

# Certifying Euclidean Sections and Finding Planted Sparse Vectors Beyond the $\sqrt{n}$ Dimension Threshold

Venkatesan Guruswami  
Department of EECS  
UC Berkeley  
Berkeley, USA  
venkatg@berkeley.edu

Jun-Ting Hsieh  
Computer Science Department  
Carnegie Mellon University  
Pittsburgh, USA  
juntingh@cs.cmu.edu

Prasad Raghavendra  
Department of EECS  
UC Berkeley  
Berkeley, USA  
raghavendra@berkeley.edu

**Abstract**—We consider the task of certifying that a random  $d$ -dimensional subspace  $X$  in  $\mathbb{R}^n$  is well-spread — every vector  $x \in X$  satisfies  $c\sqrt{n}\|x\|_2 \leq \|x\|_1 \leq \sqrt{n}\|x\|_2$ . In a seminal work, Barak et. al. [3] showed a polynomial-time certification algorithm when  $d \leq O(\sqrt{n})$ . On the other hand, when  $d \gg \sqrt{n}$ , the certification task is information-theoretically possible but there is evidence that it is computationally hard [10], [39], a phenomenon known as the information-computation gap.

In this paper, we give subexponential-time certification algorithms in the  $d \gg \sqrt{n}$  regime. Our algorithm runs in time  $\exp(\tilde{O}(n^\epsilon))$  when  $d \leq \tilde{O}(n^{\frac{1+\epsilon}{2}})$ , establishing a smooth trade-off between runtime and the dimension.

Our techniques naturally extend to the related planted problem, where the task is to recover a sparse vector planted in a random subspace. Our algorithm achieves the same runtime and dimension trade-off for this task.

**Index Terms**—Euclidean Section, Subexponential-time Algorithm

## I. INTRODUCTION

For any vector  $x \in \mathbb{R}^n$ , we know that  $\|x\|_2 \leq \|x\|_1 \leq \sqrt{n}\|x\|_2$ . Intuitively, if the upper bound is approximately tight, i.e.,  $\|x\|_1 \geq c\sqrt{n}\|x\|_2$ , then  $x$  is “dense” or “incompressible”. We say that a subspace  $X \subseteq \mathbb{R}^n$  is *well-spread* if every  $x \in X$  is dense. More formally, we define the *distortion* of  $X$ , denoted  $\Delta(X)$ , as follows,

$$\Delta(X) := \sup_{x \in X: x \neq 0} \frac{\sqrt{n}\|x\|_2}{\|x\|_1}. \quad (1)$$

The distortion always satisfies  $1 \leq \Delta(X) \leq \sqrt{n}$ . Subspaces with large dimension  $d$  and small distortion are called good *Euclidean sections* of  $\ell_1^n$ . In particular, they provide embeddings of  $\ell_2^d$  into  $\ell_1^n$  with only a small blow-up in the dimension, thus such subspaces have many applications including error-correcting codes over the reals [9], [23], compressed sensing [15], [34], high-dimensional nearest-neighbor search [29], and oblivious regression [17].

Therefore, there has been a long line of work on constructing good Euclidean sections, including explicit

constructions [22], [29], [30] and constructions requiring few random bits (a.k.a. partially derandomized constructions) [23], [31], [38]. However, all such constructions suffer in either the dimension or the distortion (see [31] and references therein).

On the other hand, it is well known that a fully random subspace of  $\mathbb{R}^n$  of dimension  $d = \Omega(n)$  (for instance, the column space of an  $n \times d$  matrix with i.i.d. Gaussian or  $\pm 1$  entries) has  $O(1)$  distortion with high probability [18], [20], [35]. However, randomized constructions suffer from the drawback that the output is not guaranteed to *always* satisfy the desired properties. Thus, *certifying* randomized constructions is often considered as a functional proxy when explicit constructions are hard to come by, as is the case for Euclidean sections.

