

# Ellipsoid Fitting up to a Constant

Jun-Ting Hsieh ✉ 🏠

Carnegie Mellon University, Pittsburgh, PA, USA

Pravesh K. Kothari ✉ 🏠

Carnegie Mellon University, Pittsburgh, PA, USA

Aaron Potechin ✉ 🏠

University of Chicago, IL, USA

Jeff Xu ✉ 🏠

Carnegie Mellon University, Pittsburgh, PA, USA

---

## Abstract

In [11, 13], Saunderson, Parrilo, and Willsky asked the following elegant geometric question: what is the largest  $m = m(d)$  such that there is an ellipsoid in  $\mathbb{R}^d$  that passes through  $v_1, v_2, \dots, v_m$  with high probability when the  $v_i$ s are chosen independently from the standard Gaussian distribution  $N(0, I_d)$ ? The existence of such an ellipsoid is equivalent to the existence of a positive semidefinite matrix  $X$  such that  $v_i^\top X v_i = 1$  for every  $1 \leq i \leq m$  – a natural example of a *random* semidefinite program. SPW conjectured that  $m = (1 - o(1))d^2/4$  with high probability. Very recently, Potechin, Turner, Venkat and Wein [10] and Kane and Diakonikolas [8] proved that  $m \gtrsim d^2/\log^{O(1)}(d)$  via a certain natural, explicit construction.

In this work, we give a substantially tighter analysis of their construction to prove that  $m \gtrsim d^2/C$  for an absolute constant  $C > 0$ . This resolves one direction of the SPW conjecture up to a constant. Our analysis proceeds via the method of *Graphical Matrix Decomposition* that has recently been used to analyze correlated random matrices arising in various areas [3, 2]. Our key new technical tool is a refined method to prove singular value upper bounds on certain correlated random matrices that are tight up to absolute dimension-independent constants. In contrast, all previous methods that analyze such matrices lose logarithmic factors in the dimension.

**2012 ACM Subject Classification** Theory of computation → Semidefinite programming

**Keywords and phrases** Semidefinite programming, random matrices, average-case complexity

**Digital Object Identifier** 10.4230/LIPIcs.ICALP.2023.78

**Category** Track A: Algorithms, Complexity and Games

**Funding** *Jun-Ting Hsieh*: Supported by NSF CAREER Award #2047933.

*Pravesh K. Kothari*: Supported by NSF CAREER Award #2047933, Alfred P. Sloan Fellowship and a Google Research Scholar Award.

*Aaron Potechin*: Supported in part by NSF grant CCF-2008920.

*Jeff Xu*: Supported in part by NSF CAREER Award #2047933, and a Cylab Presidential Fellowship.

## 1 Introduction

Given vectors  $v_1, \dots, v_m \in \mathbb{R}^d$ , we say that these vectors satisfy the *ellipsoid fitting property* if there exists an origin-centered ellipsoid that passes through all these points, i.e., if there exists a matrix  $\Lambda$  such that

1.  $v_i^\top \Lambda v_i = 1$  for all  $i \in [m]$ ,
2.  $\Lambda \succeq 0$ .

In this work, we study vectors sampled i.i.d. from the standard Gaussian distribution. It is known that when  $m \leq d+1$ , the vectors satisfy the ellipsoid fitting property with probability 1 [12]. On the other hand, when  $m > \binom{d+1}{2}$ , by a simple dimension argument, the vectors



© Jun-Ting Hsieh, Pravesh K. Kothari, Aaron Potechin, and Jeff Xu;  
licensed under Creative Commons License CC-BY 4.0

50th International Colloquium on Automata, Languages, and Programming (ICALP 2023).

Editors: Kousha Etessami, Uriel Feige, and Gabriele Puppis; Article No. 78; pp. 78:1–78:20

Leibniz International Proceedings in Informatics



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



don't satisfy the ellipsoid fitting property with probability 1. This prompts the question: *what is the largest  $m = m(d)$  such that  $v_1, \dots, v_m \sim \mathcal{N}(0, I_d)$  satisfy the ellipsoid fitting property with probability at least  $1 - o_d(1)$  (taking  $d \rightarrow \infty$ )?*

In a series of work, Saunderson et. al. [11, 12, 13] studied this problem in the context of diagonal and low-rank matrix decomposition. Motivated by numerical experiments, they conjectured that the ellipsoid fitting property for Gaussian random  $v_i$ s exhibits a phase transition at  $m \sim \frac{d^2}{4}$  (see also the experiments presented in [10]).

► **Conjecture 1** (SCPW conjecture). *Let  $\varepsilon > 0$  be a constant and  $v_1, \dots, v_m \sim \mathcal{N}(0, I_d)$  be i.i.d. standard Gaussian vectors in  $\mathbb{R}^d$ . Then,*

1. *If  $m \leq (1 - \varepsilon)\frac{d^2}{4}$ , then  $v_1, \dots, v_m$  have the ellipsoid fitting property with probability  $1 - o_d(1)$ .*
2. *If  $m \geq (1 + \varepsilon)\frac{d^2}{4}$ , then  $v_1, \dots, v_m$  have the ellipsoid fitting property with probability  $o_d(1)$ .*

Prior works have focused on establishing the positive result – that is, part (1) of the above conjecture. Early works [11, 13] established that the ellipsoid fitting property holds for  $m \leq O(d^{6/5-\varepsilon})$  independent Gaussian vector whp. In the context of proving Sum-of-Squares lower bounds for the Sherrington-Kirkpatrick model, the work [4] obtains a result that, as an immediate corollary, improves the above bound to  $O(d^{3/2-\varepsilon})$ . In fact, their work gives an implicit bound of  $m \leq O(d^2 / \text{polylog}(d))$  for ellipsoid fitting when restricted to degree-2 Sum-of-Squares.

Very recently, two independent works of Potechin et. al. [10] and Kane and Diakonikolas [8] proposed new constructions of  $\Lambda$  (that differ from the constructions obtained by the method of pseudo-calibration in [4]) and recovered the bound of  $m \leq O(d^2 / \text{polylog}(d))$ . In their works [10, 8], the authors ask the question of analyzing their construction (or a different one) to obtain an improved and almost optimal estimate of  $m = d^2/C$  for some absolute constant  $C > 0$ . The main result of this paper accomplishes this goal. Specifically, we prove:

► **Theorem 2** (Main result). *There is a universal constant  $c > 0$  such that if  $m \leq cd^2$ , then  $v_1, \dots, v_m \sim \mathcal{N}(0, I_d)$  have the ellipsoid fitting property with probability  $1 - o_d(1)$ .*

We establish Theorem 2 by analyzing the construction of Kane and Diakonikolas [8] (which is a variant of the construction proposed in [10]). Our key idea is to depart from the analysis conducted by [8] and instead rely on the *graphical matrix decomposition* method. This method decomposes a random matrix with correlated entries into a sum of structured random matrices called *graph matrices*. Graph matrices can be thought of as an analog of the *Fourier basis* in the analysis of functions over product spaces. This method was first employed in the works establishing tight sum-of-squares lower bound on the planted clique problem [5, 1, 3, 7] and has since then been employed in several follow-up works on proving sum-of-squares lower bounds and more recently in analyzing well-conditionedness of linear algebraic algorithms for generalizations of tensor decomposition [2]).

The key technical work in the analysis then becomes understanding the smallest and largest singular values of graph matrices. All prior works rely on arguments that establish bounds on the largest singular values that are accurate up to polylogarithmic factors in the underlying dimension of the matrices. The work of [2] recently showed how to use such bounds to also obtain estimates of the smallest singular values of graph matrices (which, otherwise are significantly more challenging to prove). Nevertheless, the slack in such bounds does not allow us to obtain any improvement on the previous estimates [8] in our application.

Our main technical contribution is a new technique to establish bounds on the largest singular values of graph matrices that are tight up to dimension-independent absolute constants. This allows us to obtain substantially improved estimates for the SCPW conjecture. Given the host of previous applications of such bounds, we expect that our results will have many more applications down the line.

■ **Table 1** Comparison of our result with prior work.

Construction	Bound on $m$
Conjectured	$d^2/4$
[11, 13]	$O(d^{6/5-\varepsilon})$
[4]	$O(d^{3/2-\varepsilon})$ *
[10]	$O(d^2/\text{polylog}(d))$
[8]	$O(d^2/\log^4(d))$
<b>this paper</b>	$O(d^2)$

\*The bound  $O(d^2/\text{polylog}(d))$  is implicit in their work.

## 1.1 Technical overview

Following the convention of [8], for the rest of the paper we will assume that  $v_1, \dots, v_m \sim \mathcal{N}(0, \frac{1}{d}I_d)$  such that each vector has expected norm 1. Note that this does not change the problem as we can simply scale  $\Lambda$ .

Our construction of  $\Lambda$  is the “identity perturbation construction”, which is the same one analyzed in [8] and was proposed in [10]. As an intuition, observe that  $\Lambda = I_d$  almost works:  $v_i^T I_d v_i = \|v_i\|_2^2 \approx 1$ . Thus, the idea is to define  $\Lambda$  as a perturbation of  $I_d$ :  $\Lambda = I_d - \sum_{i=1}^m w_i v_i v_i^T$ , where  $w = (w_1, \dots, w_m) \in \mathbb{R}^m$ . To determine  $w$ , observe that the constraints  $v_i^T \Lambda v_i = 1$  give  $m$  linear constraints on  $w$ , and this can be written as a linear system represented by a matrix  $M \in \mathbb{R}^{m \times m}$  with entries  $M[i, j] = \langle v_i, v_j \rangle^2$ . Thus, given that  $M$  is full rank,  $w$  is uniquely determined by  $w = M^{-1}\eta$  for some vector  $\eta$  (see Eq. (2)). This construction satisfies  $v_i^T \Lambda v_i = 1$  automatically, so the next thing is to prove that  $\Lambda \succeq 0$ . Therefore, we have two high-level goals:

1. Prove that  $M$  is full rank and analyze  $M^{-1}$ .
  2. Prove that  $R := \sum_{i=1}^m w_i v_i v_i^T$  has spectral norm bounded by 1.
- Proving the second statement immediately implies that  $\Lambda$  is a valid construction.

