Ellipsoid Fitting Up to a Constant

Jun-Ting Hsieh*   Pravesh K. Kothari†   Aaron Potechin‡   Jeff Xu§

July 13, 2023

Abstract

In [Sau11, SPW13], 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 $v_1, v_2, \ldots, v_m$ with high probability when the v_is 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 $1 \leqslant i \leqslant m$ — 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 [PTVW22] and Kane and Diakonikolas [KD22] proved that $m \geqslant d^2 / \log^{O(1)}(d)$ via certain explicit constructions.

In this work, we give a substantially tighter analysis of their construction to prove that $m \geqslant d^2/C$ 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 [BHK+19]. 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.

Contents

1 Introduction

What's the largest m so that for m points $v_1, v_2, \ldots, v_m \in \mathbb{R}^d$ sampled independently from the d-dimensional standard Gaussian distribution $\mathcal{N}(0, I_d)$, there exists an ellipsoid that passes through each of the v_is with high probability? This latter condition is equivalent to asking for a positive semidefinite matrix Λ such that $v_i^\top \Lambda v_i = 1$ for $1 \leqslant i \leqslant m$ and thus, equivalently, the question asks for the largest m such that the basic stochastic semidefinite program above remains feasible with high probability.

It is not hard to prove that for any $m \leqslant d + 1$, an ellipsoid as above exists with high probability over the v_is [SCPW12]. On the other hand, since the dimension of the smallest linear subspace that contains the positive semidefinite cone of $d \times d$ matrices is $\binom{d+1}{2} \sim d^2/2$, it is easy to prove that for $m \gg d^2/2$, there cannot be an ellipsoid passing through v_is with high probability.

In 2013, Saunderson, Parrilo and Willsky [SPW13] studied this basic geometric question and conjectured that there is a sharp phase transition for the problem (from feasibility/existence of an ellipsoid to non-existence of an ellipsoid) as m crosses $d^2/4$.

Conjecture 1.1 (SCPW conjecture). *Let $\varepsilon > 0$ be a constant and $v_1, \ldots, v_m \sim \mathcal{N}(0, I_d)$ be i.i.d. standard Gaussian vectors in $\mathbb{R}^d$. Then,*

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

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

This bound is $1/2$ of the dimension of the smallest linear subspace containing the positive semidefinite cone of $d \times d$ matrices. Said differently, the SCPW conjecture (developed in a sequence of works [Sau11, SCPW12, SPW13]) posits that the positive semidefiniteness constraint "forces" a drop of a factor 2 in the threshold m for infeasibility. The SCPW conjecture was motivated by results of numerical experiments (see also the experiments presented in [PTVW22]).

Early on, [Sau11, SPW13] established the existence of a feasible ellipsoid for any $m \leqslant O(d^{6/5-\varepsilon})$ whp. Recently, there has been a new wave of progress on this bound. A recent result [GJJ+20] on establishing Sum-of-Squares lower bounds for the Sherringtin-Kirkpatrick model, as a corollary yields an estimate of $m \leqslant O(d^{3/2-\varepsilon})$. In fact, though not explicitly stated, their work already contains ideas that imply a significantly stronger bound of $m \leqslant d^2/\operatorname{polylog}(d)$. Very recently, two independent works [PTVW22] and [KD22] analyzed two slightly different explicit constructions for Λ to recover a similar bound of $m \leqslant d^2/\operatorname{polylog}(d)$. In their works [PTVW22, KD22], 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 work achieves this goal. Specifically, we prove:

Theorem 1.2 (Main result). *For $m \leqslant cd^2$ for some universal constant $c > 0$ and $v_1, \ldots, v_m \sim \mathcal{N}(0, I_d)$ drawn independently, with probability at least $1 - o_d(1)$, there exists an ellipsoid passing through each v_i.*

Remark 1.3. We note that the failure probability is in fact 2^{-d^ε} for some small constant ε due to the nature of our proof for matrix norm bounds.

We establish Theorem 1.2 by analyzing the construction of Kane and Diakonikolas [KD22] (which is a variant of the construction proposed in [PTVW22]). Our argument can be used to

recover a bound of $c \sim 1/10^8$. We have not tried to optimize this bound. Numerical experiments suggest that the [KD22] construction we analyze cannot approach $c = 1/4$, so establishing the sharp constant in the SCPW conjecture will likely need new ideas. Table 1 shows a summary of our result compared to prior work.

Our key idea departs from the analysis technique of [KD22] and instead relies on the method of *graphical matrix decomposition*. 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 [BHK+19, HKP19, AMP16, JPR+22] 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 [BHKX22]).

The key technical work in the analysis then becomes understanding the smallest and the 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 [BHKX22] 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). Our analysis builds on their conceptual framework but with significant technical upgrades. This is because the quantitative bounds proved in [BHKX22] do not allow us to directly obtain an improvement on the previous estimates [KD22].

Our main technical contribution is a new method 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.

Concurrent Work We note that a concurrent work of Bandeira et. al. [BMMP23] also obtains a sharper analysis of [KD22] to establish a similar result as this work. They analyze the same construction of identity perturbation as us. In their work, [BMMP23] ask the question of obtaining estimates that hold with inverse exponential failure probability (as opposed to inverse polynomial failure probability that their work establishes). They also outline a proof strategy that could potentially achieve this goal. We note that our proof does indeed recover an inverse exponential failure probability naturally.

1.1 Technical overview

Following the convention of [KD22], for the rest of the paper we will assume that $v_1, \ldots, v_m \sim \mathcal{N}(0, \frac{1}{d}I_d)$ such that each vector has expected norm 1. Note that this does not change the problem as we can simply scale Λ.

Our construction of Λ is the "identity perturbation construction", which is the same one analyzed in [KD22] and was proposed in [PTVW22]. 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 Λ as a perturbation of I_d: $\Lambda = I_d - \sum_{i=1}^m w_i v_i v_i^T$, where $w = (w_1, \ldots, w_m) \in \mathbb{R}^m$. To determine w, observe that the constraints

Construction	Bound on m
Conjectured	$d^2/4$
[Sau11, SPW13]	$O(d^{6/5-\varepsilon})$
[GJJ$^+$20]	$O(d^{3/2-\varepsilon})$ *
[PTVW22]	$O(d^2/\operatorname{polylog}(d))$
[KD22]	$O(d^2/\log^4(d))$
this paper	$O(d^2)$

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

Table 1: Comparison of our result with prior work.

$v_i^T \Lambda v_i = 1$ 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 η (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 $\mathcal{R} := \sum_{i=1}^n w_i v_i v_i^T$ has spectral norm bounded by 1.

These two immediately imply that Λ 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 $\|T\|_{\mathrm{op}} \leqslant O(\frac{\sqrt{m}}{d}) < 0.5$ with $m \leqslant cd^2$ (Lemma 2.7), hence A is well-conditioned. Then, using the fact that B has rank 2, we can apply the Woodbury matrix identity (Fact 2.5 and Fact 2.6) — a statement on the inverse of low-rank corrections of matrices — to conclude that M is invertible and obtain an expression for M^{-1} (Eq. (11)). This is carried out in Section 2.3.

Next, for the second goal, we need to further expand A^{-1}. Since $\|T\|_{\mathrm{op}} < 1$, 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 Section 4.1.

Finally, given the expression of M^{-1}, we are able to express $\mathcal{R}$ using the terms that show up for M^{-1}, and the bulk of our work culminates in bounding the spectral norm of $\mathcal{R}$ in Section 4.4. Bounding $\|R\|_{\mathrm{op}} \leqslant 1$ implies that $\Lambda \succeq 0$, completing the proof.

Requiring tight norm bounds Our main technical lemmas are the spectral norm bounds of T (Lemma 2.7) and the matrices in the decomposition of R at Section 4.4. Clearly, we need our norm bound $\|T\|_{\mathrm{op}} \leqslant O(\frac{\sqrt{m}}{d})$ to be tight without polylog factors so that $m \leqslant O(d^2)$ 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 $q = \operatorname{polylog}(n)$ suffices),

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

We view the summand as a closed walk $i_1 \to i_2 \to \cdots \to i_{2q} \to i_1$ on n vertices. For a random matrix, we study the expected trace $\mathbb{E}\,\mathrm{tr}(M^{2q})$. 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 M as long as the odd moments are zero. Thus, a precise upper bound on $\mathbb{E}\,\mathrm{tr}(M^{2q})$ can be obtained by carefully counting such closed walks (see [Tao12]).

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 [BHK+19], and over the years, off-the-shelf norm bounds (e.g. [AMP16]) for graph matrices have been developed and success-fully used in several works [MRX20, GJJ+20, HK22, JPR+22, BHKX22]. 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, our bounds are even tight in the constant when the matrices are explicitly written down following the graph matrix language. That said, we do not pursue the tight constant-factor dependence in this work: we believe that an analysis of our candidate matrix following the current road-map but with norm bounds tight-to-constant would still fall short of reaching the conjectured threshold of $\frac{d^2}{4}$.

In the context of a fine-grained understanding for graph matrices, Potechin and Cai [CP20, CP22] determined the limiting distribution of the spectrum of the singular values of Z-shaped and multi-Z-shaped graph matrices. However, their results are only for these specific graph matrices, and their analysis does not technically give norm bounds as they do not rule out having a negligible proportion of larger singular values.

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 simply bound the walks by bounding the number of possible encodings. Thus, all we need to do is 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 [Tao12, AMP16, JPR+22], 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 3.7). 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

4

during these steps to resolve potential (future) confusions. This is illustrated in Section 3.5.

1.2 Comparison to prior works

Our candidate matrix construction of Λ is essentially same as [KD22], while we adopt different techniques to bound the spectral norm of the non-Identity component. In particular, they use an elegant cover (or ε-net) argument which is significantly different than ours. That said, though a major obstacle being the norm bound for the invertibility, their argument suffers an additional polylog gap from the epsilon-net argument, and this is partially why we adopt the proof strategy via graph matrix decomposition that is seemingly more complicated.

Closer to our analysis is the work of [PTVW22]. They study a construction of "least-square minimization" proposed by [Sau11], 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. In this work, 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.

In the context of Planted Affine Plane problem, [GJJ$^+$20] reaches the threshold of $\widetilde{O}(d^2)$ implicitly. They adopt the framework of pseudo-calibration [BHK$^+$19] to obtain a candidate matrix, and follow a similar recipe as ours via graph matrix decompositions and spectral analysis. 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.

Remark 1.4 (A "quieter" planted distribution?). A natural idea is to analyze the planted distribution pioneered in [MRX20, GJJ$^+$20]: 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.

2 Proof of main result

Given $v_1, v_2, \ldots, v_m$ that are i.i.d. samples from $\mathcal{N}(0, \frac{1}{d}I_d)$, recall that we must construct a matrix Λ 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 2.1). 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 2.7 and Lemma 2.11) whose proofs are deferred to later sections, and complete the proof of Theorem 1.2 in Section 2.4.

2.1 Candidate construction

The following is our candidate matrix Λ, which is the same one as [KD22].

Definition 2.1 (Candidate matrix). Given $v_1, \ldots, v_m \sim \mathcal{N}(0, \frac{1}{d}I_d)$, we define the matrix $\Lambda \in \mathbb{R}^{d \times d}$ to be

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

5

where we take $w = (w_1, w_2, \ldots, w_m)$ to be the solution to the following linear system,

$$Mw = \eta$$

for $\eta \in \mathbb{R}^m$ given by

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

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

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

We first make the following simple observation.

Observation 2.2. 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). $\qquad \square$

Structure of subsequent sections For Λ 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 2.10).

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

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

and prove that $\|\mathcal{R}\|_{\text{op}}$ is bounded by 1. This is done in Section 4. Finally, combining the analyses, we finish the proof of Theorem 1.2 in Section 2.4.

2.2 Decomposition of M

The proof of Theorem 1.2 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 2.3 (Decomposition of M)**.**

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

where J_m is the all-ones matrix, $M_\alpha, M_\beta \in \mathbb{R}^{m \times m}$ are matrices with zeros on the diagonal and $M_D \in \mathbb{R}^{m \times m}$ is a diagonal matrix, defined as follows:

6

- $M_\alpha[i,j] := \sum_{a \neq b \in [d]} v_i[a] \cdot v_i[b] \cdot v_j[a] \cdot v_j[b]$ for $i \neq j \in [m]$,

- $M_\beta[i,j] := \sum_{a \in [d]} \left(v_i[a]^2 - \frac{1}{d}\right)\left(v_j[a]^2 - \frac{1}{d}\right)$ 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

$$M[i,j] = \langle v_i, v_j \rangle^2 = \left(\sum_{a \in [d]} v_i[a]v_j[a]\right)^2$$

$$= \sum_{a \neq b \in [d]} v_i[a] \cdot v_i[b] \cdot v_j[a] \cdot v_j[b] + \sum_{a \in [d]} v_i[a]^2 \cdot v_j[a]^2.$$

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

$$\sum_{a \in [d]} v_i[a]^2 \cdot v_j[a]^2 = \sum_{a \in [d]} \left(v_i[a]^2 - \frac{1}{d}\right)\left(v_j[a]^2 - \frac{1}{d}\right) + \frac{1}{d}\left(\|v_i\|_2^2 + \|v_j\|_2^2\right) - \frac{1}{d}$$

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

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

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

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

$$= \|v_i\|_2^4 = M[i,i].$$

This completes the proof. $\qquad\square$

Remark 2.4. The intention behind this decomposition is so that for $v_i \sim \mathcal{N}(0, \frac{1}{d}I_d)$, M_α, M_β, M_D are all mean 0 (while not the same variance) since $\mathbb{E}\|v_i\|_2^2 = 1$ and $\mathbb{E}\|v_i\|_2^4 = 1 + \frac{2}{d}$. Therefore, we expect $\|M_\alpha + M_\beta + M_D\|_{op}$ to be small, which implies that A is positive definite and well-conditioned. Furthermore, observe that B has rank 2:

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

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 2.5 (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 2.6 (Woodbury matrix identity [Woo50]). *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*

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

In light of Fact 2.6, 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

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

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 Section 3.6.

Lemma 2.7 (M_α, M_β, M_D are bounded). *Suppose $m \leqslant cd^2$ for a small enough constant c. With probability $1 - o_d(1)$, we have*

1. $\|M_\alpha\|_{op} \leqslant 0.1$,

2. $\|M_\beta\|_{op} \leqslant 0.1$,

3. $\|M_D\|_{op} \leqslant O(\sqrt{\frac{\log d}{d}})$.

As an immediate consequence, we get the following:

Lemma 2.8 (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 2.7 the eigenvalues of A must lie within $1 \pm 0.2 \pm \widetilde{O}(1/\sqrt{d}) \in (0.5, 1.5)$ (we assume d is large). $\qquad\square$

Next, from Fact 2.5, we can prove that M is invertible (Lemma 2.10) by showing that the 2×2 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 η, whose proof is deferred to Appendix A.

Claim 2.9. *With probability at least $1 - o_d(1)$,*

$$\|\eta\|_2^2 \leqslant (1 + o_d(1))\frac{2m}{d}.$$

Lemma 2.10 (Bounds on r, s, u; M is invertible). *Suppose $m \leqslant cd^2$ 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. $|s| \leqslant 1 + o_d(1)$,

3. $u \in [-1, -1/2]$.

Thus, we have

$$s^2 - ru \geqslant \Omega\left(\frac{m}{d}\right).$$

As a consequence, M is invertible.

