitz, and Aravindan Vijayaraghavan. 2017. Approximation algorithms for label cover and the log-density threshold. In *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms*. SIAM, 900–919.
 - [41] Julia Chuzhoy, Yuri Makarychev, Aravindan Vijayaraghavan, and Yuan Zhou. 2015. Approximation algorithms and hardness of the k -route cut problem. *ACM Transactions on Algorithms (TALG)* 12, 1 (2015), 1–40.
 - [42] Nikhil R Devanur, Subhash A Khot, Rishi Saket, and Nisheeth K Vishnoi. 2006. Integrality gaps for sparsest cut and minimum linear arrangement problems. In *Proceedings of the thirty-eighth annual ACM symposium on Theory of computing*. 537–546.
 - [43] Uriel Feige. 2002. Relations between average case complexity and approximation complexity. In *Proceedings of the thirty-fourth annual ACM symposium on Theory of computing*. 534–543.
 - [44] Uriel Feige and Michael Langberg. 2001. Approximation algorithms for maximization problems arising in graph partitioning. *Journal of Algorithms* 41, 2 (2001), 174–211.
 - [45] Uriel Feige, David Peleg, and Guy Kortsarz. 2001. The dense k -subgraph problem. *Algorithmica* 29, 3 (2001), 410–421.
 - [46] Uriel Feige, Michael Seltser, et al. 1997. *On the densest k -subgraph problem*. Citeseer.
 - [47] Noah Fleming, Pravesh Kothari, Toniann Pitassi, et al. 2019. Semialgebraic proofs and efficient algorithm design. *Foundations and Trends® in Theoretical Computer Science* 14, 1–2 (2019), 1–221.
 - [48] Mrinalkanti Ghosh, Fernando Granha Jeronimo, Chris Jones, Aaron Potechin, and Goutham Rajendran. [2020] ©2020. Sum-of-squares lower bounds for Sherrington-Kirkpatrick via planted affine planes. In *2020 IEEE 61st Annual Symposium on Foundations of Computer Science—FOCS 2020*. IEEE Computer Soc., Los Alamitos, CA, 954–965.
 - [49] M.X. Goemans and D.P. Williamson. 1995. Improved Approximation Algorithms for Maximum Cut and Satisfiability Problems Using Semidefinite Programming. *J. ACM* 42, 6 (1995), 1115–1145. Preliminary version in *Proc. of STOC’94*.
 - [50] Doron Goldstein and Michael Langberg. 2009. The dense k subgraph problem. *arXiv preprint arXiv:0912.5327* (2009).
 - [51] Dima Grigoriev. 2001. Complexity of Positivstellensatz proofs for the knapsack. *computational complexity* 10, 2 (2001), 139–154.
 - [52] Bruce Hajek, Yihong Wu, and Jiaming Xu. 2015. Computational lower bounds for community detection on random graphs. In *Conference on Learning Theory*. PMLR, 899–928.
 - [53] Bruce Hajek, Yihong Wu, and Jiaming Xu. 2016. Achieving exact cluster recovery threshold via semidefinite programming. *IEEE Transactions on Information Theory* 62, 5 (2016), 2788–2797.
 - [54] Mohammad Taghi Hajiaghayi and Kamal Jain. 2006. The prize-collecting generalized Steiner tree problem via a new approach of primal-dual schema. In *SODA*, Vol. 6. 631–640.
 - [55] Mohammad Taghi Hajiaghayi, Kamal Jain, Lap Chi Lau, Il Mandoiu, Alexander Russell, and Vijay V Vazirani. 2006. Minimum multicolored subgraph problem in multiplex PCR primer set selection and population haplotyping. In *International Conference on Computational Science*. Springer, 758–766.
 - [56] Koki Hamada, Kazuo Iwama, and Shuichi Miyazaki. 2011. The hospital/residents problem with quota lower bounds. In *European Symposium on Algorithms*. Springer, 180–191.
 - [57] Shuichi Hirahara and Nobutaka Shimizu. 2021. Nearly optimal average-case complexity of counting bicliques under SETH. In *Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms (SODA)*. SIAM, 2346–2365.
 - [58] Samuel B Hopkins. 2020. Mean estimation with sub-Gaussian rates in polynomial time. *The Annals of Statistics* 48, 2 (2020), 1193–1213.
 - [59] Samuel B Hopkins, Pravesh K Kothari, and Aaron Potechin. 2015. Sos and planted clique: Tight analysis of MPW moments at all degrees and an optimal lower bound at degree four. *arXiv preprint arXiv:1507.05230* (2015).

- [60] Samuel B Hopkins, Pravesh K Kothari, Aaron Potechin, Prasad Raghavendra, Tselil Schramm, and David Steurer. 2017. The power of sum-of-squares for detecting hidden structures. *IEEE*, 720–731.
- [61] Samuel B Hopkins and Jerry Li. 2018. Mixture models, robustness, and sum of squares proofs. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*. 1021–1034.