To achieve the first goal, we *decompose*  $M$  into several components. Roughly, we write  $M = A + B$  where  $A$  is a perturbed identity matrix  $A = I_m - T$  and  $B$  is a rank-2 matrix (see Section 2.2). We first show that  $\|T\|_{\text{op}} \leq O(\frac{\sqrt{m}}{d}) < 0.5$  with  $m \leq O(d^2)$  (Lemma 9), hence  $A$  is well-conditioned. Then, using the fact that  $B$  has rank 2, we can apply the Woodbury matrix identity (Fact 7 and Fact 8) – a statement on the inverse of low-rank corrections of matrices – to conclude that  $M$  is invertible and obtain an expression for  $M^{-1}$ . This is carried out in Section 2.3.

Next, for the second goal, we need to further expand  $A^{-1}$ . Since  $\|T\|_{\text{op}} < 1$ , we can apply the Neumann series and write  $A^{-1} = (I_m - T)^{-1} = \sum_{k=0}^{\infty} T^k$ . For the analysis, we select certain thresholds to truncate this series such that the truncation error is small. Then, we write  $M^{-1}$  in terms of the truncated series plus a small error, which will be useful later for the analysis of  $R$ . This is carried out in the full version.

Finally, given the expression of  $M^{-1}$ ,  $R$  naturally decomposes into 4 matrices. Then, all we need to do is to bound the spectral norm of each of these matrices (see the full version). Bounding  $\|R\|_{\text{op}} \leq 1$  implies that  $\Lambda \succeq 0$ , completing the proof.

**Requiring tight norm bounds.** Our main technical lemmas are the spectral norm bounds of  $T$  (Lemma 9) and the matrices in the decomposition of  $R$ . Clearly, we need our norm bound  $\|T\|_{\text{op}} \leq O(\frac{\sqrt{m}}{d})$  to be tight without polylog factors so that  $m \leq O(d^2)$  suffices, and similarly for matrices from  $R$ .

The standard starting point is the *trace moment method*: for any symmetric matrix  $M \in \mathbb{R}^{n \times n}$  and  $q \in \mathbb{N}$  (usually taking  $q = \text{polylog}(n)$  suffices),

$$\|M\|_{\text{op}}^{2q} \leq \text{tr}(M^{2q}) = \sum_{i_1, i_2, \dots, i_{2q} \in [n]} M[i_1, i_2] M[i_2, i_3] \cdots M[i_{2q}, i_1].$$

We view the summand as a closed walk  $i_1 \rightarrow i_2 \rightarrow \cdots \rightarrow i_{2q} \rightarrow i_1$  on  $n$  vertices. For a random matrix, we study the expected trace  $\mathbb{E} \text{tr}(M^{2q})$ . In the simple case when  $M$  is a Gaussian matrix (GOE), we see that after taking the expectation, the non-vanishing terms are closed walks where each edge  $(u, v)$  is traversed even number of times. This is in fact true for any symmetric  $M$  with independent random entries as long as the odd moments of the entries are zero. Thus, a precise upper bound on  $\mathbb{E} \text{tr}(M^{2q})$  can be obtained by carefully counting such closed walks (see [14]).

Our matrices are more complicated; each entry is a mean-zero *polynomial* of Gaussian random variables. To carry out the trace method, we represent the matrices as graphs, hence the term *graph matrices*. The framework of graph matrices was first introduced by [3], and over the years, off-the-shelf norm bounds (e.g. [1]) for graph matrices have been developed and successfully used in several works [9, 4, 6, 7, 2]. However, the currently known norm bounds are only tight up to polylog factors, hence not sufficient for us. Therefore, the bulk of our paper is to prove norm bounds for these matrices that are tight up to constant factors. In fact, some of our bounds on graph matrices are even tight in the constant factor. However, we do not pursue the exact constants for two reasons. First, obtaining bounds which are tight in the constant factor would require additional technical work. Second, numerical experiments from [10] show that the identity perturbation construction we analyze has a threshold of  $\frac{d^2}{C_{IP}}$  where  $C_{IP} \approx 10$ , so it falls short of the  $\frac{d^2}{4}$  threshold and we would need a different construction to reach this threshold.

**Key idea towards tight norm bounds.** Here, we briefly discuss the high-level ideas for proving tight norm bounds. To illustrate our techniques, in Section 3 we will give a full proof for a matrix that arises in our analysis as an example, and also discuss key ideas that allow us to analyze more complicated matrices.

The key to counting walks is to specify an *encoding*, which we view as *information* required for a *walker* to complete a walk. If we can show that such an encoding uniquely identifies a walk, then we can bound the walks by bounding the number of possible encodings. Thus, it suffices to come up with an (efficient) encoding scheme and prove that the walker is able to complete a walk. Using standard encoding schemes, we quickly realize that the walker may be *confused* during the walk, i.e., the walker does not have enough information to perform the next step. Thus, we need to *pay* for additional information in the encoding to resolve confusions. So far, this is the same high-level strategy that was used in prior work [14, 1, 7], and this extra pay is often the source of extra log factors in the norm bounds.

Our key innovation is to pay for the extra information during steps that require much less information than normal. Roughly speaking, we label each step of the walk as either (1) visiting a new vertex, (2) visiting an old vertex via a new edge, (3) using an old edge but not the last time, (4) using an old edge the last time (see Definition 20). The high level idea is that the dominating walks in the trace are the ones that use only the 1st and 4th

types, while the 2nd and 3rd types require less information (which we call *gaps*). The main observation is that the walker will be confused only when there are steps of the 2nd and 3rd type involved, but we can pay extra information during these steps to resolve potential (future) confusions. This is illustrated in Section 3.5.

## 1.2 Comparison to prior work

**Comparison to Kane and Diakonikolas [8].** Our candidate matrix  $\Lambda$  is the same as theirs. A slight difference is that they write  $\Lambda = I_d + \sum_{i=1}^m \delta_i \bar{v}_i \bar{v}_i^T$  where  $\bar{v}_1, \dots, \bar{v}_m$  are the vectors normalized to the unit sphere. Then, same as our  $w$  vector,  $(\delta_1, \dots, \delta_m)$  must satisfy a linear system represented by a matrix  $\bar{M} \in \mathbb{R}^{m \times m}$  where  $\bar{M}[i, j] = \langle \bar{v}_i, \bar{v}_j \rangle^2$ . This is closely related to our  $M$  matrix, and to prove that  $\bar{M}$  is invertible, they also decompose  $\bar{M}$  into several components and bound their spectral norms. However, they were only able to bound the spectral norm by  $O(\frac{\sqrt{m \log^2 d}}{d})$ , which requires  $m \leq O(d^2 / \log^4(d))$ . We also point out that they explicitly emphasize the gap from spectral norm bound poses a significant hurdle in their analysis, which is indeed a major contribution of our work.

Next, to bound the spectral norm of  $\bar{R} := \sum_{i=1}^m \delta_i \bar{v}_i \bar{v}_i^T$ , they use an elegant cover (or  $\varepsilon$ -net) argument which is significantly different than ours. They show that for any fixed unit vector  $u \in \mathcal{S}^{d-1}$ ,  $|u^T \bar{R} u| = |\sum_{i=1}^m \delta_i \langle \bar{v}_i, u \rangle^2| \leq 1/2$  with *exponentially* small failure probability. This allows then to take a union bound over all  $2^{O(d)}$  unit vectors in an  $\varepsilon$ -net. To do this, they use the elegant trick that  $\bar{v}_i$  and  $\|v_i\|_2$  are independent random variables, so  $u^T \bar{R} u$  can be written as a sum of independent variables:  $u^T \bar{R} u = \langle \varepsilon, \gamma \rangle$ , where  $\varepsilon_i$  only depends on  $\|v_i\|_2$  and  $\gamma$  is a function of  $u$  and the  $\bar{v}_i$ 's. By Hoeffding's inequality, they get a tail probability of  $\exp\left(-\Omega\left(\frac{d^3}{m \log^2(d)}\right)\right)$ . In order to union bound over  $2^{O(d)}$  vectors, this also requires that  $m \leq O(d^2 / \log^2(d))$ . Thus, while the main source of their polylog gap is their matrix norm bound, another source is the epsilon-net argument. This is partially why we adopt the proof strategy of using graph matrix decompositions which is seemingly more complicated.

**Comparison to Potechin, Turner, Venkat and Wein [10].** They study a construction of “least-square minimization” proposed by [11], which is equivalent to projecting out the identity mass onto the subspace of matrices satisfying the constraints. In particular, their matrix analysis proceeding via Woodbury expansion and Neumann series using graph matrices serves as a road-map for our current work, and gives rise to a motivating question in the beginning for our work: can a more careful analysis get us all the way to a constant factor gap, or is the polylog gap inherent in the analysis? A priori, it is not clear whether this kind of matrix analysis, forsaking the underlying geometric insight, might get us anywhere beyond a single polylog factor, as it is conceivable that some polylog factor is inherent for matrices that may arise in the analysis. In this work, we answer this question affirmatively and en-route we develop a more refined understanding of the structured random matrices that we believe would be useful in further and more fine-grained investigations of problems in average-case complexity.