Proof. By Lemma 2.8, 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 u, we have

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

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

We defer the proof for s to Claim A.2 in the appendix. With the bounds on r, s and u, we immediatley get $s^2 - ru \geqslant \Omega(\frac{m}{d})$.

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 $U = V^T = \frac{1}{\sqrt{d}}\begin{bmatrix} 1_m & \eta \end{bmatrix} \in \mathbb{R}^{m \times 2}$ and $C = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}$.

By Lemma 2.8, A is invertible. Then by Fact 2.5, 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, $s^2 - ru \geqslant \Omega(\frac{m}{d})$ suffices to conclude that M is invertible. □

2.4 Finishing the proof of Theorem 1.2

The final piece of proving Theorem 1.2 is to show that $\mathcal{R} = \sum_{i=1}^{m} w_i v_i v_i^T$ has spectral norm bounded by 1, which immediately implies that the candidate matrix $\Lambda = I_d - \mathcal{R} \succeq 0$.

Lemma 2.11 (*$\mathcal{R}$ is bounded*). *There exists some absolute constant C_R such that for $m \leqslant \frac{d^2}{C_R}$,*

$$\|\mathcal{R}\|_{\mathrm{op}} \leqslant \frac{1}{2}.$$

The proof is deferred to Section 4. In particular, we will write an expanded expression of M^{-1} and obtain a decomposition of $\mathcal{R}$ (Proposition 4.4). Then, in Section 4.2, we prove tight spectral norm bounds for matrices in the decomposition, which then completes the proof of Lemma 2.11.

Combining Lemma 2.10 and Lemma 2.11 we can finish the proof of Theorem 1.2.

Proof of Theorem 1.2. The matrix M (recall Eq. (3)) is invertible due to Lemma 2.10, thus our candidate matrix $\Lambda = I_d - \mathcal{R}$ matrix defined in Definition 2.1 is well-defined. Furthermore, by Lemma 2.11 we have that $\|\mathcal{R}\|_{\mathrm{op}} < 1$. 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 [AMP16], while not much is known in terms of precise bounds without log factors except for several specific cases (see e.g. [Tao12]).

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 [BHK$^+$19, AMP16]. We will follow the notations used in [AMP16]. 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 3.1 (Shape). A shape τ 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_τ should be distinguished from $V(\tau)$, where V_τ 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_α and M_β defined in Proposition 2.3. For these shapes, there are two vertex-types (square and circle). The two ovals in each shape indicate the left and right boundaries U_τ and V_τ.

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

(a) GOE, zero diagonal.   (b) M_α.   (c) M_β.
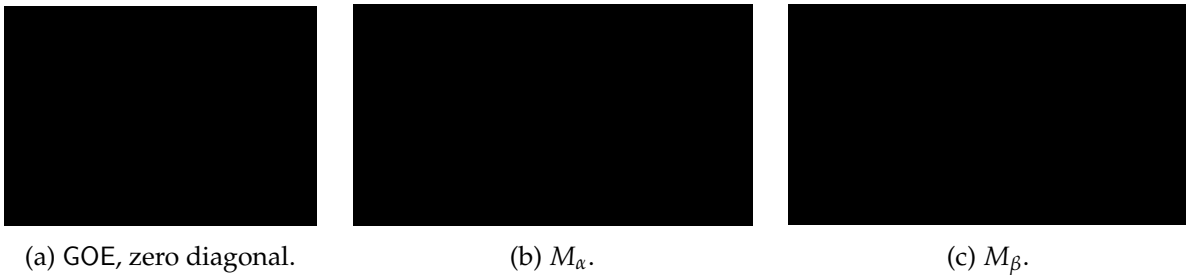
Figure 1: Graph matrix representation of a $d \times d$ GOE matrix with zero diagonal, and the $m \times m$ matrices M_α and M_β as defined in Proposition 2.3. Square vertices take labels in $[m]$ and circle vertices take labels in $[d]$. The two ovals indicate the left and right boundaries U_τ, V_τ of the shapes. If an edge e is not labeled with an index, then $t(e) = 1$ by default.

Definition 3.2 (Mapping of a shape). Given a shape τ, we call a function $\sigma : V(\tau) \to \mathbb{N}$ a mapping of the shape if

1. σ assigns a label for each vertex according to its specified vertex-type;

2. σ is an injective mapping for vertices of the same type.

Definition 3.3 (Graph matrix for shape). Given a shape τ, we define its graphical matrix M_τ to be the matrix indexed by all possible boundary labelings of S, T, and for each of its entry, we define

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

Observe that for each entry $M_\tau[S, T]$, since σ must map U_τ and V_τ 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 $i \neq j \in [m]$,

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

Note also that since σ 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 3.3, 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 $v_1, \ldots, v_m \sim \mathcal{N}(0, \frac{1}{d}I_d)$.

- 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 3.4. For our technical analysis, we may also employ this machinery on a broader range of graph matrices for shape in which we relax the local injectivity condition within each block. That said, for illustration purpose, it suffices to consider the vanilla setting of graph matrix.

Definition 3.5 (D_V size constraint). For each graph matrix τ considered in this work, let D_V be the size constraint such that $|V(\tau)| \leqslant D_V$.

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_τ with underlying random matrix G and any $q \in \mathbb{N}$,

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

where the expectation is taken over G.

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

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

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

Remark 3.6 (Labeled vertex/edge). 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 3.7 (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 τ, we call $\mathcal{L}_\tau : E(\tau) \to \{F, R, S, H\}$ a *step-labeling* of the block. The subscript τ is ignored when it is clear.

We note that the terms "fresh", "high-mul" and "return" are adopted from the GOE matrix analysis in [Tao12]. 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 $\mathbb{E}[\chi_{t(e)}(G[e])^{\mathrm{mul}(e)}]$ 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 [Tao12]), 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_α), 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 τ. 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

$$\mathsf{vtxcost} \cdot \mathsf{edgeval} \leqslant B_q(\tau)$$

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

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

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

$$\mathsf{vtxcost}(\mathsf{BlockStep}_i) \cdot \mathsf{edgeval}(\mathsf{BlockStep}_i) \leqslant B_q(\tau).$$

We point out that the block-value function B should be considered as a function of both the shape τ 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 $\mathrm{poly}(d)$ in our case, is the factor that comes up in the start of the walk to specify

13

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 3.8, the norm bound follows immediately from Markov's inequality.

Proposition 3.9. *Let M_τ be a graph matrix with dimension $\mathrm{poly}(d)$, and let $q \geqslant \Omega(\log^2 d)$. Suppose $B_q(\tau)$ is a valid block-value function, Then, with probability $1 - 2^{-q/\log d}$,*

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

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

$$\mathbf{Pr}\left[\|M_\tau\|_{\mathrm{op}} > (1+\varepsilon)B_q(\tau)\right] \leqslant \mathbf{Pr}\left[\mathrm{tr}\left((M_\tau M_\tau^T)^q\right) > (1+\varepsilon)^{2q}B_q(\tau)^{2q}\right]$$

$$\leqslant (1+\varepsilon)^{-2q}\mathrm{poly}(d) \leqslant e^{-2\varepsilon q}\mathrm{poly}(d)$$

since $\mathbb{E}[\mathrm{tr}((M_\tau M_\tau^T)^q)] \leqslant \mathrm{poly}(d) \cdot B_q(\tau)^{2q}$ by Definition 3.8. Setting $\varepsilon = \frac{1}{\log d}$ and $q \geqslant \Omega(\log^2 d)$, we have that the probability is at most $2^{-q/\log d}$. Thus, we can conclude that $\|M_\tau\|_{\mathrm{op}} \leqslant (1 + o_d(1)) \cdot B_q(\tau)$ with probability $1 - 2^{-q/\log d}$. $\square$

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

Proposition 3.10. *For any graph matrix M_τ and any valid factor assignment scheme,*

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

is a valid block-value function for τ.

Proof. It is clear that the second requirement in Definition 3.8 is satisfied. For the first requirement, observe that the trace can be bounded by the matrix dimension (specifying the start of the walk) times

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

$\square$

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 3.6). 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 Pur, to specify the destination of any R step. We will defer the specific details of 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 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 3.10; in fact, the norm bound for M_τ 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 3.11 (Vertex appearance in block-step)**.** Each labeled vertex appearance can be "first", "middle" and "last". Moreover, each vertex on the block-step boundary (U_τ or V_τ) 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 at via an R step, its cost is captured

by 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 3.11. 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,

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

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

For now, we consider matrices that either contain only h_1 or only h_2 edges (the edge factors for graph matrices with "mixed" edges will be handled in Section 4.3.1). 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 (alternatively, 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 (alternatively, 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 3.13. *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 step (including S), one R step and $k - 2$ H steps. Moreover, since k is even and $2q$ is the length of the walk, we have $4 \leqslant k \leqslant 2q$.

- If it is an h_1 edge, then the scheme assigns a factor $\frac{1}{d} \cdot (\frac{2q}{\sqrt{d}})^{k-2} \geqslant d^{-k/2}(2q)^{k/2} \geqslant (\frac{k}{d})^{k/2}$. By [Fact 3.12](#), 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 $\frac{2}{d^2} \cdot (\frac{8q^2}{d})^{k-2} \geqslant d^{-k}2^{k/2}(2q)^k \geqslant 2^{k/2}(\frac{k}{d})^k$. By [Fact 3.12](#), 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. $\qquad\square$

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 Pur, to specify the destination of any return (R) step. Note that the term "unforced return" is adopted from [Tao12] as well. In this section, we complete the bound of vertex factors by bounding the 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 unclosed 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 3.14 (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 3.15 (Potential-unforced-return factor Pur). For any time t and vertex v, let $\mathrm{Pur}_t(v)$ 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 3.16 (Bounding Pur_t for circle vertices). *For any time t, suppose the walker is currently at a circle vertex v, then*

$$\mathrm{Pur}_t(v) \leqslant \#(\text{R steps closed from } v) + \#(\text{unclosed edges incident to } v \text{ at time } t) - 1$$
$$\leqslant 2 \cdot s_t(v) + h_t(v),$$

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 $P_t(v) := \#(\text{R steps closed from } v) + \#(\text{unclosed edges incident to } v \text{ at time } t) - 1$ 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:

18

- Arriving by an R step: the edge is now closed, but the R step was not from v. So $P_t(v) = P_{t'+1}(v) - 1 \leqslant P_{t'}(v)$, while $s_t(v) = s_{t'}(v)$ and $h_t(v) = h_{t'}(v)$.

- Arriving by an S step: the edge is new, so $P_t(v) = P_{t'+1}(v) + 1 \leqslant P_{t'}(v) + 2$, and we have $s_t(v) = s_{t'}(v) + 1$.

- Arriving by an H step: $P_t(v) = P_{t'+1}(v) \leqslant P_{t'}(v) + 1$, and $h_t(v) = h_{t'}(v) + 1$.

In all three cases, assuming $P_{t'}(v) \leqslant 2 \cdot s_{t'}(v) + h_{t'}(v)$, we have $P_t(v) \leqslant 2 \cdot s_t(v) + h_t(v)$, completing the induction. □

3.5.2 Pur **bound for square vertices**

The argument of Lemma 3.16 does not apply well for vertices incident to multiple edges in a single step. In particular, this may happen for square vertices in M_α as each is arrived via 2 edges and each pushes out 2 edges (recall Figure 1). This is not an issue for M_β, but we will treat square vertices in M_β the same way to unify the analysis; in the context of Pur for square vertices, one may think of M_β as collapsing the two circle vertices in M_α.

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 3.17 (Bounding Pur_t for square vertices). *For any time t, suppose the walker is currently at a square vertex v, then*

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

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 $P_{t'+1}(v) \leqslant P_{t'}(v) + 2$.

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 $P_t(v) = P_{t'+1} - 2 \leqslant P_{t'}(v)$, 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, $P_t(v) = P_{t'+1}(v) \leqslant P_{t'}(v) + 2$ and $s_t(v) + h_t(v) = s_{t'}(v) + h_{t'}(v) + 1$.

- Arriving by two S/H steps: in this case, $P_t(v) = P_{t'+1}(v) + 2 \leqslant P_{t'}(v) + 4$, whereas $s_t(v) + h_t(v) = s_{t'}(v) + h_{t'}(v) + 2$.

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

Corollary 3.18. *For each surprise/high-mul step, it suffices for us to assign* 2 Pur *factors, which is a cost of $(2q \cdot D_V)^2$ so that each Pur factor throughout the walk is assigned. Moreover, for M_α, we pay a cost of 2 for any R step leaving a square vertex so that we can presume 2 edges being forced in Lemma 3.17.*

3.6 Wrapping up with examples

Recall Proposition 3.10 that for a graph matrix of shape τ,

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

is a valid block-value function for τ (Definition 3.8). Moreover, by Proposition 3.9, we can take $q \geqslant d^\varepsilon$ and conclude that with probability $1 - 2^{-d^\varepsilon}$,

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

For each given shape, it suffices for us to bound the block-value for each edge-labeling. And we demonstrate how this may be readily done given the above bounds.

3.6.1 Warm-up: tight bound for GOE

As a warm-up, we first see how the above framework allows us to readily deduce a tight norm bound for $G \sim \mathrm{GOE}(0, \frac{1}{d})$, 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 $2 + o_d(1)$ [Tao12]. Figure 1a shows the shape τ 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 3.16, 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_τ must be making a middle appearance; it is not first due to Definition 3.11, and it is not last as otherwise the edge appears only once throughout the walk. The vertex in V_τ is making a first appearance, so it gets a factor of $\sqrt{d}$;

- R: the vertex in V_τ 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_τ may be making its last appearance, so it gets a factor of $\sqrt{d}$;

- S: the vertex in U_τ is making a middle appearance (same as F), and the vertex in V_τ 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):

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

since q and D_V are both polylog(d). Therefore, by Proposition 3.9, we can conclude that $\|G\|_{\mathrm{op}} \leqslant 2 + o_d(1)$ with high probability, which is the correct bound.

3.6.2 Bound for M_β

We now prove a bound on $\|M_\beta\|_{op}$. Figure 1c shows the associated shape.

Lemma 3.19 (Norm bound for M_β). *Let $m \geqslant \omega(d)$ and $q = \Omega(\log^2 d)$. Then with probability $1 - 2^{-q/\log d}$,*

$$\|M_\beta\|_{op} \leqslant (1 + o_d(1))\frac{2m}{d^2} \, .$$

Proof. Let β be the shape associated with the matrix M_β. We bound $B_q(\beta)$ via our vertex/edge factor assignment schemes combined with Pur factors. Recall that each square vertex has weight m and each circle vertex has weight d. We case on the step-labels of the two edges,

1. $F \to F$: we have an F edge leading to a square vertex and circle vertex each. The first square vertex must be making a middle appearance (Definition 3.11), while the circle and the other square vertex make first appearances, giving a vertex factor of $\sqrt{m} \cdot \sqrt{d}$. Furthermore, there is no Pur factor incurred. Finally, the edge factor is $\frac{2}{d^2}$.

2. $R \to R$: we have both R edges departing from a square and circle vertex. There is no vertex making first appearance and the square vertex on the right must be making middle appearance. The other two vertices may be making last appearances. Furthermore, there is no Pur factor incurred, while we assume each R edge from a square vertex can be identified at a cost of $[2]$ modulo the ones with assigned Pur factor, giving a total vertex factor of $2\sqrt{m} \cdot \sqrt{d}$. Finally, the edge factor is $\frac{2}{d^2}$.