- [62] Samuel Brink Klevit Hopkins. 2018. *Statistical Inference and the Sum of Squares Method*. Ph.D. Dissertation. Cornell University.
- [63] Chris Jones. 2022. *Symmetrized Fourier Analysis of Convex Relaxations for Combinatorial Optimization Problems*. Ph.D. Dissertation. The University of Chicago.
- [64] Chris Jones, Aaron Potechin, Goutham Rajendran, Madhur Tulsiani, and Jeff Xu. 2022. Sum-of-squares lower bounds for sparse independent set. In *2021 IEEE 62nd Annual Symposium on Foundations of Computer Science (FOCS)*. IEEE, 406–416.
- [65] Richard M Karp. 1972. Reducibility among combinatorial problems. In *Complexity of computer computations*. Springer, 85–103.
- [66] Yash Khanna and Anand Louis. 2020. Planted Models for the Densest k -Subgraph Problem. *arXiv preprint arXiv:2004.13978* (2020).
- [67] Subhash Khot. 2006. Ruling out PTAS for graph min-bisection, dense k -subgraph, and bipartite clique. *SIAM J. Comput.* 36, 4 (2006), 1025–1071.
- [68] Subhash A Khot and Nisheeth K Vishnoi. 2015. The unique games conjecture, integrality gap for cut problems and embeddability of negative-type metrics into ℓ_1 . *Journal of the ACM (JACM)* 62, 1 (2015), 1–39.
- [69] Adam Klivans, Pravesh K Kothari, and Raghu Meka. 2018. Efficient algorithms for outlier-robust regression. In *Conference On Learning Theory*. PMLR, 1420–1430.
- [70] Stavros G Kolliopoulos and George Steiner. 2007. Partially ordered knapsack and applications to scheduling. *Discrete Applied Mathematics* 155, 8 (2007), 889–897.
- [71] Guy Kortsarz, Vahab S Mirrokni, Zeev Nutov, and Elena Tsanko. 2008. Approximating minimum-power degree and connectivity problems. In *Latin American Symposium on Theoretical Informatics*. Springer, 423–435.
- [72] Guy Kortsarz and David Peleg. 1993. *On choosing a dense subgraph*. IEEE.
- [73] Pravesh Kothari, Ryuhei Mori, Ryan O'Donnell, and David Witmer. 2017. Sum of squares lower bounds for refuting any CSP.
- [74] Pravesh K Kothari, Raghu Meka, and Prasad Raghavendra. 2017. Approximating rectangles by juntas and weakly-exponential lower bounds for LP relaxations of CSPs. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*. 590–603.
- [75] Pravesh K Kothari and David Steurer. 2017. Outlier-robust moment-estimation via sum-of-squares. *arXiv preprint arXiv:1711.11581* (2017).
- [76] Dmitriy Kunisky. 2020. Positivity-preserving extensions of sum-of-squares pseudomoments over the hypercube. *arXiv preprint arXiv:2009.07269* (2020).
- [77] Dmitriy Kunisky. 2021. *Spectral Barriers in Certification Problems*. Ph.D. Dissertation. New York University.
- [78] Dmitriy Kunisky, Alexander S Wein, and Afonso S Bandeira. 2022. Notes on computational hardness of hypothesis testing: Predictions using the low-degree likelihood ratio. In *ISAAC Congress (International Society for Analysis, its Applications and Computation)*. Springer, 1–50.
- [79] Jean B Lasserre. 2001. Global optimization with polynomials and the problem of moments. *SIAM Journal on optimization* 11, 3 (2001), 796–817.
- [80] Euiwoong Lee. 2017. Partitioning a graph into small pieces with applications to path transversal. In *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms*. SIAM, 1546–1558.
- [81] James R Lee, Prasad Raghavendra, and David Steurer. 2015. Lower bounds on the size of semidefinite programming relaxations. In *Proceedings of the forty-seventh annual ACM symposium on Theory of computing*. 567–576.
- [82] Zhentao Li, Manikandan Narayanan, and Adrian Vetta. 2014. The Complexity of the Simultaneous Cluster Problem. *J. Graph Algorithms Appl.* 18, 1 (2014), 1–34.
- [83] Allen Liu and Ankur Moitra. 2021. Settling the robust learnability of mixtures of gaussians. In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*. 518–531.
- [84] Pasin Manurangsi. 2017. Almost-polynomial ratio ETH-hardness of approximating densest k -subgraph. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*. 954–961.
- [85] Pasin Manurangsi and Dana Moshkovitz. 2015. Approximating dense max 2-CSPs. *arXiv preprint arXiv:1507.08348* (2015).
- [86] Pasin Manurangsi, Aviad Rubinfeld, and Tselil Schramm. 2020. The strongish planted clique hypothesis and its consequences. *arXiv preprint arXiv:2011.05555* (2020).