**Comparison to Ghosh et. al. [4].** In the context of the Planted Affine Plane problem, and its downstream application for the Sherrington-Kirkpatrick Hamiltonian, Ghosh et. al. reaches the threshold of  $\tilde{O}(d^{3/2-\varepsilon})$  for  $n^{O(\varepsilon)}$ -degree Sum-of-Squares. They adopt the framework of pseudo-calibration [3] to obtain a candidate matrix, and follow a similar recipe as ours via graph matrix decompositions and spectral analysis. Even though their stated result falls

short of the  $\tilde{O}(d^2)$  threshold for fitting ellipsoid, it is folklore among the SoS lower bounds community that their proof implicitly extends to  $\tilde{O}(d^2)$  when restricted to degree-2 SoS. That said, it is an interesting question whether solutions coming from a pseudo-calibration type of construction might give us some extra mileage in ultimately closing the constant gap. A natural idea is to analyze the planted distribution pioneered in [9, 4]: unfortunately, it can be easily verified that the low-degree polynomial hardness for the particular planted distribution actually falls apart even if we assume an arbitrary constant gap. Since the low-degree hardness is usually deemed as a precursor for SoS lower bounds, an analysis based on pseudo-calibration that gets us the right constant (or in fact, any constant) lands one on a pursuit for a "quieter" planting.

## 2 Proof of main result

Given  $v_1, v_2, \dots, v_m$  that are i.i.d. samples from  $\mathcal{N}(0, \frac{1}{d}I_d)$ , recall that we must construct a matrix  $\Lambda$  such that (1)  $v_i^T \Lambda v_i = 1$  for any  $i \in [m]$ , and (2)  $\Lambda \succeq 0$ .

In this section, we describe our candidate matrix (Definition 3). To prove that it satisfies the two conditions above, we need to analyze certain random matrices (and their inverses) that arise in the construction, which involves decomposing the matrices into simpler components. We will state our key spectral norm bounds (Lemma 9 and Lemma 13) whose proofs are deferred to later sections, and complete the proof of Theorem 2 in Section 2.4.

### 2.1 Candidate construction

The following is our candidate matrix  $\Lambda$ , which is the same as the one used in [8].

► **Definition 3** (Candidate matrix). *Given  $v_1, \dots, v_m \sim \mathcal{N}(0, \frac{1}{d}I_d)$ , we define the matrix  $\Lambda \in \mathbb{R}^{d \times d}$  to be*

$$\Lambda := I_d - \sum_{i=1}^m w_i v_i v_i^T \quad (1)$$

where we take  $w = (w_1, w_2, \dots, w_m)$  to be the solution to the linear system  $Mw = \eta$  for  $\eta \in \mathbb{R}^m$  given by

$$\eta_i := \|v_i\|_2^2 - 1, \quad \forall i \in [m], \quad (2)$$

and  $M \in \mathbb{R}^{m \times m}$  with entries given by

$$M[i, j] := \langle v_i, v_j \rangle^2, \quad \forall i, j \in [m]. \quad (3)$$

We first make the following simple observation.

► **Observation 4.** *For any  $i \in [m]$ , the constraint  $v_i^T \Lambda v_i = 1$  is satisfied.*

**Proof.** For any  $i \in [m]$ ,

$$v_i^T \Lambda v_i = v_i^T I_d v_i - \sum_{j \in [m]} w_j \langle v_i, v_j \rangle^2 = \|v_i\|_2^2 - \langle M[i], w \rangle = \|v_i\|_2^2 - \eta_i = 1.$$

Here  $M[i]$  is the  $i$ -th row of  $M$ , and the equality above follows from  $Mw = \eta$  and  $\eta_i = \|v_i\|_2^2 - 1$  from Eq. (2). ◀

**Structure of subsequent sections.** For  $\Lambda$  to be well-defined, we require that  $M$  is full rank (hence invertible). Note that it is easy to see that  $M$  is positive semidefinite, since  $M$  is a Gram matrix with  $M[i, j] = \langle v_i^{\otimes 2}, v_j^{\otimes 2} \rangle$ . To analyze  $M$ , we will show a decomposition of  $M$  in Section 2.2 that allows us to more easily analyze its inverse. In Section 2.3, we will prove that  $M$  is in fact positive definite (Lemma 12).

Next, to prove that  $\Lambda \succeq 0$ , we will write  $\Lambda = I_d - R$  where

$$R := \sum_{i=1}^m w_i v_i v_i^T = \sum_{i=1}^m (M^{-1} \eta) [i] \cdot v_i v_i^T, \quad (4)$$

and prove that  $\|R\|_{\text{op}}$  is bounded by 1. Finally, combining the analyses, we finish the proof of Theorem 2 in Section 2.4.

## 2.2 Decomposition of $M$

The proof of Theorem 2 requires careful analysis of the matrix  $M$  from Eq. (3) and its inverse. To this end, we first decompose  $M$  as  $M = A + B$  such that intuitively,  $A$  is perturbation of a (scaled) identity matrix and  $B$  has rank 2. We will later see how this decomposition allows us to analyze  $M^{-1}$  more conveniently.

► **Proposition 5** (Decomposition of  $M$ ).

$$M = \underbrace{M_\alpha + M_\beta + M_D}_{:=A} + \underbrace{\left(1 + \frac{1}{d}\right) I_m + \frac{1}{d} J_m + \frac{1}{d} (1_m \cdot \eta^T + \eta \cdot 1_m^T)}_{:=B} \quad (5)$$

where  $J_m$  is the all-ones matrix,  $M_\alpha, M_\beta$  are matrices with zeros on the diagonal and  $M_D$  is a diagonal matrix, defined as follows:

- $M_\alpha[i, j] := \sum_{a \neq b \in [d]} v_i[a] \cdot v_i[b] \cdot v_j[a] \cdot v_j[b]$  for  $i \neq j \in [m]$ ,
- $M_\beta[i, j] := \sum_{a \in [d]} (v_i[a]^2 - \frac{1}{d}) (v_j[a]^2 - \frac{1}{d})$  for  $i \neq j \in [m]$ ,
- $M_D[i, i] := \|v_i\|_2^4 - \frac{2}{d} \|v_i\|_2^2 - 1$  for  $i \in [m]$ .

**Proof.** For any off-diagonal entry  $i \neq j \in [m]$ , on the right-hand side we have

$$\begin{aligned} M[i, j] &= \langle v_i, v_j \rangle^2 = \left( \sum_{a \in [d]} v_i[a] v_j[a] \right)^2 \\ &= \sum_{a \neq b \in [d]} v_i[a] \cdot v_i[b] \cdot v_j[a] \cdot v_j[b] + \sum_{a \in [d]} v_i[a]^2 \cdot v_j[a]^2. \end{aligned}$$

The first term is exactly  $M_\alpha[i, j]$ . For the second term,

$$\begin{aligned} \sum_{a \in [d]} v_i[a]^2 \cdot v_j[a]^2 &= \sum_{a \in [d]} \left( v_i[a]^2 - \frac{1}{d} \right) \left( v_j[a]^2 - \frac{1}{d} \right) + \frac{1}{d} (\|v_i\|_2^2 + \|v_j\|_2^2) - \frac{1}{d} \\ &= \underbrace{\sum_{a \in [d]} \left( v_i[a]^2 - \frac{1}{d} \right) \left( v_j[a]^2 - \frac{1}{d} \right)}_{M_\beta[i, j]} + \underbrace{\frac{\|v_i\|_2^2 - 1}{d}}_{\frac{1}{d} \eta_i} + \underbrace{\frac{\|v_j\|_2^2 - 1}{d}}_{\frac{1}{d} \eta_j} + \frac{1}{d}. \end{aligned}$$

Thus,  $M[i, j] = M_\alpha[i, j] + M_\beta[i, j] + \frac{1}{d} + \frac{1}{d} (1_m \cdot \eta^T + \eta \cdot 1_m^T) [i, j]$ .

## 78:8 Ellipsoid Fitting up to a Constant

For the diagonal entries, the right-hand side of the  $(i, i)$  entry is

$$\begin{aligned} M_D[i, i] + \left(1 + \frac{1}{d}\right) + \frac{1}{d} + \frac{2}{d}\eta_i &= \left(\|v_i\|_2^4 - \frac{2}{d}\|v_i\|_2^2 - 1\right) + 1 + \frac{2}{d} + \frac{2}{d}(\|v_i\|_2^2 - 1) \\ &= \|v_i\|_2^4 = M[i, i]. \end{aligned}$$

This completes the proof.  $\blacktriangleleft$

► **Remark 6.** The intention behind this decomposition is that for  $v_i \sim \mathcal{N}(0, \frac{1}{d}I_d)$ ,  $M_\alpha$ ,  $M_\beta$ ,  $M_D$  are all mean 0 (though their variances are not the same) since  $\mathbb{E}\|v_i\|_2^2 = 1$  and  $\mathbb{E}\|v_i\|_2^4 = 1 + \frac{2}{d}$ . Therefore, we expect  $\|M_\alpha + M_\beta + M_D\|_{\text{op}}$  to be small, which implies that  $A$  is positive definite and well-conditioned. Furthermore, observe that  $B$  has rank 2:

$$B = \frac{1}{d}J_m + \frac{1}{d}(1_m \cdot \eta^T + \eta \cdot 1_m^T) = \frac{1}{d} \begin{bmatrix} 1_m & \eta \end{bmatrix} \cdot \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} 1_m \\ \eta \end{bmatrix}. \quad (6)$$

### 2.3 Inverse of $M$

The decomposition of  $M$  into  $A$  and a rank-2 matrix  $B$  (Eq. (5)) allows us to apply the Woodbury matrix identity about the inverse of low-rank corrections of invertible matrices.