3. $R \to F$: the circle vertex must be making a middle appearance, since the F edge must be closed later. The square vertex on the left may be making a last appearance, and the square vertex on the right must be making a first appearance. This gives a vertex factor of $\sqrt{m} \cdot \sqrt{m} = m$. There is no Pur factor, and the edge factor is $\frac{2}{d^2}$.

4. $F \to R$: this cannot happen. The F step means that the circle vertex is making its first appearance, but the R step means that it must have appeared before.

5. For any other step-labelings involving S and H step-labels, both vertices of the S/H edge must be making middle appearances. Thus, the vertex factor is at most $\sqrt{m}$. By Lemmas 3.16 and 3.17, the Pur factor is at most $(2q)^4$. Finally, the edge factor is at most $O(\frac{q^4}{d^2})$.

Summing over all possible step-labelings, by Proposition 3.10 we get

$$B_q(\beta) \leqslant \sqrt{md} \cdot \frac{2}{d^2} + 2\sqrt{md} \cdot \frac{2}{d^2} + m \cdot \frac{2}{d^2} + \frac{q^{O(1)}}{d^2} \leqslant \frac{2m}{d^2}(1 + o_d(1)) \, ,$$

provided that $m \gg d$. Therefore, by Proposition 3.9, we have that with probability $1 - 2^{-q/\log d}$, $\|M_\beta\|_{op} \leqslant (1 + o_d(1))\frac{2m}{d^2}$. $\qquad\square$

3.6.3 Bound for M_α

We now prove a bound on $\|M_\alpha\|_{op}$. Figure 1b shows the associated shape.

Lemma 3.20 (Norm bound for M_α). *Let $m \geqslant \omega(d)$ and $q = \Omega(\log^2 d)$. Then with probability $1 - 2^{-q/\log d}$,*

$$\|M_\alpha\|_{\mathrm{op}} \leqslant (1 + o_d(1)) \cdot \frac{1}{d^2}\left(3d\sqrt{m} + 2m\right).$$

Proof. Let α be the shape associated with the matrix M_α. Similar to the proof of Lemma 3.19, we bound $B_q(\alpha)$ by casing on the step-labelings. There are two paths from U_α to V_α, 4 edges in total. t

1. In the case of all F or all R, one of the square vertices must be making middle appearance, hence we get a vertex factor of $\sqrt{m} \cdot (\sqrt{d})^2 = d\sqrt{m}$. There is no Pur factor, and the edge factor is $(\frac{1}{\sqrt{d}})^4 = \frac{1}{d^2}$. For the case of all R, by Corollary 3.18 we pick up an additional factor of $[2]$ since we assume each R edge from a square vertex can be identified at a cost $[2]$ modulo those with assigned Pur factor.

2. If both paths are $R \to F$, then both circle vertices are making middle appearances, hence we get a vertex factor of $\sqrt{m} \cdot \sqrt{m} = m$. There is no Pur factor while we pick up a factor $[2]$ for the return from the square vertex. Finally, the edge factor is $(\frac{1}{\sqrt{d}})^4 = \frac{1}{d^2}$.

3. Analogous to M_β, an $F \to R$ path cannot happen.

4. For any other step-labelings involving S and H step-labels, there must be at least one square and one circle vertex making middle appearances, so the vertex factor is at most $\sqrt{m} \cdot \sqrt{d}$. The Pur factor is $(2q \cdot D_V)^{O(1)} = \mathrm{polylog}(d)$, and the edge factor is $(\frac{2q}{\sqrt{d}})^4$.

Summing over all possible step-labelings, by Proposition 3.10 we get

$$B_q(\alpha) \leqslant d\sqrt{m} \cdot \frac{1}{d^2} + 2d\sqrt{m} \cdot \frac{1}{d^2} + 2m \cdot \frac{1}{d^2} + \sqrt{md} \cdot \frac{q^{O(1)}}{d^2} \leqslant (1 + o_d(1)) \cdot \frac{1}{d^2}\left(3d\sqrt{m} + 2m\right).$$

Therefore, by Proposition 3.9, we have that with probability $1 - 2^{-q/\log d}$, $\|M_\alpha\|_{\mathrm{op}} \leqslant (1 + o_d(1)) \cdot \frac{1}{d^2}(3d\sqrt{m} + 2m)$. $\qquad\square$

4 Matrix decomposition of $\mathcal{R}$

In this section, we work towards analyzing $\mathcal{R}$ and proving Lemma 2.11. Recall our candidate matrix $\Lambda = I_d - \mathcal{R} \in \mathbb{R}^{d \times d}$ from Definition 2.1, where

$$\mathcal{R} := \sum_{i=1}^m w_i v_i v_i^T = \sum_{i=1}^m \left(M^{-1}\eta\right)[i] \cdot v_i v_i^T.$$

See Eq. (2) and (3) for a reminder of the definitions of $M \in \mathbb{R}^{m \times m}$ and $\eta \in \mathbb{R}^m$.

To analyze $\mathcal{R}$, we begin by obtaining an explicit expression for M^{-1} using the Woodbury matrix identity (Fact 2.6), as discussed in Section 2.3. Recall that we write $M = A + UCU^T$ where A is positive definite with high probability by Lemma 2.8, and $U = \frac{1}{\sqrt{d}}\begin{bmatrix} 1_m & \eta \end{bmatrix} \in \mathbb{R}^{m \times 2}$ and $C = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}$. Restating Eq. (7), the scalars $r, s, u \in \mathbb{R}$ are defined as follows,

$$\begin{bmatrix} r & s \\ s & u \end{bmatrix} := C^{-1} + U^T A^{-1} U = \begin{bmatrix} \frac{1_m^T A^{-1} 1_m}{d} & 1 + \frac{\eta^T A^{-1} 1_m}{d} \\ 1 + \frac{\eta^T A^{-1} 1_m}{d} & -1 + \frac{\eta^T A^{-1} \eta}{d} \end{bmatrix}. \tag{10}$$

Our next step is to show the following expansion of $M^{-1}\eta$.

22

Proposition 4.1 (Expansion of $M^{-1}\eta$). *Let $r, s, u \in \mathbb{R}$ be defined as in Eq. (10). Then,*

$$M^{-1}\eta = \frac{r+s}{s^2 - ru} \cdot A^{-1}\eta - \frac{u+s}{s^2 - ru} \cdot A^{-1}1_m . \tag{11}$$

Proof. The inverse of Eq. (10) is as follows,

$$\left(C^{-1} + U^T A^{-1} U\right)^{-1} = \begin{bmatrix} r & s \\ s & u \end{bmatrix}^{-1} = \frac{1}{ru - s^2} \begin{bmatrix} u & -s \\ -s & r \end{bmatrix} .$$

Then, applying the Woodbury matrix identity (Fact 2.6), we have

$$M^{-1} = A^{-1} - \frac{1}{ru - s^2} A^{-1} U \begin{bmatrix} u & -s \\ -s & r \end{bmatrix} U^T A^{-1}$$

$$= A^{-1} + \frac{1}{s^2 - ru} A^{-1} \left(u \cdot \frac{1_m 1_m^T}{d} - s \cdot \frac{\eta 1_m^T + 1_m \eta^T}{d} + r \cdot \frac{\eta \eta^T}{d} \right) A^{-1} .$$

Next, using the above, we have

$$M^{-1}\eta = A^{-1}\eta + \frac{1}{s^2 - ru} \left(\left(u \cdot \frac{1_m^T A^{-1}\eta}{d} - s \cdot \frac{\eta^T A^{-1}\eta}{d} \right) \cdot A^{-1}1_m \right.$$

$$\left. + \left(-s \cdot \frac{1_m^T A^{-1}\eta}{d} + r \cdot \frac{\eta^T A^{-1}\eta}{d} \right) \cdot A^{-1}\eta \right) .$$

Plugging in the definition of r, s, u in Eq. (10), we get

$$M^{-1}\eta = A^{-1}\eta + \frac{1}{s^2 - ru}(u(s-1) - s(u+1))A^{-1}1_m + \frac{1}{s^2 - ru}(-s(s-1) + r(u+1))A^{-1}\eta$$

$$= A^{-1}\eta - \frac{u+s}{s^2 - ru} \cdot A^{-1}1_m + \frac{-s^2 + ru + r + s}{s^2 - ru} \cdot A^{-1}\eta$$

$$= \frac{r+s}{s^2 - ru} \cdot A^{-1}\eta - \frac{u+s}{s^2 - ru} \cdot A^{-1}1_m ,$$

finishing the proof. $\qquad\square$

4.1 Inverse of A: Neumann series and truncation

In light of Proposition 4.1, we proceed to analyze A^{-1}. Recall from Proposition 2.3 that $A = I_m + M_\alpha + M_\beta + M_D + \frac{1}{d}I_m$. A useful tool to obtain inverses is to apply Neumann series (a.k.a. matrix Taylor expansion of $\frac{1}{1-x}$), which allows us to write

$$(I - T)^{-1} = \sum_{k=0}^{\infty} T^k$$

for $\|T\|_{\text{op}} < 1$. In our case, let $T := -(M_\alpha + M_\beta + M_D + \frac{1}{d}I_m)$, then $\|T\|_{\text{op}} < 1$ is guaranteed by Lemma 2.7. Thus, we can write

$$A^{-1} = \sum_{k=0}^{\infty} T^k = \sum_{k=0}^{\infty} (-1)^k \sum_{(Q_1, \dots, Q_k) \in \{M_\alpha, M_\beta, M_D, \frac{1}{d}I_m\}^k} Q_1 Q_2 \cdots Q_k .$$

With the norm bounds from Lemma 2.7, we can truncate the series, i.e., capping the number of occurrences of M_α, M_β, M_D and $\frac{1}{d}I_m$ by certain thresholds, such that the error is small.

Definition 4.2 (Truncated A^{-1}). We define thresholds $\tau_1 = \tau_2 = O(\log d)$, $\tau_3 = 3$, and $\tau_4 = 1$, and define the truncation of A^{-1} as

$$T_0 := \sum_{\substack{k_1,k_2,k_3,k_4 \in \mathbb{N} \\ k_i \leqslant \tau_i, \ \forall i}} (-1)^{k_1 + \cdots + k_4} \sum_{\substack{(Q_1,\dots,Q_k) \in \{M_\alpha, M_\beta, M_D, \frac{1}{d}I_m\}^k \\ i\text{-th matrix occurring } k_i \text{ times}}} Q_1 Q_2 \cdots Q_k. \tag{12}$$

In the next lemma, we upper bound the truncation error.

Lemma 4.3 (Truncation error of A^{-1}). *Suppose $m \leqslant cd^2$ for a small enough constant c. Let T_0 be the truncated series defined in Definition 4.2 with thresholds $\tau_1 = \tau_2 = O(\log d)$, $\tau_3 = 3$ and $\tau_4 = 1$. Then, with probability $1 - o_d(1)$, the truncation error $E = A^{-1} - T_0$ satisfies $\|E\|_{\mathrm{op}} \leqslant O(\frac{\log d}{d})^2$.*

Proof. From Lemma 2.7, we know that $\|M_\alpha\|_{\mathrm{op}}, \|M_\beta\|_{\mathrm{op}} \leqslant 0.1$, $\|M_D\|_{\mathrm{op}} \leqslant O(\sqrt{\log d/d})$, and $\|\frac{1}{d}I_m\|_{\mathrm{op}} = \frac{1}{d}$ with high probability. We can bound the contribution of $\|M_\alpha\|_{\mathrm{op}}$ in the truncation error by

$$\sum_{i=\tau_1+1}^{\infty} \sum_{j=0}^{\infty} \binom{i+j}{i} \cdot \|M_\alpha\|_{\mathrm{op}}^i \cdot \left(\|M_\beta\|_{\mathrm{op}} + \|M_D\|_{\mathrm{op}} + \frac{1}{d} \right)^j$$

$$\leqslant \sum_{i=\tau_1+1}^{\infty} \sum_{j=0}^{\infty} 2^{i+j} \cdot \|M_\alpha\|_{\mathrm{op}}^i \cdot (0.1 + o(1))^j$$

$$\leqslant \left(2\|M_\alpha\|_{\mathrm{op}} \right)^{\tau_1+1} \cdot O(1).$$

Similarly for M_β, M_D and $\frac{1}{d}I_m$. Therefore, we can bound the total truncation error:

$$\|E\|_{\mathrm{op}} \leqslant \left(\left(2\|M_\alpha\|_{\mathrm{op}}\right)^{\tau_1+1} + \left(2\|M_\beta\|_{\mathrm{op}}\right)^{\tau_2+1} + \left(2\|M_D\|_{\mathrm{op}}\right)^{\tau_3+1} + (2/d)^{\tau_4+1} \right) \cdot O(1)$$

$$\leqslant O\left(\frac{\log d}{d} \right)^2$$

by our choice of thresholds $\tau_1, \tau_2, \tau_3, \tau_4$. $\qquad\square$

4.2 Decomposition of $\mathcal{R}$ via truncated A^{-1}

We shift our attention back to $\mathcal{R} = \sum_{i=1}^{m} w_i v_i v_i^T$. Using the expansion of $M^{-1}\eta$ in Proposition 4.1 (Eq. (11)) and the truncation of A^{-1}, we decompose $\mathcal{R}$ as follows.

Proposition 4.4 (Decomposition of $\mathcal{R}$). *Suppose $m \leqslant cd^2$ for a small enough constant c. Let T_0 be the truncated series of A^{-1} defined in Definition 4.2. Then,*

$$\mathcal{R} = \mathcal{R}_1 + \mathcal{R}_2 + E_{\mathcal{R}},$$

where

$$\mathcal{R}_1 := \frac{r+s}{s^2 - ru} \sum_{i \in [m]} (T_0 \eta)[i] \cdot v_i v_i^T,$$

$$\mathcal{R}_2 := \frac{-u-s}{s^2 - ru} \sum_{i \in [m]} (T_0 1_m)[i] \cdot v_i v_i^T,$$

and $\|E_{\mathcal{R}}\|_{\mathrm{op}} \leqslant o(1)$ with probability $1 - o_d(1)$.

We first state the following claim which is needed for the error analysis.

Claim 4.5. *With probability* $1 - e^{-d}$, $\left\| \sum_{i=1}^m v_i v_i^T \right\|_{\mathrm{op}} \leqslant (1 + o_d(1)) \frac{m}{d}$.