This motivates the algorithmic task of certifying that a random subspace is well-spread. With  $\exp(d)$  time, this problem is trivial as we can brute-force search over an  $\epsilon$ -net in  $\mathbb{R}^d$ . The best known non-trivial result is by Barak et. al. [3], who showed that given a random matrix  $A \in \mathbb{R}^{n \times d}$  with i.i.d. Gaussian entries and  $d \lesssim \sqrt{n}$ , there is a polynomial-time algorithm certifying that  $\text{colspan}(A)$  has  $O(1)$  distortion (with high probability over  $A$ )<sup>1</sup>. However, no efficient certification algorithm is known when  $d \gg \sqrt{n}$  (note that the problem is harder for larger  $d$ ).

More recently, “evidence” of computational hardness was established for the  $d \gg \sqrt{n}$  regime in the form of lower bounds against low-degree polynomials [10], [39]. This suggests that there is an *information-computation gap* for this problem with the computational threshold at  $d \sim \sqrt{n}$ , i.e., any polynomial-time algorithm for  $d \gg \sqrt{n}$  would require significant breakthroughs (see e.g. [24], [37] for expositions of the low-degree hardness framework).

The lower bounds by [10], [39] leave open the possibil-

<sup>1</sup> This was implicitly proved in [3] and the result holds generally for matrices with sub-gaussian entries. An explicit proof was given in [10].

ity of *subexponential*-time certification algorithms when  $d$  is between  $\sqrt{n}$  and  $n$ . This sets the stage for our first result.

**Theorem 1** (Informal Theorem III.3). Fix  $\varepsilon \in (0, 1)$ , and let  $d, n \in \mathbb{N}$  such that  $d = O(n^{\frac{1+\varepsilon}{2}} / \log n)$ . Let  $A \sim \mathcal{N}(0, 1)^{n \times d}$ . Then, there is a certification algorithm that runs in  $2^{\tilde{O}(n^\varepsilon)}$  time and, with probability  $1 - o(1)$  over  $A$ , certifies that  $\text{colspan}(A)$  has  $O(1)$  distortion.

Notice that Theorem 1 establishes a smooth trade-off between the dimension  $d$  and the runtime. Specifically, when  $\varepsilon = 0$ , we have  $d \leq \tilde{O}(\sqrt{n})$  and the runtime is polynomial, matching the best known algorithm by [3]. When  $\varepsilon$  increases to 1, the dimension  $d$  increases to  $n$  while the runtime increases to exponential.

We note that the phenomenon of information-computation gap is widespread across a variety of certification and inference problems. For these problems, there is a parameter regime (often called the “hard” regime) in which it is conjectured that no polynomial-time algorithm exists. In many cases, smooth trade-offs were established between runtime and the problem parameters, similar to that of Theorem 1. Examples of such trade-offs include runtime vs. the number of constraints in refuting random constraint satisfaction problems [21], [26], [42], runtime vs. the approximation factor in polynomial optimization or tensor PCA [7], [27], [46], runtime vs. the sparsity parameter in certifying the restricted isometry property (RIP) for random matrices [13], [36], and runtime vs. the sparsity of the hidden vector in sparse PCA [14], [16].

a) *Finding planted sparse vector in a random subspace.*: For average-case optimization problems, techniques for certification can often be adapted to solving the *search* problem for the related *planted* model; this is called the *proofs-to-algorithms* paradigm [19] in the literature. For the problem of certifying distortion of a random subspace (in which there is no sparse vector), the corresponding planted problem is to find a *sparse* vector planted in a random subspace.

We first formally define the planted sparse vector problem:

**Model 1** (Planted sparse vector problem). Fix an unknown unit vector  $v \in \mathbb{R}^n$ , and let  $d \leq n \in \mathbb{N}$ . Let  $\tilde{A}$  be a random  $n \times d$  matrix sampled as follows: (1) let  $A$  be the random matrix such that the first column is  $v$  and the other  $d - 1$  columns are i.i.d.  $\mathcal{N}(0, \frac{1}{n} \mathbb{I}_n)$  vectors; (2) let  $R \in \mathbb{R}^{d \times d}$  be an arbitrary unknown rotation matrix; (3) set  $\tilde{A} = AR$ .

The task is that given  $\tilde{A}$ , output a unit vector  $\hat{v} \in \mathbb{R}^n$  such that  $\langle \hat{v}, v \rangle^2 \geq 1 - o(1)$ .