► **Fact 7 (Matrix Invertibility).** Suppose  $A \in \mathbb{R}^{n_1 \times n_1}$  and  $C \in \mathbb{R}^{n_2 \times n_2}$  are both invertible matrices, and  $U \in \mathbb{R}^{n_1 \times n_2}$  and  $V \in \mathbb{R}^{n_2 \times n_1}$  are arbitrary. Then,  $A + UCV$  is invertible if and only if  $C^{-1} + VA^{-1}U$  is invertible.

► **Fact 8 (Woodbury matrix identity [15]).** Suppose  $A \in \mathbb{R}^{n_1 \times n_1}$  and  $C \in \mathbb{R}^{n_2 \times n_2}$  are both invertible matrices, and  $U \in \mathbb{R}^{n_1 \times n_2}$  and  $V \in \mathbb{R}^{n_2 \times n_1}$  are arbitrary. Then

$$(A + UCV)^{-1} = A^{-1} - A^{-1}U(C^{-1} + VA^{-1}U)^{-1}VA^{-1}.$$

In light of Fact 8, we can write  $B$  in Eq. (6) as  $B = UCU^T$  where  $U = V^T = \frac{1}{\sqrt{d}} \begin{bmatrix} 1_m & \eta \end{bmatrix} \in \mathbb{R}^{m \times 2}$  and  $C = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}$ , and  $M = A + UCU^T$ . Note that  $C^{-1} = \begin{bmatrix} 0 & 1 \\ 1 & -1 \end{bmatrix}$ , and we have

$$C^{-1} + U^T A^{-1}U = \begin{bmatrix} \frac{1_m^T A^{-1} 1_m}{1 + \frac{\eta^T A^{-1} 1_m}{d}} & 1 + \frac{\eta^T A^{-1} 1_m}{d} \\ 1 + \frac{\eta^T A^{-1} 1_m}{d} & -1 + \frac{\eta^T A^{-1} \eta}{d} \end{bmatrix} := \begin{bmatrix} r & s \\ s & u \end{bmatrix}. \quad (7)$$

We first need to show that  $A$  is invertible. Recall from Eq. (5) that  $A = (1 + \frac{1}{d})I_m + M_\alpha + M_\beta + M_D$ . We will prove the following lemma, whose proof is deferred to the full version.

► **Lemma 9** ( $M_\alpha, M_\beta, M_D$  are bounded). Suppose  $m \leq cd^2$  for a small enough constant  $c$ . With probability  $1 - o_d(1)$ , we have

1.  $\|M_\alpha\|_{\text{op}} \leq 0.1$ ,
2.  $\|M_\beta\|_{\text{op}} \leq 0.1$ ,
3.  $\|M_D\|_{\text{op}} \leq O(\sqrt{\frac{\log d}{d}})$ .

As an immediate consequence, we get the following:

► **Lemma 10** ( $A$  is well-conditioned). With probability  $1 - o_d(1)$ , the matrix  $A$  from Eq. (5) is positive definite (hence full rank), and

$$0.5I_m \preceq A \preceq 1.5I_m.$$



**Proof.** Since  $A = (1 + \frac{1}{d})I_m + M_\alpha + M_\beta + M_D$ , by Lemma 9 the eigenvalues of  $A$  must lie within  $1 \pm 0.2 \pm \tilde{O}(1/\sqrt{d}) \in (0.5, 1.5)$  (we assume  $d$  is large). ◀

Next, from Fact 7, we can prove that  $M$  is invertible (Lemma 12) by showing that the  $2 \times 2$  matrix  $C^{-1} + U^T A^{-1} U$  is invertible, which is in fact equivalent to  $ru - s^2 \neq 0$ . We first need the following bound on the norm of  $\eta$ , whose proof is deferred to the full version.

▷ **Claim 11.** With probability at least  $1 - o_d(1)$ ,  $\|\eta\|_2^2 \leq (1 + o_d(1)) \frac{2m}{d}$ .

► **Lemma 12** (Bounds on  $r, s, u$ ;  $M$  is invertible). *Suppose  $m \leq cd^2$  for a small enough constant  $c$ . Let  $r, s, u \in \mathbb{R}$  be defined as in Eq. (7). With probability at least  $1 - o_d(1)$ , we have*

1.  $r \in \frac{m}{d} \cdot [2/3, 2]$ ,
2.  $|s| \leq O(\sqrt{d})$ ,
3.  $u \in [-1, -1/2]$ .

Thus, we have  $s^2 - ru \geq \Omega(\frac{m}{d})$ . As a consequence,  $M$  is invertible.

**Proof.** By Lemma 10, we know that  $\frac{2}{3}I_m \preceq A^{-1} \preceq 2I_m$ . Thus,  $r = \frac{1}{d} 1_m^T A^{-1} 1_m \in \frac{1}{d} \|1_m\|_2^2 \cdot [2/3, 2]$ , hence  $r \in \frac{m}{d} \cdot [2/3, 2]$ .

For  $s$ , we know that  $\|\eta\|_2^2 \leq (1 + o(1)) \frac{2m}{d}$  by Claim 11. Thus,

$$\frac{1}{d} |\eta^T A^{-1} 1_m| \leq \frac{1}{d} \|A^{-1}\|_{\text{op}} \cdot \|\eta\|_2 \cdot \|1_m\|_2 < (1 + o_d(1)) \cdot 2\sqrt{\frac{2m^2}{d^3}} \leq O(\sqrt{d}).$$

Thus,  $|s| = \left| 1 + \frac{\eta^T A^{-1} 1_m}{d} \right| \leq O(\sqrt{d})$ .

For  $u$ , we have

$$\frac{1}{d} |\eta^T A^{-1} \eta| \leq \frac{1}{d} \|A^{-1}\|_{\text{op}} \cdot \|\eta\|_2^2 < (1 + o_d(1)) \cdot \frac{4m}{d^2} < \frac{1}{2},$$

where the last inequality follows for some  $m < cd^2$  for small enough  $c$ . Thus,  $u = -1 + \frac{\eta^T A^{-1} \eta}{d} \in [-1, -1/2]$ .

With the bounds on  $r, s$  and  $u$ , we immediately get  $s^2 - ru \geq \Omega(\frac{m}{d})$ .

To prove that  $M$  is invertible, let us first recall that we write  $M = A + UCU^T$  where  $A$  is defined in Eq. (5) and  $U = V^T = \frac{1}{\sqrt{d}} [1_m \quad \eta] \in \mathbb{R}^{m \times 2}$  and  $C = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}$ .

By Lemma 10,  $A$  is invertible. Then by Fact 7, we know that  $M$  is invertible if and only if  $C^{-1} + U^T A^{-1} U := \begin{bmatrix} r & s \\ s & u \end{bmatrix}$  (see Eq. (7)) is invertible, which is equivalent to  $ru - s^2 \neq 0$ . Thus,  $s^2 - ru \geq \Omega(\frac{m}{d})$  suffices to conclude that  $M$  is invertible. ◀

## 2.4 Finishing the proof of Theorem 2

The final piece of proving Theorem 2 is to show that  $R = \sum_{i=1}^m w_i v_i v_i^T$  has spectral norm bounded by 1, which immediately implies that the candidate matrix  $\Lambda = I_d - R \succeq 0$ .

► **Lemma 13** ( $R$  is bounded). *There exists some absolute constant  $c_R$  s.t. for  $m \leq \frac{d^2}{c_R}$ , whp*

$$\|R\|_{\text{op}} \leq \frac{1}{2}.$$

The proof is deferred to the full version. In particular, we will write an expanded expression of  $M^{-1}$  and obtain a decomposition of  $R$ . Then, we prove tight spectral norm bounds for matrices in the decomposition, which then completes the proof of Lemma 13.

Combining Lemma 12 and Lemma 13 we can finish the proof of Theorem 2.

**Proof of Theorem 2.** The matrix  $M$  (recall Eq. (3)) is invertible due to Lemma 12, thus our candidate matrix  $\Lambda = I_d - R$  matrix defined in Definition 3 is well-defined. Furthermore, by the norm bound in Lemma 13, we have  $\|R\|_{\text{op}} < 1$ . This proves that  $\Lambda \succ 0$ . ◀

### 3 Machinery for tight norm bounds of graph matrices

One of the main technical contributions of this paper is providing tight spectral norm bounds (up to constants per vertex/edge) for *structured random matrices with correlated entries* (a.k.a. graph matrices). We note that prior to this work, most known norm bounds for such matrices are only tight up to some logarithmic factors [1], while not much is known in terms of precise bounds without log factors except for several specific cases (see e.g. [14]).

#### 3.1 Preliminaries

We first give a lightweight introduction to the theory of graph matrices. For interested readers who seek a thorough introduction or a more formal treatment, we refer them to its origin in a sequence of works in Sum-of-Squares lower bounds [3, 1]. We will follow the notations used in [1]. Throughout this section, we assume that there is an underlying (random) input matrix  $G$  and a Fourier basis  $\{\chi_t\}_{t \in \mathbb{N}}$ .

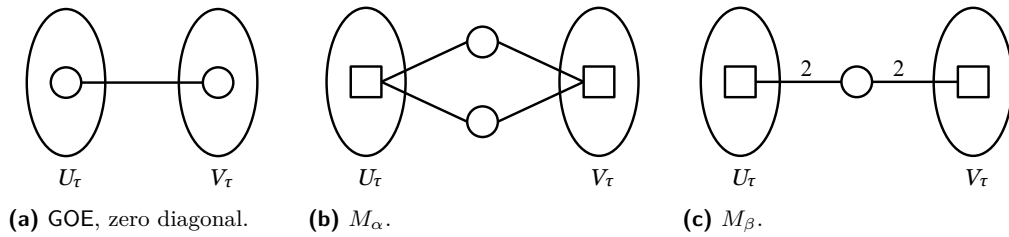
We first define *shapes*, which are representations of structured matrices whose entries depend on  $G$ .