Proof. We can write $\sum_{i=1}^m v_i v_i^T = VV^T$ where $V \in R^{d \times m}$ has i.i.d. $\mathcal{N}(0, \frac{1}{d})$ entries, and note that $\left\| \sum_{i=1}^m v_i v_i^T \right\|_{\mathrm{op}} = \sigma_{\max}(V)^2$. Since we assume $m \geqslant \omega(d)$, by standard concentration of the largest singular value of rectangular Gaussian matrices [Tro15], we have that $\sigma_{\max}(V) \leqslant (1 + o_d(1)) \sqrt{\frac{m}{d}}$ with probability at least $1 - e^{-d}$. $\qquad\square$

Proof of Proposition 4.4. Recall that $\mathcal{R} = \sum_{i \in [m]} (M^{-1}\eta)[i] \cdot v_i v_i^T$. Unpacking the expression of $M^{-1}\eta$ in Proposition 4.1 and plugging in $A^{-1} = T_0 + E$, we have $\mathcal{R} = \mathcal{R}_1 + \mathcal{R}_2 + E_{\mathcal{R}}$ where

$$E_{\mathcal{R}} = \sum_{i \in [m]} \delta_i v_i v_i^T, \quad \delta_i := \frac{r+s}{s^2 - ru} \langle E[i], \eta \rangle - \frac{u+s}{s^2 - ru} \langle E[i], 1_m \rangle.$$

We first upper bound $|\delta_i|$ for all $i \in [m]$. First, observe that $|\langle E[i], \eta \rangle| \leqslant \|E\|_{\mathrm{op}} \cdot \|\eta\|_2$ and $|\langle E[i], 1_m \rangle| \leqslant \|E\|_{\mathrm{op}} \cdot \|1_m\|_2$. By Claim 2.9 and Lemma 4.3, we have $\|E\|_{\mathrm{op}} \leqslant O(\frac{\log d}{d})^2$ and $\|\eta\|_2 \leqslant O(\sqrt{\frac{m}{d}})$ with high probability. Moreover, by Lemma 2.10, we have that $r = \Theta(\frac{m}{d})$, $|s| \leqslant O(\sqrt{d})$ and $-1 \leqslant u \leqslant -1/2$, hence $s^2 - ru \geqslant \Omega(\frac{m}{d})$. Thus,

$$|\delta_i| \leqslant O\left(\frac{\log d}{d} \right)^2 \cdot \left(\sqrt{\frac{m}{d}} + \frac{d^{3/2}}{\sqrt{m}} \right) \leqslant O\left(\frac{\log^2 d}{\sqrt{md}} \right)$$

as we assume that $m \leqslant O(d^2)$.

Finally, Claim 4.5 states that $\|\sum_{i=1}^m v_i v_i^T\|_{\mathrm{op}} \leqslant O(\frac{m}{d})$. Since

$$-\sum_{i \in [m]} |\delta_i| v_i v_i^T \preceq E_{\mathcal{R}} \preceq \sum_{i \in [m]} |\delta_i| v_i v_i^T,$$

we may conclude that $\|E_{\mathcal{R}}\|_{\mathrm{op}} \leqslant O(\frac{\log^2 d}{\sqrt{md}}) \cdot O(\frac{m}{d}) \leqslant O(\frac{\log^2 d}{\sqrt{d}})$ as $m \leqslant O(d^2)$. $\qquad\square$

4.3 Overview: each term is a dangling path of injective gadgets

By writing A^{-1} as a truncated series (Definition 4.2) and the decomposition of $\mathcal{R}$ (Proposition 4.4), we may view $\mathcal{R}_1$ and $\mathcal{R}_2$ as linear combinations of $d \times d$ matrices of the forms

$$\sum_{i \in [m]} (Q_1 Q_2 \cdots Q_k \eta)[i] \cdot v_i v_i^T \quad \text{and} \quad \sum_{i \in [m]} (Q_1 Q_2 \cdots Q_k 1_m)[i] \cdot v_i v_i^T \tag{13}$$

respectively, where $Q_1, \ldots, Q_k \in \{M_\alpha, M_\beta, M_D, \frac{1}{d} I_m\}$ are $m \times m$ matrices.

Using the machinery of graph matrices described in Section 3.1, we can systematically represent these matrices as shapes (with underlying input $\{v_1, \ldots, v_m\} \subseteq \mathbb{R}^d$). For starters, the off-diagonal part of the matrix $\sum_{i=1}^m v_i v_i^T$ is represented by the shape in Figure 2a; it serves as the "base" for other matrices in Eq. (13). Each matrix in Eq. (13) can be represented by attaching "gadgets" (Figure 2b) to the square vertex in the middle. For the case of $\mathcal{R}_1$, since $\eta_j = \|v_j\|_2^2 - 1 = \sum_{a=1}^d h_2(v_j[a])$ for $j \in [m]$ (see Eq. (2)), each shape has an extra h_2 attached at the end. Figure 2c shows an example of such a shape.

In the following, we formally define such shapes which we call *dangling* shapes.
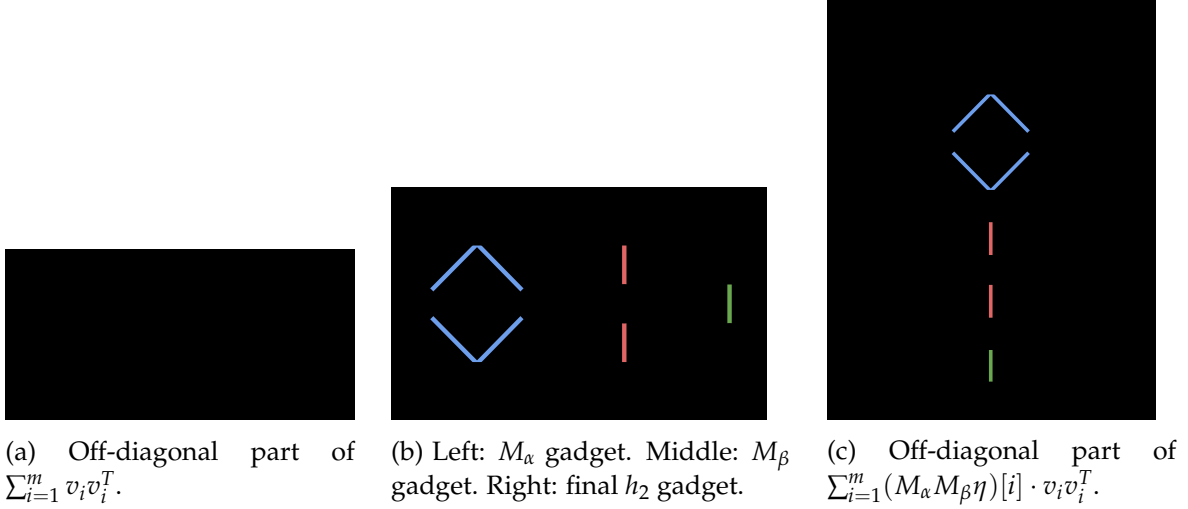
25

(a) Off-diagonal part of $\sum_{i=1}^m v_i v_i^T$.

(b) Left: M_α gadget. Middle: M_β gadget. Right: final h_2 gadget.

(c) Off-diagonal part of $\sum_{i=1}^m (M_\alpha M_\beta \eta)[i] \cdot v_i v_i^T$.

Figure 2: Examples of graph matrices in the decomposition of $\mathcal{R}$. Recall that square vertices take labels in $[m]$ and circle vertices take labels in $[d]$. Unlabeled edges have Hermite index 1.

Definition 4.6 (Injective gadgets). We call each shape in $\{M_\alpha, M_\beta, M_D, \frac{1}{d} I_m\}$ an *injective gadget* in the dangling shape.

Definition 4.7 (Dangling shapes). We further separate the matrices in the decomposition of $\mathcal{R}$ (Eq. (13)) into diagonal and off-diagonal terms:

1. For the off-diagonal terms, we start with a path from U to V (each containing a circle vertex receiving labels in $[d]$) passing through a middle square vertex that receives a label in $[m]$.

2. For diagonal terms, we have a double edge (that shall be distinguished from an h_2 edge as the double edge here stands for $v_i[a]^2$) connected to a middle square vertex that receives a label in $[m]$.

For each term, we specify a length-$k \geqslant 0$ dangling path that starts from the middle square vertex such that

- for any $k > 0$, each step comes from one of the following gadgets in $\{M_\alpha, M_\beta, M_D, \frac{1}{d} I_m\}$;

- The dangling path is not necessarily injective, that we may have each vertex appearing at multiple locations along the path. However, since it is a walk along the above gadgets, the path is *locally injective* within each gadget.

For matrices from $\mathcal{R}_1$, there is an additional η factor. Thus, we attach an h_2 gadget (Figure 2b) to the end of the dangling path. We call this the "final h_2 gadget".

Definition 4.8 (Gadget incursion). Let A, B be two gadgets along the dangling path. We call it a gadget-incursion if there are unnecessary vertex intersections between $V(A)$ and $V(B)$ beyond the "necessary intersection": when A, B are adjacent, $V_A = U_B$ is the necessary boundary intersection, any intersection in $V(A) \setminus V_A$ and $V(B) \setminus U_B$ is a gadget incursion; similarly, when A, B are not adjacent, any intersection in $V(A)$ and $V(B)$ is a gadget incursion.

26

Vertex and edge appearance   Recall from Section 3.2 that we bound the norm of a graph matrix by analyzing length-$2q$ "block-walks" of the shape and bounding the vertex/edge factor of each "block-step". To this end, we need to consider both *global* and *local* appearances of a labeled vertex. We remind the reader that a labeled square (resp. circle) vertex is an element in $[m]$ (resp. $[d]$), and a labeled edge is an element in $[m] \times [d]$ (see Remark 3.6).

Definition 4.9 (Local versus global appearance). Given a block-walk, we call each labeled vertex's appearance within the given block a local appearance, and each vertex's appearance throughout the walk a global appearance.

   Moreover, we say that a labeled vertex/edge is making its *global* first/last appearance if it is the first/last appearance of that labeled vertex throughout the walk. Similarly, we say it is making its *local* first/last appearance if it is the first/last appearance within the given block in the walk.

   We also need the following definition, which is a special case that we need to handle. The term "reverse-charging" will be clear once we describe our edge charging scheme in Section 4.4.

Definition 4.10 (Reverse-charging step/edge). For a given walk, and a block in the walk, we call a step $u \to v$ reverse-charging if

1. the underlying edge is making its last *global* appearance throughout the walk;

2. the underlying edge's first *global* appearance is also in the current block;

3. (Reverse) the first appearance of the underlying edge goes from v to u.

4.3.1   Further set-up for step-labeling and edge factor scheme

Our argument for tight matrix norm bounds requires assigning each edge (or step) a step-label (Definition 3.7) that represents whether it is making first/middle/last appearance, and assigning edge factors based on the edge type (recall our edge factor scheme in Section 3.4). However, further care is warranted for dangling shapes when an edge appears with different Hermite indices in the walk (e.g., appears both as an h_1 and h_2 edge). In this case, it is no longer true that an h_k edge needs to appear at least twice in the walk for the random variable to be non-vanishing. For example, suppose an edge appears as h_1, h_1, and h_2 in the walk. Then, even though h_2 only appears once, this term is non-vanishing under expectation:

$$\mathbb{E}_{x \sim \mathcal{N}(0,1/d)} \left[h_1(x)^2 h_2(x) \right] = \mathbb{E} \left[x^2 (x^2 - \frac{1}{d}) \right] = \mathbb{E} \left[x^4 \right] - \mathbb{E} \left[\frac{x^2}{d} \right] = \frac{2}{d^2} . \tag{14}$$

   The matrices that arise in our analysis may contain h_1, h_2, h_3, h_4 edges. For $i \leqslant 4$, we treat an h_i edge as i edge-copies in our edge factor assignment scheme.

Definition 4.11 (Step-labeling scheme for mixed edges). For each step regardless of the Hermite index, assign a step-label to all its edge-copies as follows,

1. Assign an F step if it is making its first appearance;

2. Assign an H step if it is making its middle appearance

3. Assign an R step if it is making its last appearance.

We next describe our edge factor assignment scheme.

Lemma 4.12 (Edge factor assignment scheme). *For any graph matrix of size at most D_V that contains h_1, h_2, h_3, h_4 edges and walks of length $2q \geqslant \Omega(\log d)$, we can assign values to each edge-copy among the edge's appearance throughout the walk such that*

1. *Each edge-copy of an h_1, h_2 edge with F / R step-label gets assigned a value $\frac{2^{1/4}}{\sqrt{d}}$.*

2. *Each edge-copy of an edge with step-label H and any edge-copy of an h_3, h_4 edge gets assigned a value $\frac{32qD_V}{\sqrt{d}}$.*

3. *In the case of a random variable appearing as h_1, h_1, h_2 (in an arbitrary order), it gets assigned value of $\frac{2}{d^2}$ in total, in particular, each edge-copy of the h_2 edge gets assigned a value $\frac{\sqrt{2}}{\sqrt{d}}$.*

Proof. We first note that $\mathbb{E}_{x \sim \mathcal{N}(0,1/d)}[h_k(x)^2] = \frac{k!}{d^k}$. For $k = 1, 2$, if an h_k edge only appears twice and no other Hermite index occurs, then it must have $2k$ edge-copies with step-labels F and R, giving an edge factor of $(\frac{2^{1/4}}{\sqrt{d}})^{2k} = \frac{2^{k/2}}{d^k}$, which is larger than $\frac{k!}{d^k}$ for $k = 1, 2$.

Next, we consider the case when h_3, h_4 edges are involved or when an edge appears more than twice, i.e., some edge-copies are assigned step-label H. Let a_k be the number of times h_k appears in the walk, and let $t := \sum_{k \leqslant 4} k \cdot a_k$ be the total number of edge-copies. Applying Cauchy-Schwarz twice and Fact 3.12,

$$\left| \mathbb{E}_{x \sim \mathcal{N}(0,1/d)} \left[\prod_{k \leqslant 4} h_k(x)^{a_k} \right] \right| \leqslant \mathbb{E} \left[h_1(x)^{2a_1} h_2(x)^{2a_2} \right]^{1/2} \mathbb{E} \left[h_3(x)^{2a_3} h_4(x)^{2a_4} \right]^{1/2}$$

$$\leqslant \prod_{k \leqslant 4} \mathbb{E} \left[h_k(x)^{4a_k} \right]^{1/4} \leqslant \prod_{k \leqslant 4} \left(\frac{4ka_k}{d} \right)^{k \cdot a_k / 2}.$$

$$\leqslant \left(\frac{4t}{d} \right)^{t/2}.$$

We next show that the edge factors assigned to the t edge-copies upper bound the above. Let t_0 be the number of edge-copies that get assigned $\frac{2^{1/4}}{\sqrt{d}}$. We must have $0 \leqslant t_0 < t$ and $t_0 \leqslant 4$. Then, the assignment scheme gives

$$\left(\frac{2^{1/4}}{\sqrt{d}} \right)^{t_0} \cdot \left(\frac{32qD_V}{\sqrt{d}} \right)^{t-t_0} \geqslant d^{-t/2} (32qD_V)^{t-t_0}.$$

Since the length of the walk is $2q$ and the size of the graph matrix (shape) is $\leqslant D_V$, we have $t \leqslant 8qD_V$. Thus, if $t \leqslant 8$, then clearly $(32qD_V)^{t-t_0} \geqslant (4t)^{t/2}$; otherwise, $t - t_0 \geqslant t/2$ and $(32qD_V)^{t-t_0} \geqslant (4t)^{t/2}$. This shows that the edge factors correctly account for the values from the Hermite characters.

For the special case when an edge appears as h_1, h_1, h_2, the factor $\frac{2}{d^2}$ follows from Eq. (14). This completes the proof. $\square$

4.4 Local Analysis for $\mathcal{R}$

Lemma 4.13. *For some constant $\varepsilon > 0$, for any $q < d^\varepsilon$, the block-value function for R is bounded by*

$$B_q(\mathcal{R}) \leqslant \frac{1}{10}.$$

We first state our vertex factor assignment scheme (with redistribution) which assigns vertex factors to labeled vertices according to their global appearances. It is the same one as described in Section 3.3.