- [87] Cheng Mao, Alexander S. Wein, and Shenduo Zhang. 2023. Detection-Recovery Gap for Planted Dense Cycles. *arXiv:2302.06737* [math.ST]
- [88] Raghu Meka, Aaron Potechin, and Avi Wigderson. 2015. Sum-of-squares lower bounds for planted clique. In *Proceedings of the forty-seventh annual ACM symposium on Theory of computing*. 87–96.
- [89] Sidhanth Mohanty, Prasad Raghavendra, and Jeff Xu. 2020. Lifting sum-of-squares lower bounds: degree-2 to degree-4. 840–853.
- [90] Yurii Nesterov. 2000. Squared functional systems and optimization problems. In *High performance optimization*. Springer, 405–440.
- [91] Ryan O'Donnell. 2017. SOS is not obviously automatizable, even approximately. In *8th Innovations in Theoretical Computer Science Conference (ITCS 2017)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik.
- [92] Shuo Pang. 2021. SOS lower bound for exact planted clique. In *36th Computational Complexity Conference*. LIPIcs. Leibniz Int. Proc. Inform., Vol. 200. Schloss Dagstuhl. Leibniz-Zent. Inform., Wadern, Art. 26, 63.
- [93] Pablo A Parrilo. 2000. *Structured semidefinite programs and semialgebraic geometry methods in robustness and optimization*. Ph.D. Dissertation. California Institute of Technology.
- [94] David Pisinger. 2007. The quadratic knapsack problem—a survey. *Discrete applied mathematics* 155, 5 (2007), 623–648.
- [95] Aaron Potechin and Goutham Rajendran. 2020. Machinery for Proving Sum-of-Squares Lower Bounds on Certification Problems. *arXiv preprint arXiv:2011.04253* (2020).
- [96] Aaron Potechin and Goutham Rajendran. 2022. Sub-exponential time Sum-of-Squares lower bounds for Principal Components Analysis. *Advances in Neural Information Processing Systems* (2022).
- [97] Prasad Raghavendra. 2008. Optimal algorithms and inapproximability results for every CSP? 245–254.
- [98] Prasad Raghavendra, Tselil Schramm, and David Steurer. 2018. High dimensional estimation via sum-of-squares proofs. In *Proceedings of the International Congress of Mathematicians: Rio de Janeiro 2018*. World Scientific, 3389–3423.
- [99] Prasad Raghavendra and David Steurer. 2010. Graph expansion and the unique games conjecture. In *Proceedings of the forty-second ACM symposium on Theory of computing*. 755–764.
- [100] Prasad Raghavendra and Benjamin Weitz. 2017. On the Bit Complexity of Sum-of-Squares Proofs. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik.
- [101] Goutham Rajendran. 2018. Combinatorial Optimization via the Sum of Squares Hierarchy. *arXiv preprint arXiv:2208.04374* (2018).
- [102] Goutham Rajendran. 2022. *Nonlinear Random Matrices and Applications to the Sum of Squares Hierarchy*. Ph.D. Dissertation. The University of Chicago.
- [103] Goutham Rajendran and Madhur Tulsiani. 2023. Concentration of polynomial random matrices via Efron-Stein inequalities. In *Proceedings of the 2023 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*. SIAM, 3614–3653.
- [104] Tselil Schramm. 2017. *Random Matrices and the Sum-of-Squares Hierarchy*. Ph.D. Dissertation. UC Berkeley.
- [105] Tselil Schramm and Alexander S Wein. 2022. Computational barriers to estimation from low-degree polynomials. *The Annals of Statistics* 50, 3 (2022), 1833–1858.
- [106] Naum Zuselevich Shor. 1987. An approach to obtaining global extremums in polynomial mathematical programming problems. *Cybernetics* 23, 5 (1987), 695–700.
- [107] Piotr Skowron, Piotr Faliszewski, and Jérôme Lang. 2016. Finding a collective set of items: From proportional multirepresentation to group recommendation. *Artificial Intelligence* 241 (2016), 191–216.
- [108] Anand Srivastav and Katja Wolf. 1998. Finding dense subgraphs with semidefinite programming. In *International Workshop on Approximation Algorithms for Combinatorial Optimization*. Springer, 181–191.
- [109] Akiko Suzuki and Takeshi Tokuyama. 2008. Dense subgraph problems with output-density conditions. *ACM Transactions on Algorithms (TALG)* 4, 4 (2008), 1–18.
- [110] Sumedh Tirodkar and Sundar Vishwanathan. 2017. On the approximability of the minimum rainbow subgraph problem and other related problems. *Algorithmica* 79, 3 (2017), 909–924.
- [111] Madhur Tulsiani. 2009. CSP Gaps and Reductions in the Lasserre Hierarchy.
- [112] Ilias Zadik, Min Jae Song, Alexander S Wein, and Joan Bruna. 2022. Lattice-based methods surpass sum-of-squares in clustering. In *Conference on Learning Theory*. PMLR, 1247–1248.

Received 2022-11-07; accepted 2023-02-06