► **Definition 14 (Shape).** A shape  $\tau$  is a tuple  $(V(\tau), U_\tau, V_\tau, E(\tau))$  associated with a (multi) graph  $(V(\tau), E(\tau))$ . Each vertex in  $V(\tau)$  is associated with a vertex-type that indicates the range of the labels for the particular vertex. Each edge  $e \in E(\tau)$  is also associated with a Fourier index  $t(e) \in \mathbb{N}$ . Moreover, we have  $U_\tau, V_\tau \subseteq V(\tau)$  as the left and right boundary of the shape.

We remind the reader that  $V_\tau$  should be distinguished from  $V(\tau)$ , where  $V_\tau$  is the right boundary set, while  $V(\tau)$  is the set of all vertices in the graph.

Figure 1 show the shapes for matrices  $M_\alpha$  and  $M_\beta$  defined in Proposition 5. For these shapes, there are two vertex-types (square and circle). The two ovals in each shape indicate the left and right boundaries  $U_\tau$  and  $V_\tau$ .

We next describe how to associate a shape to a matrix (given the underlying matrix  $G$ ).



■ **Figure 1** Graph matrix representation of a  $d \times d$  GOE matrix with zero diagonal, and the  $m \times m$  matrices  $M_\alpha$  and  $M_\beta$  as defined in Proposition 5. Square vertices take labels in  $[m]$  and circle vertices take labels in  $[d]$ . The two ovals indicate the left and right boundaries of the shapes. If an edge  $e$  is not labeled with an index, then  $t(e) = 1$  by default.

► **Definition 15 (Mapping of a shape).** Given a shape  $\tau$ , we call a function  $\sigma : V(\tau) \rightarrow \mathbb{N}$  a mapping of the shape if

1.  $\sigma$  assigns a label for each vertex according to its specified vertex-type;
2.  $\sigma$  is an injective mapping for vertices of the same type.

► **Definition 16** (Graph matrix for shape). *Given a shape  $\tau$ , we define its graphical matrix  $M_\tau$  to be the matrix indexed by all possible boundary labelings of  $S, T$ , and for each of its entry, we define*

$$M_\tau[S, T] = \sum_{\substack{\sigma: V(\tau) \rightarrow \mathbb{N} \\ \sigma(U_\tau)=S, \sigma(V_\tau)=T}} \prod_{e \in E(\tau)} \chi_{t(e)}(G[\sigma(e)]).$$

Observe that for each entry  $M_\tau[S, T]$ , since  $\sigma$  must map  $U_\tau$  and  $V_\tau$  to  $S$  and  $T$ ,  $M_\tau[S, T]$  is simply a sum over labelings of the “middle” vertices  $V(\tau) \setminus (U_\tau \cup V_\tau)$ . Take Figure 1 for example. Suppose  $G \in \mathbb{R}^{m \times d}$  and square and circle vertices take labels in  $[m]$  and  $[d]$  respectively, then we can write out the entries of the matrix: for  $i \neq j \in [m]$ ,

$$\begin{aligned} M_\alpha[i, j] &= \sum_{a \neq b \in [d]} \chi_1(G[i, a]) \cdot \chi_1(G[i, b]) \cdot \chi_1(G[j, a]) \cdot \chi_1(G[j, b]), \\ M_\beta[i, j] &= \sum_{a \in [d]} \chi_2(G[i, a]) \cdot \chi_2(G[j, a]). \end{aligned}$$

Note also that since  $\sigma$  must be injective for vertices of the same type and  $U_\tau \neq V_\tau$  in both examples, there is no mapping such that  $\sigma(U_\tau) = \sigma(V_\tau)$ . Thus, by Definition 16, both matrices have zeros on the diagonal.

**Adaptation to our setting.** The above is a general introduction for graph matrices. In this work, we specialize to the following setting:

- $G \in \mathbb{R}^{m \times d}$  is a random Gaussian matrix whose rows are  $v_1, \dots, v_m \sim \mathcal{N}(0, \frac{1}{d} I_d)$ .
- The Fourier characters  $\{\chi_t\}_{t \in \mathbb{N}}$  are the (scaled) Hermite polynomials.
- For all graph matrices that arise in our analysis,
  - $|S| = |T| = 1$ ,
  - There are two vertex-types: square vertices take labels in  $[m]$  and circle vertices take labels in  $[d]$ .

► **Remark 17.** For our technical analysis, we also employ our techniques on a generalization of graph matrices where we relax the injectivity condition. That said, for the purpose of illustrating our techniques, it suffices to consider ordinary graph matrices.

► **Definition 18** ( $D_V$  size constraint). *Let  $D_V$  be a size constraint such that for each graph matrix  $\tau$  considered in this work,  $|V(\tau)| \leq D_V$ .*

For concreteness, we will take  $D_V = \text{polylog}(d)$  throughout this work.

**Trace moment method.** For all our norm bounds, we will use the trace moment method: for any graph matrix  $M_\tau$  with underlying random matrix  $G$  and any  $q \in \mathbb{N}$ ,

$$\mathbb{E} \|M_\tau\|_{\text{op}}^{2q} \leq \mathbb{E} \text{tr} \left( (M_\tau M_\tau^T)^q \right) = \mathbb{E} \sum_{\substack{S_1, T_1, S_2, T_2, \dots, S_{q-1}, T_{q-1}: \\ \text{boundaries}}} M_\tau[S_1, T_1] M_\tau^T[T_1, S_2] \cdots M_\tau^T[T_{q-1}, S_1].$$

where the expectation is taken over  $G$ .

Notice that the summation is over *closed walks* across the boundaries:  $S_1 \rightarrow T_1 \rightarrow S_2 \rightarrow T_2 \rightarrow \cdots \rightarrow S_1$ , where  $S_1, T_1, \dots$  are boundary labelings of  $M_\tau$ . In particular, the walk is consist of  $2q$ -steps of a “block walk”, with the  $(2t - 1)$ -th step across a block described by  $M_\tau$  and the  $(2t)$ -th step across a block described by  $M_\tau^T$ .

The crucial observation is that after taking expectation, all closed walks must walk on each labeled edge (i.e., Fourier character) an *even* number of times, since all odd moments of the Fourier characters are zero. Therefore, bounding the matrix norm is reduced to bounding the contribution of all such walks.

$$\mathbb{E}\|M_\tau\|_{\text{op}}^{2q} \leq \sum_{\mathcal{P}: \text{ closed walk}} \prod_{e \in E(\mathcal{P})} \mathbb{E} \left[ \chi_{t(e)}(G[e])^{\text{mul}_{\mathcal{P}}(e)} \right], \quad (8)$$

where  $E(\mathcal{P})$  denotes the set of labeled edges used by the walk  $\mathcal{P}$ ,  $\text{mul}_{\mathcal{P}}(e)$  denotes the number of times  $e$  appears in the walk, and  $t(e)$  denotes the Fourier index (with slight abuse of notation).

► **Remark 19.** We remind the reader not to confuse vertices/edges in the walk with vertices/edges in the shape. The vertices in a walk are “labeled” by elements in  $[m]$  or  $[d]$  (depending on the vertex-type). Similarly, each edge  $e \in E(\mathcal{P})$  in a walk is labeled by an element in  $[m] \times [d]$ . We will use the terms “labeled vertex” and “labeled edge” unless it is clear from context.

### 3.2 Global bounds via a local analysis

Observe that Eq. (8) is a weighted sum of closed walks of length  $2q$ . To obtain an upper bound, the standard approach is to specify an efficient *encoding scheme* that uniquely identifies each closed walk, and then upper bound the total number of such encodings.

We begin by defining a step-labeling – a categorization of each step in the closed walk.

► **Definition 20** (Step-labeling). *For each step throughout the walk, we assign it the following label,*

1.  *$F$  (a fresh step): it uses a new labeled edge making the first appearance and leads to a destination not seen before;*
2.  *$S$  (a surprise step): it uses a new labeled edge to arrive at a vertex previously visited in the walk;*
3.  *$H$  (a high-mul step): it uses a labeled edge that appears before, and the edge is making a middle appearance (i.e., it will appear again in the subsequent walk);*
4.  *$R$  (a return step): it uses a labeled edge that appears before, and the edge is making its last appearance.*

*Analogously, for any shape  $\tau$ , we call  $\mathcal{L}_\tau : E(\tau) \rightarrow \{F, R, S, H\}$  a step-labeling of the block. The subscript  $\tau$  is ignored when it is clear.*

We note that the terms “fresh”, “high-mul” and “return” are adopted from the GOE matrix analysis in [14]. Next, to obtain a final bound for Eq. (8), we consider two *factors* for each step (which depend on the step-label):

1. **Vertex factor:** a combinatorial factor that specifies the destination of the step;
2. **Edge factor:** an analytical factor from the edge which accounts for the  $\mathbb{E}[\chi_{t(e)}(G[e])^{\text{mul}(e)}]$  term in Eq. (8).

For example, a vertex factor for an  $F$  step to a circle vertex can be  $d$ , an upper bound on the number of possible destinations. One can think of vertex factors as the information needed for a *decoder* to complete a closed walk. Essentially, the step-labeling and appropriate vertex factors should uniquely identify a closed walk, and combined with edge factors, we can obtain an upper bound for Eq. (8).

We note that the approach stated above is a *global* encoding scheme. One may proceed via a global analysis – carefully bounding the number of step-labelings allowed (e.g., using the fact that the  $F$  and  $R$  steps must form a Dyck word [14]), and then combining all vertex and edge factors to obtain a final bound. However, to get tight norm bounds for complicated graph matrices (like  $M_\alpha$ ), the global analysis becomes unwieldy.