Vertex factor assignment scheme

1. For each labeled vertex i making its first or last *global* appearance, assign a factor of $\sqrt{\text{weight}(i)}$;

2. For each labeled vertex i making its middle *global* appearance, assign a factor of 1 if it is reached via an R step, otherwise assign a factor $2q \cdot D_V$ as well as its corresponding $O(1)$ Pur factors.

We next describe the scheme that assigns edge-copies to vertices. In most cases, we charge edge-copies (on the dangling path) to the vertex it leads to, unless it is a *reverse-charging* edge (Definition 4.10). Recall that we define a step $u \to v$ to be reverse-charging if it is making its last global appearance and if its first global appearance is also in the current block going from $v \to u$.

"Top-down" edge-copy charging primitive

1. Assign both edges on the $U - V$ path to the first square vertex in the middle;

2. For any step $u \to v$ before the final h_2 gadget, assign the edge to v unless this is a reverse-charging edge (Definition 4.10), in which case we assign to u;

3. For the final h_2 gadget (if any), we reserve its assignment from the current scheme.

See Section 4.5 for an illustration of the above scheme. We next show the following invariant throughout the walk,

Proposition 4.14. *We can assign each edge-copy to at most one vertex,*

1. *for a circle vertex, it is assigned* 1 *edge-copy if it is making first/last* global *appearance, and* 2 *edge-copies if both;*

2. *for a square vertex on the dangling path, it is assigned* 2 *edge-copies if it is making first/last* global *appearance, and* 4 *edge-copies if both;*

3. *for any square vertex's global middle appearance yet local first appearance in the block, it is assigned at least* 1 *edge-copy;*

4. *no surprise/high-mul step is assigned to any vertex's* global *first/last appearance.*

Proof. We start by giving the argument when $U \neq V$, and then show it can be modified into an argument for $U = V$.

Charging vertices outside $U \cup V$ and the final h_2 gadget. The above scheme applies immediately to vertices that appear in the current block while not in $U \cup V$ or the last square vertex as well as the final circle vertex it connects to in the h_2 gadget.

It follows by observing that any circle (resp. square) vertex that makes its first *global* appearance is the destination of 1 (resp. 2) edge-copies that are not reverse-charging, in which case the edge-copies are assigned to the particular vertex. This holds analogously for vertices making the last

29

global appearance but not the first *global* appearance in the block, since the edges are not reverse-charging. We note that this is true for the first square vertex as well since we assign both edges on the $U - V$ path to the first square vertex.

On the other hand, for vertices making both the first and the last *global* appearances, consider the assignment until the final h_2 gadget, the charging above gives us 1 and 2 edges for each such vertex when it first appears *locally*. Furthermore, notice that each of these edges need to be closed, and they are assigned to the destination vertex, i.e., the particular vertex in inspection, and this completes the proof of our assignment restricted to vertices outside $U \cup V$ and the final h_2 gadget.

Charging the final gadget and finding "block-reserve". The goal is to identify an edge-copy as block-reserve, and assign vertex factors to corresponding edges in the final h_2 gadget (if any). We first consider the case when there is no final h_2 attachment, i.e. the term from $\frac{u+s}{s^2-ru} A^{-1} 1_m$. Recalling our bounds from Lemma 2.10 that $s^2 - ru = \Omega\left(\frac{m}{d}\right)$, and $s = O(\sqrt{d}), |u| = O(1)$ (here the weaker bound on s suffices), we observe that

$$\left| \frac{u+s}{s^2-ru} \right| = O\left(\frac{\sqrt{d}}{m/d} \right) = O\left(\frac{1}{\sqrt{d}} \right),$$

and the normalizing constant can be regarded as an edge-copy. This is assigned as the block-reserve factor if there is no final h_2 gadget, and there is no vertex factor involved when there is no final h_2 gadget since this may be the last appearance of a square vertex while its factor has already been assigned to the edges that first lead to this vertex in the current block.

We now proceed to assign edges from the final h_2 gadget to vertex factors involved in the final h_2 gadget, moreover, we will identify one edge-copy of assigned factor at most $\frac{\sqrt{2}}{\sqrt{d}}$ as "block-reserve" that allows us to further charge vertices in $U \cup V$.

For the vertices in the final h_2 gadget, we observe the following,

1. in the final gadget, the last square vertex cannot be making its first *global* appearance yet it may be making its last appearance (since it is on the gadget boundary, we consider it appears in both the final h_2 attachment gadget and the gadget that precedes it);

2. the last circle vertex may be making either its first or last *global* appearance but it cannot be both;

3. the h_2 edge in-between has not yet been assigned, and we can split it as two edge-copies, each of weight $\frac{2^{1/4}}{\sqrt{d}}$ if F/R under the *global* step-labeling.

We now observe the following,

1. If the final h_2 attachment is not a reverse-charging edge assigned to the source square vertex, the square vertex has also been assigned by 2 factors if making its first/last *global* appearance, and 4 if both, in the current block by the previous charging. In this case, note that we can reserve one of the two edge-copies, that is a factor of at most $\frac{2^{1/4}}{\sqrt{d}}$,

   - If the circle vertex is making first/last appearance, we either have both edge-copies receiving F/R labeling, or a mix of H, R labeling. In the first case, we have two assigned factors of $\frac{2^{1/4}}{\sqrt{d}}$. Assign one to the circle vertex's first/last appearance, and another to the block-reserve. In the second case, we have a mix of H, R edges, that we at least have

the underlying random variable appears twice as h_1 edges and at least once as an h_2 edge, by moving $\tilde{O}(1)$ polylog factors to the second appearance of the underlying random variable locally, we have again two edge copies of value $\frac{2^{1/4}}{\sqrt{d}}$, and the assignment follows from above.

- If the circle vertex is making a middle appearance, we may assign related Pur factors and vertex-factor cost to one edge-copy such that one edge-copy is of value $\tilde{O}(\frac{1}{\sqrt{d}})$ while the other is still at most $\frac{2^{1/4}}{\sqrt{d}}$; assign the edge-copy with $\tilde{O}(\frac{1}{\sqrt{d}})$ to the middle appearance factor and assign the edge-copy with factor $\frac{2^{1/4}}{\sqrt{d}}$ for block-reserve.

2. If the final attachment h_2 edge is a reverse-charging step that corresponds to an edge whose F copy leads to the final square vertex, we assign both h_2 edges to the final square vertex as the square vertex may be making its last *global* appearance. Note that if the square vertex makes its first *global* appearance in the current block, it has already been assigned 2 edges.

   This leaves the last circle vertex potentially uncharged as it may be making its last appearance as well. Furthermore, we need to find one more "reserve" edge-copy for the block which would then be used to charge $U \cup V$.

3. **Finding critical edge:** we now give a procedure to identify the critical edge, that is, an h_2 edge assigned to a circle vertex in the above top-down charging process. The process maintains a current circle vertex and its current gadget along the top-down path. In particular, the gadget considered is an M_β gadget throughout. The process starts from vertex s, that is the circle vertex involved in the final h_2 attachment, and the gadget considered is the one in which s opens up the F step of the current edge e^*, the h_2 edge in the final gadget. Note that this is an M_β gadget. We now case on the step-labeling of the top half of the M_β gadget in inspection, in particular, the edges leading to the circle vertex s in that particular gadget,

   - This is an F, non-reverse-charging R, or H step assigned to s, this is the critical edge we aim to find, as we have two edge-copies assigned to a circle vertex, and the process terminates.

   - This is a reverse-charging R edge: update the circle vertex to be the top vertex of the current M_β gadget s, and update the edge e^* to be the h_2 edge in the top half of the current gadget, and repeat the above process. Note that the updated gadget must appear, and additionally, on top of the current gadget in the dangling path.

   We now observe that this process ultimately terminates since each time we move up towards the top of the dangling path. Once the critical edge is found, note that it contains two edge-copies, assign one to the vertex's factor, and another to the block-reserve. In the case of H edges involved, locally assign the edge-value as $\frac{1}{\sqrt{d}}$ and $\tilde{O}(\frac{1}{\sqrt{d}})$, and assign the copy with $\frac{1}{\sqrt{d}}$ for block-reserve, while the other for vertex's factor.

Reserving surprise/high-mul step from vertex-factor assignment outside $U \cup V$ We first observe that in the above assignment, it is clear that the edges assigned to vertex's first appearance cannot be surprise-visit nor high-mul step. That said, it suffices for us to consider the assignment for vertex's last appearance factor (whose first appearance is not in the same block). Consider

31

such vertex's first local appearance, they may either be H/S edges if not R under the global edge-labeling. If so, since the vertex is making the last appearance in the current block, each corresponds to an R-step in this block. Moreover, observe that any such R-step is intended for "reverse-charging" if the vertex in inspection is making both first and last global appearance in the block, and thus it is not assigned to the vertex factor of the source vertex. That said, it suffices for us to swap the H/S step with the R step so that no surprise/high-mul step is assigned to vertex's (polynomial) factor.

At this point, for a block with $U \neq V$, we have assigned

1. 1 or 2 edge for each vertex's first/last appearance in the current block outside $U_\alpha \cup V_\alpha$ depending on the vertex's type;

2. 1 edge-copy of value $O(\frac{1}{\sqrt{d}})$ has been identified as the block-reserve.

3. it is also straightforward to observe that any edge assigned for vertex's first/last appearance cannot be a surprise visit nor high-mul visit.

Charging circle vertices in $U \cup V$

Proposition 4.15 (Unbreakable $U - V$ path). *There is always a $U - V$ path such that each edge along the path is of odd multiplicity. Call this path P_{safe}.*

Proof. It suffices for us to restrict our attention to paths of only h_1 edges. By our gadget property, each vertex except $U_\alpha \cup V_\alpha$ is incident to an even number of h_1 edges, while only vertex copy in $U_\alpha \cup V_\alpha$ is of odd degree. Suppose the path is broken, consider the (maximal) component C_U connected to U_α (but not V_α), we first observe that the edge-multiplicity of $E(C_U, V(\alpha) \setminus C_U)$ must be even, as otherwise, some edge is of odd multiplicity, and the edge is included inside C_U as opposed to be across the cut. That said,

$$\sum_{v \in C_U} \deg(v) = 2 \cdot \text{mul}(E(C_U)) + \text{mul}(E(C_U, V(\alpha) \setminus C_U)).$$

Observe that the LHS is odd as any vertex copy except U_α has even degree, and we have exactly 1 odd-degree vertex copy. On the other hand, RHS follows as each edge-multiplicity in $E(C_U)$ contributes a factor 2 to degree of vertices in C_U while edges across the cut contribute 1 for each multiplicity; since $E(C_U, V(\alpha) \setminus C_U)$ is even by connectivity argument above, the RHS is also even and we have a contradiction. $\qquad\square$

Corollary 4.16. *There is at least one vertex v^* making middle-block appearance, i.e. it is a vertex that appears in previous blocks, and is appearing again in future blocks.*

We recall our charging so far that we have one edge-copy reserved, while we have two circle vertices in $U \cup V$ uncharged. And we now show that the single-edge copy is sufficient. i.e., at most one vertex factor is picked up among these two vertices.

Consider the path P_{safe}, and observe that it passes through both vertices in $U \cup V$. Take v^* to be the first vertex on the path that pushes out an $F/H/S$ edge. In the case $v^* = U$, note that it suffices for us to assign the reserve factor for the vertex in V; similarly, assign the reserve factor for U for $v^* = V$. In the case $v^* \notin U \cup V$, note that the previous scheme assigns at least one edge

for v^* if it is a circle, and 2 if it is a square when v^* first appears in the current block, and this is in fact not needed since v^* is making a middle appearance. That said, we have at least 2 factors, 1 from the reserve factor, and 1 from the edge-assignment for v^* in the previous scheme, we now assign these two factors for $U \cup V$.

Remark 4.17. v^* is picked such that it is either reached by an R edge, or it is the boundary vertex U_α. In other words, no factor is needed for specifying v^* of the block.

and this completes the proof to our proposition when $U \neq V$.

Analysis for $U = V$ We now consider the case when $U = V$.

Definition 4.18 (SQ_1: first square vertex in R term). For each R term, call the first square vertex on top of the dangling path the SQ_1 vertex.

Remark 4.19. This vertex is an exception from any other square vertex as its first appearance might be the destination of two edge-copies that correspond to the same underlying random variable. This happens when an edge making its first and last appearance at the same time for the $U = V$ term.

For $U = V$, the charging for vertices outside is identical except for the first square vertex SQ_1: since the two edges assigned to it now correspond to the same underlying edge-copy, and may now receive F, R-labeling. That said, both edges are now assigned to the first square vertex. If the square vertex is making either first/last appearance but not both in the current block, the current assignment is sufficient. However, there are no reverse-charging edge copies protecting SQ_1, and that warrants a further analysis.

We follow the previous strategy in identifying block-reserve: even though for $U = V$, the vertex $U = V$ is by definition not contributing any vertex-appearance factor, we now intend the block-reserve factor to pay for either additional Pur factor due to the dormant step stemming from $U = V$, or the last appearance factor of SQ_1 but not both.

1. SQ_1 makes both first and last appearance in the current block: in this case, the edges assigned to SQ_1 in the beginning of the top-down charging process are assigned to the first appearance of SQ_1. That said, it remains for us to identify two extra edge-copies for the last appearance factor.

   Charging for terms without final h_2 attachment: these terms come with a normalization of $\left|\frac{u+s}{s^2-ru}\right| = O(\frac{1}{d})$, and these are two edge copies that we assign to the last appearance of SQ_1.

   Charging for terms with final h_2 attachment: we first note that any edge incident to SQ_1 must be closed, and further case in whether there is any surprise visit arriving at SQ_1 throughout the dangling path.

   (a) Suppose that there is no surprise visit arriving at SQ_1 throughout the dangling path. Consider the final departure from SQ_1, and note that it pushes out at least two edge-copies.

      - If this is an M_β gadget, let r be the circle vertex it connects to via an h_2 edge. Suppose the random variable corresponding to (SQ_1, r) first appears as an h_2 edge, in

this case, the circle vertex has been assigned two edge-copies the first time it appears as it is reached using h_2 edge, that said, we can reserve the other two edge copies from the final attachment for the missing last appearance factor of SQ_1;

- If this is an M_β gadget, yet the random variable (SQ_1, r) first appears as an h_1 edge in the current block. Note that there is at least one more edge-copy of h_1 edge that is currently assigned an H-label, observe that we can replace its label to be R, and assign both F/R copies to the vertex factor of the circle vertices. We now relabel the final h_2 step to be H^*, and note that these two edges get assigned a value at most $\frac{2}{d}$, and assign them both to the missing last appearance factor of SQ_1.

- To see that H^* does not require extra $D_V \cdot q$ factor, we observe that this is a circle vertex traversed before in the block, and for each block, H^* is unique. That said, it suffices for us to use a special label in [2] when H^* first appears in the block to identify the edge. Moreover, note that swapping H^* with the R step does not add to Pur factor of the destination circle vertex as this edge no longer appears;

- If this is an M_α gadget, let r_1, r_2 be the circle vertices the square vertex is connected to. Note that for each r_i, there must be at least two more edge-copies of random variable of (SQ_1, r_i) that receive H label, i.e., we have a factor of $\tilde{O}(\frac{1}{d})$. That said, we may reassign the factors and extract one full factor of $\frac{1}{\sqrt{d}}$ (with the remaining being $\tilde{O}(\frac{1}{\sqrt{d}})$, and since we have two of them that combine to a factor of $\frac{1}{d}$, assign it to the missing last appearance factor of SQ_1.