**Local analysis.** One of our main insights is to use a *local* analysis. We now give a high-level overview of our strategy while deferring the specific details of our vertex/edge factor assignment scheme to subsequent sections. Recall that a closed walk consists of “block-steps” described by the shape  $\tau$ . Thus, we treat each walk as a “block walk” and bound the contributions of a walk block by block. This prompts us to bound the contribution of the walk at a given block-step to the final trace in Eq. (8) by

$$\text{vtxcost} \cdot \text{edgeval} \leq B_q(\tau)$$

where  $B_q(\tau)$  is some desired upper bound that depends on the vertex/edge factor assignment scheme. We define it formally in the following.

► **Definition 21** (Block value function). *Fix  $q \in \mathbb{N}$  and a shape  $\tau$ . For any vertex/edge factor assignment scheme, we call  $B_q(\tau)$  a valid block-value function for  $\tau$  of the given scheme if*

$$\mathbb{E} [\text{tr}((M_\tau M_\tau^T)^q)] \leq (\text{matrix dimension}) \cdot B_q(\tau)^{2q},$$

and for each block-step  $\text{BlockStep}_i$  throughout the walk,

$$\text{vtxcost}(\text{BlockStep}_i) \cdot \text{edgeval}(\text{BlockStep}_i) \leq B_q(\tau).$$

We point out that the block-value function  $B$  should be considered as a function of both the shape  $\tau$  and the length of the walk  $q$  (we will drop the subscript when it is clear throughout this work), and it also depends on the assignment scheme. Thus, our task is to find a vertex/edge factor assignment scheme such that  $B_q(\tau)$  is as small as possible. Moreover, the matrix dimension, which is at most  $\text{poly}(d)$  in our case, is the factor that comes up in the start of the walk to specify the original vertex, and can be ignored as it is ultimately an  $1 + o(1)$  factor once we take a long enough walk.

Given Definition 21, the norm bound follows immediately.

► **Proposition 22.** *Let  $M_\tau$  be a graph matrix with dimension  $\text{poly}(d)$ , and let  $q = \Omega(\log^2 d)$ . Suppose  $B_q(\tau)$  is a valid block-value function. Then, with probability  $1 - \frac{1}{\text{poly}(d)}$ ,*

$$\|M_\tau\|_{\text{op}} \leq (1 + o_d(1)) \cdot B_q(\tau).$$

**Proof.** We apply Markov’s inequality: for any  $\varepsilon > 0$ ,

$$\begin{aligned} \Pr[\|M_\tau\|_{\text{op}} > (1 + \varepsilon)B_q(\tau)] &\leq \Pr[\text{tr}((M_\tau M_\tau^T)^q) > (1 + \varepsilon)^{2q} B_q(\tau)^{2q}] \\ &\leq (1 + \varepsilon)^{-2q} \text{poly}(d) \\ &\leq \frac{1}{\text{poly}(d)} \end{aligned}$$

for  $q = \Omega(\frac{1}{\varepsilon} \log d)$ . Setting  $\varepsilon = \frac{1}{\log d}$ , we can conclude that  $\|M_\tau\|_{\text{op}} \leq (1 + o_d(1)) \cdot B_q(\tau)$  with high probability. ◀

The next proposition shows that we can easily obtain a valid  $B_q(\tau)$  once we have an appropriate factor assignment scheme.

► **Proposition 23.** *For any graph matrix  $M_\tau$  and any valid factor assignment scheme,*

$$B_q(\tau) = \sum_{\mathcal{L}: \text{step-labelings for } E(\tau)} \text{vtxcost}(\mathcal{L}) \cdot \text{edgeval}(\mathcal{L})$$

*is a valid block-value function for  $\tau$ .*

**Proof.** The second requirement in Definition 21 is clear. For the first requirement, observe that the trace can be bounded by the matrix dimension (specifying the start of the walk) times

$$\sum_{\substack{\mathcal{L}_1, \dots, \mathcal{L}_{2q}: \\ \text{step-labelings for } E(\tau)}} \prod_{i=1}^{2q} \text{vtxcost}(\mathcal{L}_i) \cdot \text{edgeval}(\mathcal{L}_i) \leq \left( \sum_{\mathcal{L}: \text{step-labelings for } E(\tau)} \text{vtxcost}(\mathcal{L}) \cdot \text{edgeval}(\mathcal{L}) \right)^{2q} \cdot \blacktriangleleft$$

With this set-up, the main task is then to find an appropriate vertex/edge factor assignment scheme and obtain a good upper bound on  $B_q(\tau)$ .

### 3.3 Vertex factor assignment scheme

We now proceed to bound the vertex factors for each step-label. We note that in this section, “vertices” refer to “labeled vertices” in the walk (having labels in  $[m]$  or  $[d]$ ; recall Remark 19). First, we define the weight of a square (resp. circle) vertex to be  $m$  (resp.  $d$ ), since we need an element in  $[m]$  (resp.  $[d]$ ) to specify which vertex to go to in the walk.

We first show a “naive” vertex factor assignment scheme. In the following scheme, we use a *potential unforced return* factor, denoted  $\text{Pur}$ , to specify the destination of any  $R$  step. We will defer the specific details of  $\text{Pur}$  to Section 3.5.

#### Vanilla vertex factor assignment scheme.

1. For each vertex  $i$  that first appears via an  $F$  step, a label in  $\text{weight}(i)$  is required;
2. For each vertex  $i$  that appears beyond the first time:
  - If it is arrived via an  $R$  step, the destination may need to be specified, and this is captured by the  $\text{Pur}$  factor.
  - If it is *not* arrived via an  $R$  step, then it must be an  $S$  or  $H$  step. A vertex cost in  $2q \cdot D_V$  is sufficient to identify the destination, where we recall  $2q$  is the length of our walk, and  $D_V$  the size upper bound of each block.

The first thing to check is that this scheme combined with an step-labeling uniquely identifies a closed walk (given the start of the walk). This is immediate for  $F$  and  $R$  steps by definition. For  $S$  and  $H$  steps, since the destination is visited before in the walk,  $2q \cdot D_V$  is sufficient as it is an upper bound on the number of vertices in the walk.

A potential complication with analyzing the above assignment scheme directly is that it exhibits a significant difference in the vertex factors. For example, consider a vertex that appears only twice in the walk on a tree. Its first appearance requires a label in  $[n]$ , while its subsequent appearance does not require any cost if it is reached using an  $R$  step because backtracking from a tree is fixed (since there is only one parent). This disparity can result in a very loose upper bound for the trace when applying Proposition 23; in fact, the norm bound for  $M_\tau$  obtained in this manner is equivalent to using the naive row-sum bound.

**Redistribution.** One of our main technical insights is to split the factors such that both first and last appearance contributes a factor of comparable magnitude; we call this *redistribution*.

We first formally define “appearance” in a block-step to clarify our terminology,

► **Definition 24** (Vertex appearance in block-step). *Each labeled vertex appearance can be “first”, “middle” and “last”. Moreover, each vertex on the block-step boundary ( $U_\tau$  or  $V_\tau$ ) appears in both adjacent blocks.*

For example, suppose a vertex first appears in the right-boundary of block  $i$  and last appears in the left-boundary of block  $j$ , then it will make middle appearances in the left-boundary of block  $i + 1$  and right-boundary of block  $j - 1$  as well.

We are now ready to introduce the following vertex-factor assignment scheme with redistribution that assigns vertex-factor to each vertex’s appearance to handle the disparity.

**Vertex factor assignment scheme with redistribution.**

1. For each vertex  $i$  that makes its first appearance, assign a cost of  $\sqrt{\text{weight}(i)}$ ;
2. For any vertex’s middle appearance, if it is not arrived at via an  $R$  step, assign a cost of  $2q \cdot D_V$  (where we recall  $2q$  is the length of our walk, and  $D_V$  the size constraint of each block);
3. For any vertex’s middle appearance, if it is arrived via an  $R$  step, its cost is captured by  $\text{Pur}$ ;
4. For each vertex  $i$  that makes its last appearance, assign a cost of  $\sqrt{\text{weight}(i)}$  that serves as a *backpay*.

**Deducing vertex factor from local step-labeling.** As presented, the vertex factor assignment scheme requires knowing which vertex is making first/middle/last appearance. We further show that the vertex appearances, or more accurately, an upper bound of the vertex factors, can be deduced by a given step-labeling of the block. Fix traversal direction from  $U$  to  $V$ ,

**Localized vertex factor assignment from step-labeling.**

1. For any vertex  $v$  that is on the left-boundary  $U$ , it cannot be making the first appearance since it necessarily appears in the previous block;
2. For any vertex  $v$  that is on the right-boundary  $V$ , it cannot be making the last appearance since it necessarily appears in the subsequent block;
3. For any vertex  $v$  reached via some  $S/R/H$  step, it cannot be making its first appearance;
4. For any vertex  $v$  that incident to some  $F/S/H$  step, it cannot be making its last appearance since the edge necessarily appears again.

The first two points are due to Definition 24. The last point is because each labeled edge (i.e., Fourier character) must be traversed by an  $R$  step to close it.

### 3.4 Bounding edge-factors

To bound the contribution of the walks, we need to consider factors coming from the edges traversed by the walk. Recall from Eq. (8) that each edge  $e$  in a closed walk  $P$  gets a factor  $\mathbb{E}[\chi_{t(e)}^{\text{mul}_P(e)}]$ , where  $t(e)$  is the Fourier index associated with the edge.