(b) There is a surprise visit arriving at SQ_1. We first observe this must be either an h_2 edge, or a pair of distinct h_1 edges. The surprise visit must be closed already, and their underlying R copies are intended for reverse-charging in the top-down charging scheme for the last appearance of SQ_1. In this case, the factor for the last appearance has already been assigned edges.

2. **Finding "block-reserve" for $U = V$:** SQ_1 is not making both first and last appearance in the current block, we first note that the departure from $U \cup V$ is special as this is the only vertex that can appear on the boundary again without being reached by any edge, and this is not a dormant gadget M_D. That said, the most recent departure may contribute Pur factor to the vertex in $U \cup V$, and we now identify an edge-copy for this factor.

(a) **Charging for terms without final h_2 attachment:** for $A^{-1}1_m$ term where the final h_2 gadget is missing, we pick up a normalization constant of $O(\frac{1}{d})$ and we designate this as the block-reserve.

(b) **Charging for terms with final h_2 attachment:** for terms with the final h_2 gadget, the analysis of finding block-reserve from the case $U \neq V$ applies identically for finding an h_2 edge that gets assigned to a circle vertex in the top-down traversal process. In particular, we may designate one edge-copy among the two copies of the identified h_2 edge as a block-reserve to charge the corresponding $\tilde{O}(1)$ factors from Pur.

Remark 4.20. Note that the edge identified from the above process is in fact stronger, as it carries a factor of $O(\frac{1}{\sqrt{d}})$ as opposed to $\tilde{O}(\frac{1}{\sqrt{d}})$. That said, since for $U \neq V$ terms, the block-reserve is only assigned for $\tilde{O}(1)$ factors from Pur, either is sufficient.

$\square$

4.5 Illustration via diagrams
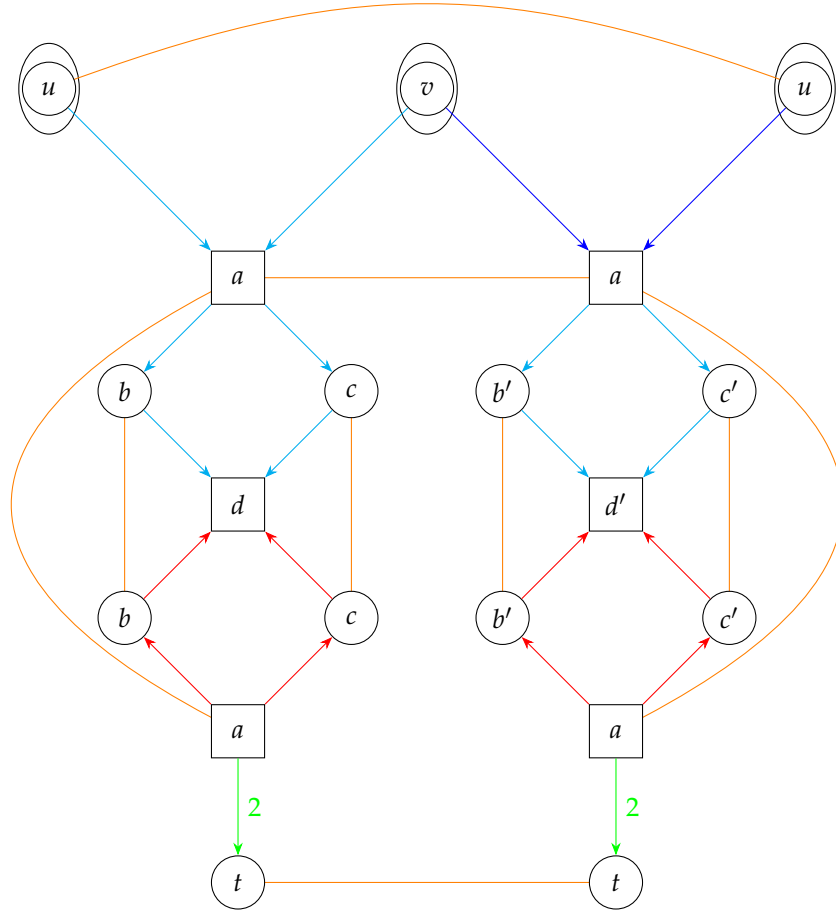
This figure illustrates the charging scheme. The light blue edges are *F* edges and we assign their factor to their destination. The dark blue edges are *R* edges whose first appearance is in a different block. We also assign these edges to their destination. The red edges are reverse-charging *R* edges whose first appearance is in the current block. We assign these edges to the destination of the corresponding *F* edge (which is generally the source for this edge).

Note that two edge factors are missing from v. We obtain these factors from the two green edges pointing towards t.
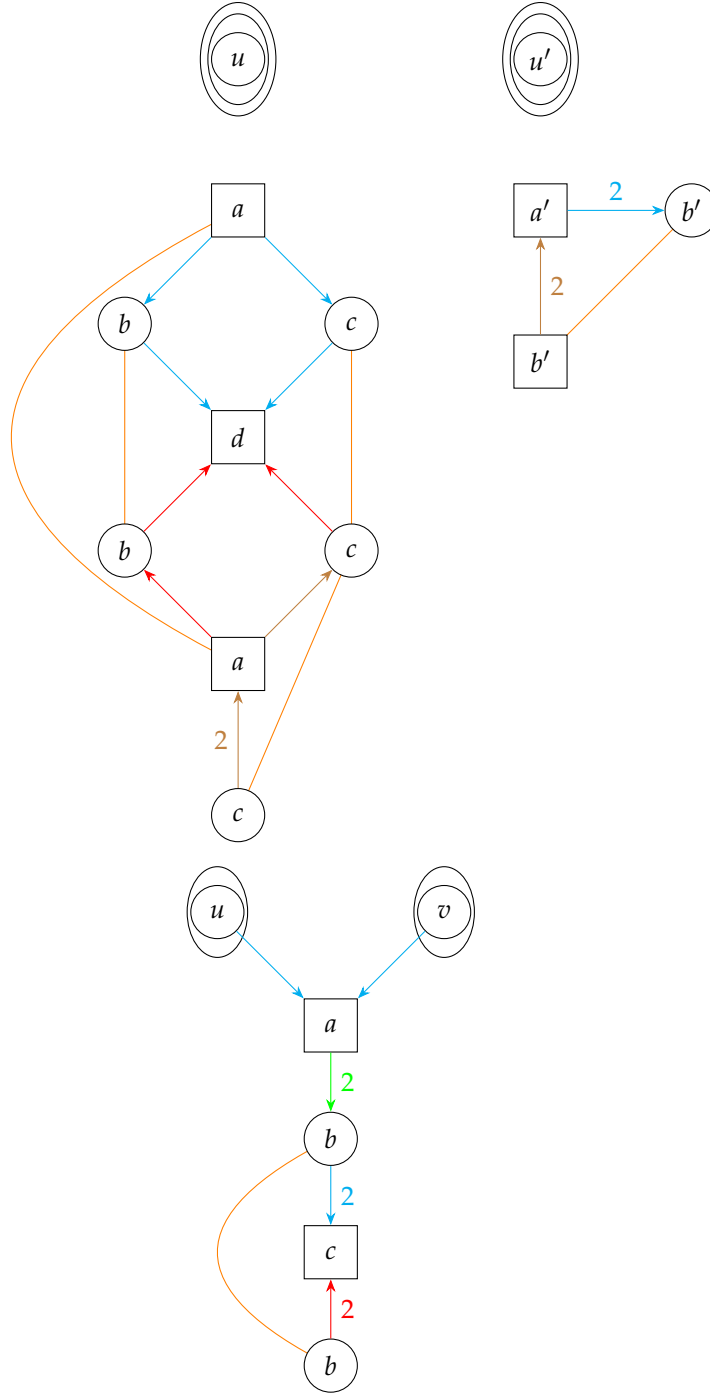
4.6 Pur **bound for square vertices in** $\mathcal{R}$

The prior Pur bound does not apply well for square vertices in $\mathcal{R}$, in particular, it should be pointed out that even before non-trivial intersection within each block along the dangling path, the Pur argument based on 1-in-1-out (or its slightly generalized version of 2-in-2-out) falls apart in the analysis for $\mathcal{R}$.

In particular, one may consider a walk on R_S with the square vertex along the $U - V$ path fixed throughout the walk. It is easy to verify the the fixed square vertex may have growing unclosed F

edges without any surprise visit/high-mul step. That said, it is not sufficient to use the slack from such factors to offset the potential confusion due to R edges.

In this section, we give a new argument to handle the Pur factor in the vanilla setting of R when there is no non-trivial intersection along the dangling gadget-path, and then extend it for the general cases of R where each block is not necessarily injective due to intersection across gadgets.

Gap from square middle appearance   For starters, we observe that a vertex may only push out unforced returns if it is making a middle appearance given that it maintains a list of incident edges, including the additional information which are closed. When it is making the last appearance, any currently unclosed edge needs to be closed, and therefore shall be pushed out, giving us a fixed set of edges being pushed-out (where we momentarily ignore the question whether one needs to distinguish among the edges in the edge-set). This immediately renders us the following bound on Pur readily,

Lemma 4.21 (Pur factor via middle appearance: beginner version). *For any vertex v, let $\mathsf{MidApp}(v)$ be the number of middle appearances of v throughout the walk, we have*

$$\mathsf{Pur}(v) \leqslant 3 \cdot \mathsf{MidApp}(v)$$

provided each square vertex pushes out at most 3 edges in each block throughout the walk, and each block is vertex-injective.

With the above Pur bound, it may not be meaningful if we cannot obtain a slack from MidApp. Fortunately, this is indeed the case for our setting, and we first observe this in the vanilla setting where there is no gadget-incursion within each block, (in particular, this already applies immediately if the edges are injective witin each block),

1. We have assigned each square vertex at least 1 edge when it is making a middle appearance: to see this, the top-down charging scheme assigns each square vertex 2 edges when the square-vertex appears for the first time in the block; in the case of charging $U \cup V$, 1 edge may be re-routed from a square vertex making a *global* middle appearance. That said, at least 1 edge is assigned to each square vertex when it makes a middle appearance.

2. Note that for a fixed vertex v in a given block, its *local* first appearance at the given block may be corresponding to a *global* middle appearance, and such *mismatch* is the source of our slack;

3. Observe that each middle-appearance corresponds to a mismatch described above (provided each block is injective), and each such mismatch of local-global first appearance assigns a vertex making global middle appearance one edge-copy, that is a factor of $O(\frac{1}{\sqrt{d}})$;

4. However, since each vertex's middle appearance does not get assigned any vertex factor in our scheme, we may use the $\frac{1}{\sqrt{d}}$ gap to offset the 3 Pur factors corresponding to the particular *global* middle appearance, that is

$$O(\frac{1}{\sqrt{d}}) \cdot (q \cdot D_V)^3 = o_d(1).$$

Extension to gadget intersections To handle dangling paths with potentially intersecting gadgets, it should be pointed out the Pur bound goes through as stated while we do not necessarily have a gap from middle-appearance. In particular, it is possible now that in a given block, a square vertex appears for various times and only has 2 edges assigned to it for its first local appearance, as any of its subsequent appearance in the given block follow via closing some F edge that gets opened up earlier in the work, and thus assigned to the destination as opposed to the given square vertex.

Towards generalizing the prior argument, we consider a specific subclass of *global* middle appearance of a vertex through the block walk,

Definition 4.22 (Middle appearance for a square vertex). For a given walk and a given block-step $\mathsf{BlockStep}_i$, we say a labeled vertex $v \in [m]$ makes a middle appearance in $\mathsf{BlockStep}_i$ if

1. v makes appearance at $\mathsf{BlockStep}_i$;

2. v makes appearance at some $\mathsf{BlockStep}_j$ for $j < i$, and at some $\mathsf{BlockStep}_{j'}$ for $j' > i$.

With some abuse of notation, we continue to let $\mathsf{MidApp}(v)$ denote the number of middle appearances of v. Note that this is clear when we are working with blocks that do not have block-injectivity, while it is equivalent to the previous definition in vertex-injective blocks.

In particular, we emphasize that following the above definition, in the case of a labeled vertex first appears at $\mathsf{BlockStep}_i$, and appears multiple times in various gadgets in the dangling-path of $\mathsf{BlockStep}_i$, it is not considered as making a middle appearance at $\mathsf{BlockStep}_i$.

Lemma 4.23 (Pur bound for square vertices). *For any square vertex v that does not push out dormant gadgets, at any time-t,*

$$\mathsf{Pur}_t(v) \leqslant P_t(v) \leqslant 3 \cdot \mathsf{MidApp}_t(v) + 3(s_t(v) + h_t(v))$$

where we define

$$P_t(v) := \#(R \text{ steps closed from } v \text{ by time } t) + \#((\text{unclosed edges incident to } v \text{ at } t)) - 4\,.$$

In other words, by assuming at 4 possible return legs to be fixed each time a vertex is on the boundary, the number of unforced returns from a vertex by time t is at most $3 \cdot \mathsf{MidApp}_t(v) + 3(s_t(v) + h_t(v))$.

Proof. The first inequality is definitional as we assume each vertex may have 4 edges being fixed, which incurs a cost of $[4]$ for each vertex each time it pushes out an R step. Analogous to previous Pur bounds, the base case is immediate when vertex first appears in the walk. In particular, we note that the above bound can be strengthened for the $\mathsf{BlockStep}_i$ in which vertex v makes (globally) its first appearance.

Claim 4.24. *For any square vertex v, for the $\mathsf{BlockStep}_i$, at any time-t within the $\mathsf{BlockStep}_i$,*

$$P_t(v) \leqslant 3(s_t(v) + h_t(v)) - 2\,.$$

Proof. Notice this is immediate when vertex v first appears, as it is incident to 2 edges, and thus we have $P_t^{(1)} = -2$ (as we have a -2 term since we maintain 3 edges to be fixed instead of just 2). The invariant holds as any subsequent departure opens up at most 2 F edges, and any subsequent arrival either closes both edges, or either arrival is along a surprise/high-mul visit, and give a net-gain of at most 3 in the number of unclosed F edges. This proves our claim. $\qquad\square$

It remains for us to consider the appearance of v in subsequent blocks, in particular, we start with the *locally* first appearance of the subsequent block. Let t be the time-mark in which v is making its first *local* appearance in the current-block. Notice at time t when the vertex first appears, it may be arrived via a single edge (as opposed to 2 due to the $U - V$ path), and therefore it may push out at most 3 edges.

Appearance at the second block   If not, this is currently the second block in which v appears: applying the claim on the first block, and observe that the most recent departure opens up at most 2 F edges, while the current arrival closes 1 (unless H or S, in which case a net-gain of 3 suffices), we have

$$P_t(v) \leqslant 3(s_t(v) + h_t(v)) - 2 + 2 - 1 = 3(s_t(v) + h_t(v)) - 1$$

where the $+2$ corresponds to the 2 F edges opened up at the most recent departure, and the -2 term corresponds to the term in the hypothesis on first-block, and the -1 comes from the current arrival closing at least one R edge.

For any subsequent appearance of v in the current block, if any, the following is immediate,

$$P_{t'}(v) - P_t(v) \leqslant 3(s_{t'}(v) - s_t(v)) + 3(h_{t'}(v) - h_t(v)) + 1$$

as we observe that

1. The departure from the first vertex may open up at most 3 edges instead of 2;

2. Any subsequent departure and arrival closes 2 edges, and opens up at most 2 edges, hence the previous argument applies, giving a net-gain of $+1$ due to the extra opening in the first departure.

That said, for any appearance of v in the second block at time t', we have

$$P_{t'}(v) = P_{t(v)} + (P_{t'}(v) - P_t(v)) \leqslant 3(s_{t'}(v) + h_{t'}(v))$$

Appearance at the future blocks   For any block, let t_0 be the local first appearance, and t_1 be the local final appearance, applying the above argument gives

$$P_{t_1}(v) - P_{t_0}(v) \leqslant 3(s_{t_1}(v) - s_{t_0}(v)) + 3(h_{t_1}(v) - h_{t_0}(v)) + 1 \,.$$

That said, it suffices for us to bound $P_{t_0}(v)$. This is bounded by

$$P_t(v) \leqslant 3 \cdot \mathsf{MidApp}_t(v) + 3(s_t(v) + h_t(v))$$

Consider the base case when v appears at the third block, the most recent departure opens up at most 2 new F edges. To offset this, we use the gain in $\mathsf{MidApp}_t(v)$, as the appearance of v in the second block is now counted as a middle appearance once v appears in the third block, which gives a $+3$ on the RHS. The bound extends to any subsequent block immediately. This completes our proof of Pur bound.

Extension to dormant gadgets To capture the Pur change due to dormant gadget, we note that for each square vertex at each block, we can assume one dormant edge it pushes out being fixed while assign a Pur factor for any other dormant gadget it pushes out at that block. Notice this is a cost at most 2 for each square vertex throughout the walk, as we may need to assume 1 dormant edge for its first appearance, and another for its last appearance. For any middle appearance, it suffices for us to assign a Pur factor for any dormant edge it pushes out, as opposed to all-but-one in the case of first/last appearance. This prompts to define the following counter,

Definition 4.25 (Dormant-excess). For each square vertex v, at any block $\mathsf{BlockStep}_i$, let D_i be the number of dormant-gadgets it pushes out at this block, define $D_i(v) := D_i - 1[\text{first/last appearance at i}]$, and additionally, the counter function is defined for the whole walk by taking

$$D(v) = \sum_{i:v \text{ appears in BlockStep}_i} D_i(v).$$

Remark 4.26. By using an extra additive constant of 2, each return using dormant leg is either forced or accounted for in $D(v)$.

Claim 4.27. *Each dormant-excess gadget corresponds to a circle vertex reached using an h_2 edge, and gets assigned a factor of at most $\tilde{O}(\frac{1}{\sqrt{d}})$ in the combinatorial charging argument.*

Proof. Note that in the combinatorial charging argument, we have assigned both h_2 edges to the circle vertex's vertex factor, unless the square vertex is potentially making its first and last appearance at the same block, which is not ruled out by the definition of dormant excess. That said, for the circle vertex, it gets assigned at most a factor of $\sqrt{d}$ in our scheme, while both edges assigned at least $\tilde{O}(\frac{1}{\sqrt{d}})$ each, and combining the above yields the desired. □

Corollary 4.28. *By assuming at most 6 return legs to be forced, the number of unforced-return from v throughout the walk is at most*

$$\mathsf{Pur}(v) \leqslant 3 \cdot \mathsf{MidApp}_t(v) + 3(s_t(v) + h_t(v)) + D(v).$$

This allows us to effectively ignore Pur factor in the block-value analysis, as each Pur factor can be distributed among surprise visit, high-mul visit, or middle appearance such that each is assigned at most 3 Pur factors. Moreover, since each comes with a $O(\frac{1}{\sqrt{d}})$ factor, combining with the assigned Pur factor contributes an $o_d(1)$ term provided $\frac{q^3}{\sqrt{d}} = o_d(1)$, which is sufficient for us as we set $q = d^\varepsilon$ for a small enough constant ε. □

4.7   Wrapping up

Given the edge-assignment scheme to vertex appearance, we now show why this immediately gives the desired bound on $B(\mathcal{R})$: we have the following factors,

1. Each circle vertex gives a factor of $\sqrt{d}$ for their first/last appearance; the assignment gives each vertex one *global F/R* edge-copy each, that is a factor of $\frac{\sqrt{2}}{\sqrt{d}}$, giving a bound of

$$\sqrt{d} \cdot \frac{\sqrt{2}}{\sqrt{d}} = \sqrt{2};$$

2. Each square vertex gives a factor of $\sqrt{m}$ for their first/last appearance, and the assignment above gives *global F/R* each two edge-copies each, that is a factor of

$$\sqrt{m} \cdot \frac{2}{d} \leqslant \frac{2\sqrt{m}}{d} \; ;$$

3. Each square vertex that makes a *global* middle appearance gives a $O(1)$ factors of Pur, while each such vertex appearance is assigned one edge, that is a factor of

$$(2q \cdot D_V)^{O(1)} \cdot O\left(\frac{\sqrt{2}q^2}{\sqrt{d}}\right) \ll 1 \; .$$

Corollary 4.29. *For each vertex's appearance and edges assigned to it, we have a factor of at most*

$$(1 + o_d(1)) \cdot \sqrt{d} \cdot \frac{\sqrt{2}}{\sqrt{d}} \cdot 12 \leqslant (1 + o_d(1)) \cdot 6\sqrt{2}$$

for a circle vertex, and

$$(1 + o_d(1)) \cdot 6 \cdot \sqrt{m} \cdot \frac{2}{d} \leqslant (1 + o_d(1))\frac{12\sqrt{m}}{d}$$

for a square vertex where the factor 6 for each vertex comes from our bound that each vertex arrived using a forced R edge can be specified at a cost of [6].

Proof to Lemma 4.13. By our edge-copy charging scheme and Proposition 4.14, we observe the following,

1. Treat the $U - V$ path as a gadget, and we have 2 choices depending on whether $U = V$ for the upcoming block;

2. For each gadget-step, we sum over the edge-labelings;

3. For each gadget along the dangling path, we pick up at most the following factor,

   * M_α: it gives 2 circle vertices and 1 square vertex, which is a factor of at most

   $$B_\alpha := (1 + o_d(1)) \cdot 3^4 \cdot \frac{4}{d^2} \cdot \sqrt{md^2} \cdot 6^3 \; ;$$

   * M_β: it gives 1 circle vertex and 1 square vertex, which is a factor of at most

   $$B_\beta := (1 + o_d(1)) \cdot 3^4 \cdot \frac{2}{d^2} \cdot \sqrt{md} \cdot 6^2 \; ;$$

   * M_D: it gives 1 circle vertex, which is a factor of at most $(1 + o_d(1)) \cdot 3^2$ if it is the first M_D gadget of the block with the factor 3 counting the step-label of the edge-copies in the gadget, or a factor of at most $B_D := \tilde{O}(\frac{1}{\sqrt{d}})$ for any subsequent M_D gadgets.

4. For the $U - V$ path, the square vertex SQ_1 gives a factor of at most

$$(1 + o_d(1)) \cdot 6 \cdot \sqrt{\frac{m}{d^2}}$$

and the circle vertex combined gives a factor of at most

$$(1 + o_d(1)) \cdot (\sqrt{2})^2 \cdot 6$$

where the factor of 6 comes from at most one circle vertex being reached using R edge.

41

Therefore, combining the above bounds, we have

$$B(\mathcal{R}) \leqslant (1 + o_d(1)) \cdot 6(\sqrt{\frac{m}{d^2}} \cdot 6 \cdot \sqrt{2}^2) \cdot 3^2 \cdot \prod_{\leqslant \text{polylog } d \text{ gadgets}} (B_\alpha + B_\beta + B_D) < \frac{1}{2}$$

provided

$$B_\alpha, B_\beta < \frac{1}{2}$$

and

$$6(\sqrt{\frac{m}{d^2}} \cdot 6 \cdot \sqrt{2}^2) \cdot 3^2 < \frac{1}{2}$$

It can be verified that it suffices for us to take $m < \frac{1}{7000000} \cdot d^2$ though we do not emphasize upon the particular constant as we believe a more careful argument by tracking our above bounds can render an improved constant without much work. □

Lemma 4.30 (Restatement of Lemma 2.11). *With probability at least $1 - 2^{-d^\varepsilon}$ for some constant $\varepsilon > 0$,*

$$\|\mathcal{R}\|_{\mathrm{op}} < \frac{1}{2}.$$

Proof. This follows by setting $q = d^\varepsilon$ for some constant $\varepsilon > 0$ in Proposition 3.9, and combines with our block-value function for small enough constant c. □

Acknowledgements

References

[AMP16] Kwangjun Ahn, Dhruv Medarametla, and Aaron Potechin. Graph matrices: Norm bounds and applications. *arXiv preprint arXiv:1604.03423*, 2016. 2, 4, 10

[BHK+19] 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. 0, 2, 4, 5, 10

[BHKX22] 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. 2, 4

[BMMP23] Afonso S. Bandeira, Antoine Maillard, Shahar Mendelson, and Elliot Paquette. Fitting an ellipsoid to a quadratic number of random points. 2023. 2

[CP20] Wenjun Cai and Aaron Potechin. The spectrum of the singular values of z-shaped graph matrices. *ArXiv*, abs/2006.14144, 2020. 4

[CP22]   Wenjun Cai and Aaron Potechin. On mixing distributions via random orthogonal matrices and the spectrum of the singular values of multi-z shaped graph matrices. 2022. 4

[GJJ+20]   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. 1, 3, 4, 5

[HK22]   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. 4

[HKP19]   Christopher Hoffman, Matthew Kahle, and Elliot Paquette. Spectral gaps of random graphs and applications. *International Mathematics Research Notices*, 2019. 2

[JPR+22]   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. 2, 4

[KD22]   Daniel M Kane and Ilias Diakonikolas. A Nearly Tight Bound for Fitting an Ellipsoid to Gaussian Random Points. *arXiv preprint arXiv:2212.11221*, 2022. 0, 1, 2, 3, 5

[MRX20]   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. 4, 5

[PTVW22]   Aaron Potechin, Paxton Turner, Prayaag Venkat, and Alexander S Wein. Near-optimal fitting of ellipsoids to random points. *arXiv preprint arXiv:2208.09493*, 2022. 0, 1, 2, 3, 5

[Sau11]   James Saunderson. *Subspace identification via convex optimization*. PhD thesis, Massachusetts Institute of Technology, 2011. 0, 1, 3, 5

[SCPW12]   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. 1

[SPW13]   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. 0, 1, 3

[Tao12]   Terence Tao. *Topics in random matrix theory*, volume 132. American Mathematical Soc., 2012. 4, 10, 13, 17, 20

[Tro15]   Joel A Tropp. An introduction to matrix concentration inequalities. *arXiv preprint arXiv:1501.01571*, 2015. 25

[Woo50]   Max A Woodbury. Inverting modified matrices. *Memorandum Rept. 42, Statistical Research Group*, 1950. 8

A Deferred calculations

Claim A.1 (Restatement of Claim 2.9). *With probability at least $1 - o_d(1)$,*

$$\|\eta\|_2^2 \leqslant (1 + o_d(1))\frac{2m}{d}.$$

Proof. We first unpack the inner product,

$$\|\eta\|_2^2 = \sum_{i \in [m]} \eta_i^2 = \sum_{i \in [m]} \left(\sum_{j \in [d]} (v_i[j]^2 - \frac{1}{d}) \right)^2 = \sum_{i \in [m]} \sum_{j_1, j_2 \in [d]} \left(v_i[j_1]^2 - \frac{1}{d} \right) \left(v_i[j_2]^2 - \frac{1}{d} \right).$$

For any $q > 0$, we then have

$$\mathbb{E}[\|\eta\|_2^{2q}] = \mathbb{E} \left[\sum_{\substack{s_1, s_2, \ldots, s_q \in [m]}} \sum_{\substack{a_1, a_2, \ldots, a_q \in [d] \\ b_1, b_2, \ldots, b_q \in [d]}} \prod_{i=1}^{q} (v_{s_i}[a_i]^2 - \frac{1}{d}) \cdot (v_{s_i}[b_i]^2 - \frac{1}{d}) \right] \leqslant (1 + o_d(1)) \left(\frac{2md}{d^2} \right)^q$$

where the final bound follows in a similar way to our norm bounds as

1. For each new s_i, we case on whether $a_i = b_i$, and if so, we pick a factor of d for $a_i = b_i$ and a factor of m for s_i;

2. Otherwise, if we have $a_i \neq b_i$, since each edge is mean 0, we can assign a factor of $\sqrt{d \cdot 2q}$ to each of the a_i, b_i for their first two appearances, and a factor of q for a_i, b_i making their third appearances and onwards; similarly, we assign a factor of $\sqrt{m \cdot q}$ for the first two appearances of s_i or q if s_i is not making the first two appearances;

3. Each H_2 edge $(v_{s_i}[t_i]^2 - \frac{1}{d})$ gets assigned its standard deviation that is $\frac{\sqrt{2}}{d}$ when it appears for the first two times, otherwise a factor of $O(\frac{q}{d})$;

4. To see the final bound, notice we can case on whether each H_2 edge is making their first two appearances.

   - In the case both edges making first two appearances, we pick up an edge-value of exactly $(\frac{\sqrt{2}}{d})^2 = \frac{2}{d^2}$; otherwise, we pick up an edge-value at most $(\frac{q}{d})^2 = \frac{q^2}{d^2}$;

   - Assuming both edges making first two appearances, and if $a_i = b_i$ matches, we pick up a factor of
   $$(1 + o_d(1))\frac{2}{d^2} \cdot md$$

   - Assuming both edges making first two appearances while $a_i \neq b_i$, we pick up a factor of at most
   $$(1 + o_d(1))\frac{2}{d^2} \cdot \sqrt{md^2 q^3}$$

   - If some edge is making third appearance or beyond, we pick up a factor at most
   $$(1 + o_d(1))\frac{q^2}{d^2} \cdot O(q^2\sqrt{dq})$$

   as we have at most one new "circle" vertex that takes label in $[d]$ and gets assigned weight at most $\sqrt{dq}$.

44

- Summing over the above cases gives us a bound of

$$(1 + o_d(1))\frac{2}{d^2} \cdot md = (1 + o_d(1))\frac{2m}{d}$$

Taking the $\frac{1}{q}$-th root and apply Markov's gives us the desired. □

Claim A.2 (Bound of s). *Recall that $s = 1 + \frac{1}{d}\eta^T A^{-1}1_m$, we have*

$$|s| \leqslant 1 + o_d(1)$$

Proof. It suffices for us to bound $|\frac{1}{d}\eta^T A^{-1}1_m| \leqslant o_d(1)$. We adopt the truncation strategy in our main analysis of $\mathcal{R}$, and split it as

$$\frac{1}{d}\eta^T A^{-1}1_m = \frac{1}{d}\eta(T_0 + E)1_m = \frac{1}{d}\eta T_0 1_m + \frac{1}{d}\eta E 1_m$$

where we focus on the analysis of the first term. □

Lemma A.3.

$$B\left(\frac{1}{d}\eta T_0 1_m\right) \leqslant \tilde{O}\left(\frac{1}{\sqrt{d}}\right)$$

Proof. We mimic our analysis for the well-conditionedness of A^{-1}, in particular, each term $\eta T_0 1_m$ is a floating component and moreover, we apply our edge-assignment scheme from analysis of M by traversing from the circle-vertex on one end (for concreteness, take it to be the one from η^T). With some abuse of notation, let this vertex be circ_1 again. Consider the following assignment-scheme,

1. Assign each F, H and non-reverse-charging R to the vertex it leads to;

2. Assign each reverse-charging R step to the destination vertex of the underlying F copy;

3. Assign the normalizing constant of $\frac{1}{d}$ to the starting circle vertex circ_1.

The desired block-value bound then follows immediately by the following proposition.