In our case, the Fourier characters are the scaled Hermite polynomials. Recall that we assume that our vectors are sampled as  $v_i \sim \mathcal{N}(0, \frac{1}{d}I_d)$ . Thus, we define the polynomials  $\{H_t\}_{t \in \mathbb{N}}$  such that they are orthogonal and  $\mathbb{E}_{x \sim \mathcal{N}(0, 1/d)}[H_t(x)^2] = t! \cdot d^{-t}$ . Specifically,



## 78:16 Ellipsoid Fitting up to a Constant

1.  $H_1(x) = x$ ,
2.  $H_2(x) = x^2 - \frac{1}{d}$ .

We first state the following bound on the moments of  $H_t$ , which follows directly from standard bounds on the moments of Hermite polynomials:

► **Fact 25** (Moments of Hermite polynomials). *Let  $d \in \mathbb{N}$ . For any  $t \in \mathbb{N}$  and even  $k \in \mathbb{N}$ ,*

$$\mathbb{E}_{x \sim \mathcal{N}(0, 1/d)} [H_t(x)^k] \leq \frac{1}{d^{kt/2}} (k-1)^{kt/2} (t!)^{k/2} \leq (t!)^{k/2} \left(\frac{k}{d}\right)^{kt/2}.$$

For matrices that arise in our analysis, we only have  $H_1$  and  $H_2$  edges. The following is our edge-factor assignment scheme to account for contributions from the Fourier characters.

### Edge-factor assignment scheme.

For an  $H_1$  edge,

1.  $F/S$ : assign a factor of  $\frac{1}{\sqrt{d}}$  for its first appearance;
2.  $H$ : assign a factor of  $\frac{2q}{\sqrt{d}}$  for its middle appearance;
3.  $R$ : assign a factor of  $\frac{1}{\sqrt{d}}$  for its last appearance.

For an  $H_2$  edge,

1.  $F/S$ : assign a factor of  $\frac{\sqrt{2}}{d}$  for its first appearance (equivalently, we can view a single  $H_2$  edge as two edge-copies of  $H_1$  and assign each a factor of  $\frac{\sqrt{2}}{\sqrt{d}}$  which is a valid upper bound);
2.  $H$ : assign a factor of  $\frac{8q^2}{d}$  for its middle appearance;
3.  $R$ : assign a factor of  $\frac{\sqrt{2}}{d}$  for its last appearance (equivalently, we can view a single  $H_2$  edge as two edge-copies of  $H_1$  and assign each a factor of  $\frac{\sqrt{2}}{\sqrt{d}}$  which is a valid upper bound).

► **Proposition 26.** *The above scheme correctly accounts for the edge factors from  $H_1$  and  $H_2$  edges.*

**Proof.** If an edge has multiplicity 2, then it must be traversed by one  $F/S$  step and one  $R$  step.

- If it is an  $H_1$  edge, then the scheme assigns a factor  $\frac{1}{d}$ , which equals  $\mathbb{E}_{x \sim \mathcal{N}(0, 1/d)} [H_1(x)^2]$ .
- If it is an  $H_2$  edge, then the scheme assigns a factor  $\frac{2}{d^2}$ , which equals  $\mathbb{E}_{x \sim \mathcal{N}(0, 1/d)} [H_2(x)^2]$ .

For an edge with multiplicity  $k > 2$ , it must be traversed by one  $F/S$  step, one  $R$  step and  $k-2$   $H$  steps. Moreover, since  $k$  is even and  $2q$  is the length of the walk, we have  $4 \leq k \leq 2q$ .

- If it is an  $H_1$  edge, then the scheme assigns a factor  $\frac{1}{d} \cdot \left(\frac{2q}{\sqrt{d}}\right)^{k-2} \geq d^{-k/2} (2q)^{k/2} \geq \left(\frac{k}{d}\right)^{k/2}$ .

By Fact 25, it is an upper bound on  $\mathbb{E}_{x \sim \mathcal{N}(0, 1/d)} [H_1(x)^k]$ .

- If it is an  $H_2$  edge, then the scheme assigns a factor  $\frac{2}{d^2} \cdot \left(\frac{8q^2}{d}\right)^{k-2} \geq d^{-k} 2^{k/2} (2q)^k \geq 2^{k/2} \left(\frac{k}{d}\right)^k$ . By Fact 25, it is an upper bound on  $\mathbb{E}_{x \sim \mathcal{N}(0, 1/d)} [H_2(x)^k]$ .

This shows that the edge factor assignment scheme above is correct. ◀

## 3.5 Bounding return cost (Pur factors)

In our vertex factor assignment scheme described in Section 3.3, we use a *potential unforced return* factor, denoted  $\text{Pur}$ , to specify the destination of any return ( $R$ ) step. Note that the term “unforced return” is adopted from [14] as well. In this section, we complete the bound of vertex factors by bounding the  $\text{Pur}$  factor.



For starters, we will define a potential function for each vertex at time  $t$ , which measures the number of returns  $R$  pushed out from the particular vertex by time  $t$  that may require a label in  $2q \cdot D_V$ . Notice that a label in  $2q \cdot D_V$  is sufficient for any destination vertex arrived via an  $R$  step because the vertex appears before; however, this may be a loose bound.

We observe the following: a label in  $2q \cdot D_V$  may be spared if the vertex is incident to only one un-closed  $F/S$  edge; we call this a *forced* return. Formally, we define a return step as *unforced* if it does not fall into the above categories,

► **Definition 27** (Unforced return). *We call a return ( $R$ ) step an unforced return if the source vertex is incident to more than 1 (or 2 in the case of a square vertex) unclosed edge.*

We now proceed to formalize the above two observations by introducing a potential function to help us bound the number of unforced returns from any given vertex throughout the walk. The number of unforced returns throughout the walk would then be immediately given once we sum over all vertices in the walk.

► **Definition 28** (Potential-unforced-return factor  $Pur$ ). *For any time  $t$  and vertex  $v$ , let  $Pur_t(v)$  be defined as the number of potential unforced return from  $v$  throughout the walk until time  $t$ .*

### 3.5.1 $Pur$ bound for circle vertices

In our setting, each circle vertex pushes out at most 1 edge during the walk, analogous to the case of typical adjacency matrix. This serves as a starting point for our  $Pur$  bound for circle vertices.

► **Lemma 29** (Bounding  $Pur_t$  for circle vertices). *For any time  $t$ , suppose the walker is currently at a circle vertex  $v$ , then*

$$\begin{aligned} Pur_t(v) &\leq \#(R \text{ steps closed from } v) + \#(\text{unclosed edges incident to } v \text{ at time } t) - 1 \\ &\leq 2 \cdot s_t(v) + h_t(v), \end{aligned}$$

where we define the following counter functions:

1.  $s_t(v)$  is the number of  $S$  steps arriving at  $v$  by time  $t$ ;
2.  $h_t(v)$  is the number of  $H$  steps arriving at  $v$  by time  $t$ .

**Proof.** We first prove the first inequality. The  $R$  steps closed from  $v$  may all be unforced returns, and the unclosed edges incident to  $v$  may be closed by unforced returns in the future. Note that we have a  $-1$  in the above bound because for each vertex we may by default assume the return is using a particular edge, hence at each time we know there is an edge presumed-to-be forced.

We prove the second inequality by induction. Define  $P_t(v) := \#(R \text{ steps closed from } v) + \#(\text{unclosed edges incident to } v \text{ at time } t) - 1$  for convenience. At the time when  $v$  is first created by an  $F$  step,  $P_t(v) = 0$  (1 open edge minus 1) and  $s_t(v) = h_t(v) = 0$ .

At time  $t$ , suppose the last time  $v$  was visited was at time  $t' < t$ , and suppose that the inequality holds true for  $t'$ . Note that at time  $t' + 1$ ,  $P_{t'+1}(v) = P_{t'}(v) + 1$  if a new edge was created by an  $F$  or  $N$  step leaving  $v$ , otherwise  $P_{t'+1}(v) = P_{t'}(v)$  (for  $R$  step it adds 1 to the number of closed edges closed from  $v$ , but decreases 1 open edge). On the other hand,  $s_{t'}(v)$  and  $h_{t'}(v)$  remain the same (we don't count out-going steps for  $s_t(v), h_t(v)$ ).

When we reach  $v$  at time  $t$ , we case on the type of steps:

- Arriving by an  $R$  step: the edge is now closed, but the  $R$  step was not from  $v$ . So  $P_t(v) = P_{t'+1}(v) - 1 \leq P_{t'}(v)$ , while  $s_t(v) = s_{t'}(v)$  and  $h_t(v) = h_{t'}(v)$ .
  - Arriving by an  $S$  step: the edge is new, so  $P_t(v) = P_{t'+1}(v) + 1 \leq P_{t'}(v) + 2$ , and we have  $s_t(v) = s_{t'}(v) + 1$ .
  - Arriving by an  $H$  step:  $P_t(v) = P_{t'+1}(v) \leq P_{t'}(v) + 1$ , and  $h_t(v) = h_{t'}(v) + 1$ .
- In all three cases, assuming  $P_{t'}(v) \leq 2 \cdot s_{t'}(v) + h_{t'}(v)$ , we have  $P_t(v) \leq 2 \cdot s_t(v) + h_t(v)$ , completing the induction.  $\blacktriangleleft$

### 3.5.2 Pur bound for square vertices

The argument of Lemma 29 does not apply well for vertices incident to multiple edges in a single step. In particular, this may happen for square vertices in  $M_\alpha$  as each is arrived via 2 edges and each pushes out 2 edges (recall Figure 1). This is not an issue for  $M_\beta$ , but we will treat square vertices in  $M_\beta$  the same way to unify the analysis; in the context of Pur for square vertices, one may think of  $M_\beta$  as collapsing the two circle vertices in  $M_\alpha$ .

To handle this issue, we observe that it suffices for us to pay an extra cost of [2] for each square vertex, which would allow us to further presume 2 edges being forced. We then generalize the prior argument to capture this change.

► **Lemma 30** (Bounding  $\text{Pur}_t$  for square vertices). *For any time  $t$ , suppose the walker is currently at a square vertex  $v$ , then*

$$\begin{aligned} \text{Pur}_t(v) &\leq \#(R \text{ steps closed from } v) + \#(\text{unclosed edges incident to } v \text{ at time } t) - 2 \\ &\leq 2(s_t(v) + h_t(v)). \end{aligned}$$

where  $s_t(v)$  and  $h_t(v)$  are the number of  $S$  and  $H$  steps arriving at  $v$  by time  $t$ , respectively.

**Proof.** We prove this by induction. Note that this is immediate for the base case when  $v$  first appears since a square vertex is incident to 2 edges. Define  $P_t(v) := \#(R \text{ steps closed from } v) + \#(\text{unclosed edges incident to } v \text{ at time } t) - 2$  for convenience. Suppose the inequality is true at time  $t'$ , and assume vertex  $v$  appears again at time  $t$ . The departure at time  $t' + 1$  from  $v$  may open up at most 2 edges, hence  $P_{t'+1}(v) \leq P_{t'}(v) + 2$ .

When we reach  $v$  at time  $t$  (via 2 edges), we case on the type of steps:

- Arriving by two  $R$  steps: the two edges closed by the  $R$  steps are not closed from  $v$ . So  $P_t(v) = P_{t'+1}(v) - 2 \leq P_{t'}(v)$ , while  $s_t(v) = s_{t'}(v)$  and  $h_t(v) = h_{t'}(v)$ .
- Arriving by one  $S/H$  and one  $R$  step: in this case,  $P_t(v) = P_{t'+1}(v) \leq P_{t'}(v) + 2$  and  $s_t(v) + h_t(v) = s_{t'}(v) + h_{t'}(v) + 1$ .
- Arriving by two  $S/H$  steps: in this case,  $P_t(v) = P_{t'+1}(v) + 2 \leq P_{t'}(v) + 4$ , whereas  $s_t(v) + h_t(v) = s_{t'}(v) + h_{t'}(v) + 2$ .

In all three cases, we have  $P_t(v) \leq 2(s_t(v) + h_t(v))$ , completing the induction.  $\blacktriangleleft$

► **Corollary 31.** *For each surprise/high-mul visit, it suffices for us to assign a Pur factor of 2, which is a cost of  $(2q \cdot D_V)^2$  so that each Pur factor throughout the walk is assigned.*

### 3.6 Wrapping up with a toy example

Recall Proposition 23 that for a graph matrix of shape  $\tau$ ,

$$B_q(\tau) = \sum_{\mathcal{L}: \text{step-labelings for } E(\tau)} \text{vtxcost}(\mathcal{L}) \cdot \text{edgeval}(\mathcal{L}) \quad (9)$$

is a valid block-value function for  $\tau$  (Definition 21). Moreover, by Proposition 22, we can take  $q = \text{polylog}(d)$  and conclude that with probability  $1 - o(1)$ ,

$$\|M_\tau\|_{\text{op}} \leq (1 + o(1)) \cdot B_q(\tau).$$

For each given shape, it suffices for us to bound the block-value for each edge-labeling. We demonstrate how this may be readily done given the above bounds using the GOE example, and defer the analysis of the specific matrices that show up in our setting to the full version of the paper.

### 3.6.1 Tight bound for GOE

We now show how the above framework allows us to readily deduce a tight norm bound for  $G \sim \text{GOE}(0, \frac{1}{d})$ , where  $G$  is a  $d \times d$  symmetric matrix with each (off-diagonal) entry sampled from  $\mathcal{N}(0, \frac{1}{d})$ . It is well-known that the correct norm of  $G$  is  $2 + o_d(1)$  [14]. Figure 1a shows the shape  $\tau$  associated with  $G$ , which simply consists of one edge. We now proceed to bound Eq. (9).

**Edge factor.** According to our edge factor scheme described in Section 3.4 (for  $H_1$  edges), an  $F/R/S$  step-label gets a factor of  $\frac{1}{\sqrt{d}}$  while an  $H$  step-label gets  $\frac{2q}{\sqrt{d}}$ .

**Pur factor.** By Lemma 29, there is no Pur factor for  $F/R$ , while  $S$  and  $H$  get 2 and 1 Pur factors respectively.

**Vertex factor.** The weight of a circle vertex is  $d$ , thus any vertex making a first or last appearance gets a factor of  $\sqrt{d}$ . We now case on the step-label and apply the vertex factor assignment scheme described in Section 3.3.

- $F$ : the vertex in  $U_\tau$  must be making a middle appearance; it is not first due to Definition 24, and it is not last as otherwise the edge appears only once throughout the walk. The vertex in  $V_\tau$  is making a first appearance, so it gets a factor of  $\sqrt{d}$ ;
- $R$ : the vertex in  $V_\tau$  is making a middle appearance, since it is incident to an  $R$  edge (hence not first appearance), and it is on the boundary hence bound to appear again the next block. The vertex in  $U_\tau$  may be making its last appearance, so it gets a factor of  $\sqrt{d}$ ;
- $S$ : the vertex in  $U_\tau$  is making a middle appearance (same as  $F$ ), and the vertex in  $V_\tau$  is making a middle appearance since it cannot be first and must appear again. In addition, it gets 2 factors of Pur, which gives a bound of  $(2q \cdot D_V)^2$ ;
- $H$ : analogous to the above, both vertices are making middle appearance, and it gets 1 factor of Pur, giving a bound of  $2q \cdot D_V$ .

Combining the vertex and edge factors, we can bound Eq. (9):

$$B_q(\tau) = \sqrt{d} \cdot \frac{1}{\sqrt{d}} + \sqrt{d} \cdot \frac{1}{\sqrt{d}} + (2q \cdot D_V)^2 \cdot \frac{1}{\sqrt{d}} + (2q \cdot D_V) \cdot \frac{2q}{\sqrt{d}} \leq 2 + o_d(1),$$

since  $q$  and  $D_V$  are both  $\text{polylog}(d)$ . Therefore, by Proposition 22, we can conclude that  $\|G\|_{\text{op}} \leq 2 + o_d(1)$  with high probability, which is the correct bound.

---

## References

- 1 Kwangjun Ahn, Dhruv Medarametla, and Aaron Potechin. Graph matrices: Norm bounds and applications. *arXiv preprint*, 2016. [arXiv:1604.03423](#).
- 2 Mitali Bafna, Jun-Ting Hsieh, Pravesh K Kothari, and Jeff Xu. Polynomial-Time Power-Sum Decomposition of Polynomials. In *2022 IEEE 63rd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 956–967. IEEE, 2022.

- 3 Boaz Barak, Samuel Hopkins, Jonathan Kelner, Pravesh K Kothari, Ankur Moitra, and Aaron Potechin. A Nearly Tight Sum-of-Squares Lower Bound for the Planted Clique Problem. *SIAM Journal on Computing*, 48(2):687–735, 2019.
- 4 Mrinalkanti Ghosh, Fernando Granha Jeronimo, Chris Jones, Aaron Potechin, and Goutham Rajendran. Sum-of-squares lower bounds for Sherrington-Kirkpatrick via planted affine planes. In *2020 IEEE 61st Annual Symposium on Foundations of Computer Science (FOCS)*, pages 954–965. IEEE, 2020.
- 5 Christopher Hoffman, Matthew Kahle, and Elliot Paquette. Spectral gaps of random graphs and applications. *International Mathematics Research Notices*, 2019.
- 6 Jun-Ting Hsieh and Pravesh K Kothari. Algorithmic Thresholds for Refuting Random Polynomial Systems. In *Proceedings of the 2022 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1154–1203. SIAM, 2022.
- 7 Chris Jones, Aaron Potechin, Goutham Rajendran, Madhur Tulsiani, and Jeff Xu. Sum-of-squares lower bounds for sparse independent set. In *2021 IEEE 62nd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 406–416. IEEE, 2022.
- 8 Daniel M Kane and Ilias Diakonikolas. A Nearly Tight Bound for Fitting an Ellipsoid to Gaussian Random Points. *arXiv preprint*, 2022. [arXiv:2212.11221](#).
- 9 Sidhanth Mohanty, Prasad Raghavendra, and Jeff Xu. Lifting sum-of-squares lower bounds: degree-2 to degree-4. In *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing*, pages 840–853, 2020.
- 10 Aaron Potechin, Paxton Turner, Prayaag Venkat, and Alexander S Wein. Near-optimal fitting of ellipsoids to random points. *arXiv preprint*, 2022. [arXiv:2208.09493](#).
- 11 James Saunderson. *Subspace identification via convex optimization*. PhD thesis, Massachusetts Institute of Technology, 2011.
- 12 James Saunderson, Venkat Chandrasekaran, Pablo A Parrilo, and Alan S Willsky. Diagonal and low-rank matrix decompositions, correlation matrices, and ellipsoid fitting. *SIAM Journal on Matrix Analysis and Applications*, 33(4):1395–1416, 2012.
- 13 James Saunderson, Pablo A Parrilo, and Alan S Willsky. Diagonal and low-rank decompositions and fitting ellipsoids to random points. In *52nd IEEE Conference on Decision and Control*, pages 6031–6036. IEEE, 2013.
- 14 Terence Tao. *Topics in random matrix theory*, volume 132. American Mathematical Soc., 2012.
- 15 Max A Woodbury. Inverting modified matrices. *Memorandum Rept. 42, Statistical Research Group*, 1950.