Proposition A.4. *Each vertex making first/last appearance outside circ_1 is assigned the required (non-surprise) F/R edge-copies. Moreover, there is either an extra edge-copy of factor $\tilde{O}(\frac{1}{\sqrt{d}})$, or circ_1 is making first/last but not both appearances while it is assigned a factor of $\frac{1}{d}$.*

Proof. The first part follows immediately from our analysis of M, that we assign each vertex's first factor to the edge that discovers it, and if the vertex is making the last appearance in the same block, the factor is assigned by the R step of the edge that discovers it. To see that the assignment of last appearance is using R edge alone when the first, last appearances are not in the same block, note that each vertex is assigned the first edge "locally" that discovers the first vertex. In the case that this edge is non-R, the edge must be closed later on and we can swap the assignment of the underlying R copy of that edge. The swapping factor is valid as the R copy of the edge is intended for protecting the vertex's both first and last appearance in the current block, which is again not needed in this case.

To see the second part of the claim, note that the traversal path of the floating component starts from a circle vertex while ends at a square vertex. On the one hand, note that it is immediate if

the first circle vertex $circ_1$ is making first/last but not both appearances in the current block as it is assigned the normalizing constant of $\frac{1}{d}$, giving a gap of $\frac{1}{\sqrt{d}}$. On the other hand, if the $circ_1$ is making both first and last appearances, there are two cases,

1. The final departure from $circ_1$ is a reverse-charging step, in this case, there is a surprise visit within this block, and it gives an extra copy of edge-value $\tilde{O}(\frac{1}{\sqrt{d}})$;

2. The final departure from $circ_1$ is not reverse-charging, this edge either makes a middle appearance before as an H-step, or its underlying F copy is a surprise visit arriving at $circ_1$. In either case, this gives an extra edge-copy of factor $\tilde{O}(\frac{1}{\sqrt{d}})$.

☐

☐

B   Analysis of M_D

Recall from Proposition 2.3 that $M_D[i,i] = \|v_i\|_2^4 - \frac{2}{d}\|v_i\|_2^2 - 1$. For ease of technical analysis, it is helpful to note here that we can further decompose M_D to be the following matrices:

Proposition B.1 (Decomposition of M_D). *We write*

$$M_D = M_{D,1} + M_{D,2} + \left(2 + \frac{2}{d}\right) M_{D,3},$$

where for any $i \in [m]$,

1. $M_{D,1}[i,i] = \sum_{a \neq b \in [d]} \left(v_i[a]^2 - \frac{1}{d}\right)\left(v_i[b]^2 - \frac{1}{d}\right)$;

2. $M_{D,2}[i,i] = \sum_{a \in [d]}\left(v_i[a]^4 - \frac{6}{d}v_i[a]^2 + \frac{3}{d^2}\right)$;

3. $M_{D,3}[i,i] = \sum_{a \in [d]}\left(v_i[a]^2 - \frac{1}{d}\right)$.

Proof. $M_D[i,i] = \|v_i\|_2^4 - \frac{2}{d}\|v_i\|_2^2 - 1$. We first unpack $\|v_i\|_2^4$: $\|v_i\|_2^4 = (\sum_{a \in [d]} v_i[a]^2)^2 = \sum_a v_i[a]^4 + \sum_{a \neq b} v_i[a]^2 v_i[b]^2$. On the one hand, for any $a \neq b \in [d]$,

$$v_i[a]^2 v_i[b]^2 = \left(\left(v_i[a]^2 - \frac{1}{d}\right) + \frac{1}{d}\right) \cdot \left(\left(v_i[b]^2 - \frac{1}{d}\right) + \frac{1}{d}\right),$$

thus,

$$\sum_{a \neq b \in [d]} v_i[a]^2 v_i[b]^2 = \underbrace{\sum_{a \neq b \in [d]}\left(\left(v_i[a]^2 - \frac{1}{d}\right)\left(v_i[b]^2 - \frac{1}{d}\right)\right)}_{M_{D,1}} + \underbrace{2 \cdot \frac{d-1}{d}\sum_a\left(v_i[a]^2 - \frac{1}{d}\right)}_{(2-\frac{2}{d})M_{D,3}} + \frac{d(d-1)}{d^2}.$$

On the other hand,

$$\sum_{a \in [d]} v_i[a]^4 = \sum_{a \in [d]}\left(\left(v_i[a]^4 - \frac{6v_i[a]^2}{d} + \frac{3}{d^2}\right) + \frac{6}{d}\left(v_i[a]^2 - \frac{1}{d}\right) + \frac{3}{d^2}\right)$$

46

$$= \sum_{a \in [d]} \left(v_i[a]^4 - \frac{6v_i[a]^2}{d} + \frac{3}{d^2} \right) + \frac{6}{d} \underbrace{\sum_{a \in [d]} \left(v_i[a]^2 - \frac{1}{d} \right)}_{\frac{6}{d} M_{D,3}} + \frac{3}{d} \cdot$$

Thus, $\|v_i\|_2^4 = M_{D,1} + M_{D,2} + (2 + \frac{4}{d})M_{D,3} + 1 + \frac{2}{d}$.

Next, observe that $\|v_i\|_2^2 = M_{D,3}[i,i] + 1$. Thus, we have

$$M_D[i,i] = \|v_i\|_2^4 - \frac{2}{d}\|v_i\|_2^2 - 1 = M_{D,1} + M_{D,2} + \left(2 + \frac{2}{d}\right)M_{D,3},$$

completing the proof. $\qquad \square$

Lemma B.2 (Norm bound for M_D). *For matrices $M_{D,1}, M_{D,2}, M_{D,3}$ defined in [Proposition B.1](#), we have the following bounds*

1. $B(M_{D,1}) \leqslant \tilde{O}(\frac{1}{d^2} \cdot \sqrt{d^2}) = \tilde{O}(\frac{1}{d})$;

2. $B(M_{D,2}) \leqslant \tilde{O}(\frac{\sqrt{d}}{\sqrt{d^8}})$;

3. $B(M_{D,3}) \leqslant \tilde{O}(\frac{\sqrt{d}}{d}) = \tilde{O}(\frac{1}{\sqrt{d}})$.

As a consequence, $\|M_D\|_{\mathrm{op}} \leqslant \tilde{O}(\frac{1}{\sqrt{d}}) = o_d(1)$ with probability $1 - 2^{-q/\log d}$.

Proof. We observe the following,

1. The square vertex is in $U \cap V$ and hence bound to be making a middle appearance (ignoring the first and the last block of the walk as they are offset by long-path);

2. Each circle vertex is either making a first/middle/last appearance, and note that instead of keeping track of Pur factor, it suffices for us to assign it a cost of at most $\sqrt{n} \cdot (q \cdot D_V)$;

3. For $M_{D,1}$, each edge gets assigned a factor of at most $\frac{q}{d}$, and we have at most two circle vertices outside making first/middle/last appearances, which give a bound of $O(\frac{q^2}{d^2} \cdot \sqrt{d^2}(q \cdot D_V)^2)$.

4. $M_{D,3}$ is identical to $M_{D,1}$ except there is only one circle vertex outside, which gives a bound of $O(\frac{q}{d} \cdot \sqrt{d} \cdot q \cdot D_V)$.

5. For $M_{D,2}$, each edge gets assigned a factor of at most $\frac{q}{\sqrt{d^8}}$, and we have one single circle vertex outside, which combines to a bound of $O(\frac{q}{\sqrt{d^8}} \cdot \sqrt{d} \cdot (q \cdot D_V))$.

Finally, recalling that we set $q^{O(1)} \cdot D_V \ll \sqrt{d}$ completes the proof to our lemma. $\qquad \square$

C   Sketch of an Alternative Analysis

Definition C.1. For each chain, we define local F, R, S, and H edges as follows. Here we traverse the chain from top to bottom.

1. We say that an edge is a local F edge if it is appearing for the first time in the current chain and its destination is appearing for the first time in the current chain.

2. We say that an edge is a local R edge if it is appearing for the last time in the current chain.

3. We say that an edge is a local H edge if it appears again both before and afterwards in the current chain.

4. We say that an edge is a local S edge if it is appearing for the first time but its destination has appeared before in the current chain.

For our local analysis, there are two circle vertices which we may need to be careful about.

Definition C.2. Let v_{left} be the circle vertex which is at the top left, let v_{right} be the circle vertex which is at the top right (which may be equal to v_{left}), and let v_{bottom} be the circle vertex which is at the bottom.

We can always take the minimum weight vertex separator to be v_{left}, so we do not need to worry about v_{left}. However, as we will see, we have to be careful about v_{right} and v_{bottom}.

Definition C.3. We say that an edge is locally vanishing if we have that after the local intersections, the product of it and the edges parallel to it has a nonzero constant term.

Example C.4. Some examples of locally vanishing edges are as follows.

1. Two parallel edges with label 1 are locally vanishing as $x^2 = \sqrt{2}\frac{x^2-1}{\sqrt{2}} + 1$

2. Two parallel edges with label 2 are locally vanishing as

$$\left(\frac{x^2-1}{\sqrt{2}}\right)^2 = \sqrt{6}\left(\frac{x^4 - 6x^2 + 3}{\sqrt{24}}\right) + 2\sqrt{2}\left(\frac{x^2-1}{\sqrt{2}}\right) + 1$$

3. More generally, two parallel edges are locally vanishing if they have the same label k and are not locally vanishing if they have different labels.

4. Three parallel edges with labels 1, 1, and 2 are locally vanishing as

$$x^2\left(\frac{x^2-1}{\sqrt{2}}\right) = 2\sqrt{3}\frac{x^4 - 6x^2 + 3}{\sqrt{24}} + 5\frac{x^2-1}{\sqrt{2}} + \sqrt{2}$$

Definition C.5. We say that a vertex v is locally isolated if v is not equal to the top left circle or the top right circle and all edges incident to v are locally vanishing.

For all edges except for two special edges e_r^* and e_b^* which we describe below, we assign them as follows.

1. For each local F edge, we assign it to its destination. For the edge at the top between v_{right} and its neighboring square vertex, we consider its destination to be the square vertex.

2. For each local R edge, we assign it to its origin unless it is a dangling edge with label 2. If it is a dangling edge with label 2, we split it between its endpoints.

3. For each local S and H edge, we assign half of it as a bonus to its origin and half of it as a bonus to its destination.

Lemma C.6. *All vertices except v_{right} and v_{bottom} have the required edge factors.*

Proof. For each circle vertex u except for v_{left} (which we doesn't need any edges), v_{right}, and v_{bottom}, consider the first and last time u appears in the current chain. The first time u appears, there must be a local F edge pointing to it which gives u one edge. If u only appears once, it cannot be locally isolated, so it only needs one edge. Otherwise, the last time u appears, it can only be locally isolated if all edges incident to it are R edges. In this case, u obtains a second edge.

Similarly, for each square vertex v, consider the first and last time v appears in the current chain. The first time v appears, there must be two local F edges with label 1 or one local F edge with label 2 pointing to it. The last time v appears, it can only be locally isolated if all edges incident to it are R edges, in which case v obtains two additional edges. $\square$

Remark C.7. Note that v_{right} is an exception because the first time it appears, it does not have a local F edge pointing to it. Similarly, v_{bottom} is an exception because the last time it appears, it may be incident to an R edge with label 2 without gaining an additional edge.

Definition C.8. When $v_{left} \neq v_{right}$ and v_{right} is not equal to the final circle vertex, we find/define e_r^* through the following iterative process. We start with the edge e between v_{right} and its neighbor. Note that this is a local F edge going down from a vertex $v = v_{right}$ which is not locally isolated and which is not the final circle vertex. We now have the following cases

1. e does not appear again and the other endpoint of e is the final circle vertex. In this case, we take $e_r^* = e$ and assign one edge factor from it to v_{right}. If it has label 2, we keep one edge factor in reserve in case $e_b^* = e_r^* = e$.

2. e does not appear again and its other endpoint is not the final circle vertex. In this case, letting v' be the other endpoint of e, we take e' to be an edge going down from v'. If e' is not a local F edge, we take $e_r^* = e'$ and assign it to v_{right}. We can do this because v' is not locally isolated so it already has all of the edge factors it needs. If e' is a local F edge, we again have a local F edge e' going down from a vertex which is not locally isolated and is not the final circle vertex, so we repeat this process.

3. e appears again. In this case, let e' be the next time e appears. If e' is not a local R edge whose destination is v, we take $e_r^* = e'$ and assign it to v_{right}. If e' is a local R edge whose destination is v, we let e'' be an edge going down from v.

   If e'' is not a local F edge then we take $e_r^* = e''$ and assign it to v_{right}. If e'' is a local F edge then we are still in the situation where we have a local F edge going down from a vertex which is not locally isolated and is not equal to the final circle vertex, so we repeat this process.

Definition C.9. We find/define e_b^* through the following iterative procedure. We start with the edge e with label 2 between v_{bottom} and its neighbor. Note that e must be a local R edge. We then do the following.

1. If e is not a locally vanishing edge then we take $e_b^* = e$. We assign half of $e_b^* = e$ to v_{bottom} and keep half in reserve in case $e_b^* = e_r^*$.

2. If e is a locally vanishing edge then e must appear above. Consider the previous time e appears and let e' be this copy of e. There are a few possibilities for e'.

   (a) e' is an edge with label 2 going from a copy of v_{bottom} to the bottom square vertex of the current block which is not equal to the top vertex of the current block. If e' is not a local F edge then we take $e_b^* = e'$. We assign half of $e_b^* = e$ to v_{bottom} and keep half in reserve in case $e_b^* = e_r^*$. If e' is a local F edge, let e'' be the edge from the top square vertex of the current block to the copy of v_{bottom}. If e'' is not a local R edge then we take $e_b^* = e''$. We assign half of $e_b^* = e''$ to v_{bottom} and keep half in reserve in case $e_b^* = e_r^*$. If e'' is a local R edge then we are in the same situation as before so we repeat this process.

   (b) e' is an edge with label 2 from the top square vertex in the current block which does not equal the bottom square vertex of the current block to a copy of v_{bottom}. In this case, we take $e_b^* = e'$. If $e_b^* = e'$ is not a local F edge then we assign half of it to v_{bottom} and keep half in reserve in case $e_b^* = e_r^*$. If $e_b^* = e'$ is a local F edge then we assign it to v_{bottom}. Note that in this case, we cannot have that $e_b^* = e_r^*$. The reason for this is that e_r^* can only be a local F edge in case 1, in which case it does not appear again (while e' appears again by definition).

   (c) e' is an edge with label 2 which hangs off of a square vertex which is both the top and bottom square vertex for the current block. In this case, we take $e^* = e'$ and assign all of it to v_{bottom}. Note that in this case, we cannot have that $e_b^* = e_r^*$.

   (d) e' is a local H edge with label 1. In this case, we take $e_b^* = e'$ and assign it to v_{bottom}. Here we may have that $e_b^* = e_r^*$ but if we do, one of the endpoints of $e_b^* = e'$ is not locally isolated so we can take an edge factor from it and give the edge factor to v_{right}.