For concreteness and comparison to prior works, we will focus on the special case where  $v$  has noisy Bernoulli-Rademacher entries:

**Definition I.1** (Noisy Bernoulli-Rademacher distribution [10]). Given parameter  $\rho \in (0, 1)$  and  $\sigma \in [0, 1/\sqrt{1-\rho})$ , we define  $\text{nBR}(\rho, \sigma)$  to be the random variable such that

$$x = \begin{cases} \mathcal{N}(0, \sigma^2/n) & \text{with probability } 1 - \rho, \\ +\frac{1}{\sqrt{\rho'n}} & \text{with probability } \rho/2, \\ -\frac{1}{\sqrt{\rho'n}} & \text{with probability } \rho/2, \end{cases}$$

where  $\rho' := \frac{\rho}{1-(1-\rho)\sigma^2}$ .

The parameter  $\rho'$  is set such that  $\mathbb{E}_{x \sim \text{nBR}(\rho, \sigma)}[x^2] = 1/n$ , so with high probability a vector  $v \sim \text{nBR}(\rho, \sigma)^n$  will have  $\|v\|_2 \in (1 \pm o(1))$ . For concreteness, one can think of  $\rho, \sigma$  as  $\frac{1}{\text{polylog}(n)}$ . This distribution is used in the hardness result of [10], and is the noisy version of the “noiseless” Bernoulli-Rademacher distribution considered in [12], [39], [47]<sup>2</sup> and used as the “spike” in related sparse recovery problems like sparse PCA [2], [14], [16], [32].

The problem of finding the planted sparse vector was introduced by Spielman, Wang, and Wright [43] in the context of dictionary learning. This problem has since received a lot of attention due to various applications in learning theory and optimization (see e.g. [11]). Barak et. al. [4] was the first example of the proofs-to-algorithms paradigm for this problem. Specifically, their algorithm uses the certification algorithm from [3] as a main ingredient, and thus it works when  $d \ll \sqrt{n}$  (same as the polynomial-time regime for certification). Various other algorithms have been proposed [11], [12], [39], [41], [47], and the current best known algorithm is by Mao and Wein [39] which succeeds when  $\rho d \ll \sqrt{n}$  (rather than  $d \ll \sqrt{n}$  required in [4], [25]).

On the other hand, for  $d \gg \sqrt{n}$  (the hard regime for certification), the lower bounds by [10], [39] mentioned earlier also apply to the planted problem, as their hardness results are for the (easier) *detection* problem. Specifically, Chen and d’Orsi [10] showed that when  $\rho d \geq \tilde{\Omega}(\sqrt{n})$ , all  $\text{polylog}(n)$ -degree polynomials fail to distinguish between  $A \sim \mathcal{N}(0, 1/n)^{n \times d}$  and  $\tilde{A}$  sampled from Model 1 with planted vector  $v \sim \text{nBR}(\rho, \sigma)^n$  for  $\rho \gg 1/n$  and  $\sigma \leq 1/\text{polylog}(n)$ .

Thus, it is natural to consider adapting our techniques for subexponential-time certification to the planted problem in the hard regime. This is our second result.

**Theorem 2** (Informal Theorem IV.2). Let  $t \leq d \leq n \in \mathbb{N}$  and  $\rho \leq \frac{1}{\text{polylog}(n)}$  such that  $t \geq \frac{\rho d^2}{n} \text{polylog}(n)$ . There is a randomized algorithm with running time  $2^{\tilde{O}(t)}$  with the following guarantee: Given  $\tilde{A} \in \mathbb{R}^{n \times d}$  drawn from

<sup>2</sup> Surprisingly, [12], [47] showed that one can recover  $v$  even when  $n = d + 1$  if  $v$  is a *noiseless* Bernoulli-Rademacher vector, but their algorithms break down if one adds an inverse-polynomial amount of noise to  $v$ .

Model 1 with planted vector  $v \sim \text{nBR}(\rho, \frac{1}{\text{polylog}(n)})^n$ , with high probability, the algorithm outputs a unit vector  $\hat{v} \in \text{colspan}(\tilde{A})$  such that  $\langle v, \hat{v} \rangle^2 \geq 1 - o(1)$ .

For intuition, consider parameters  $\rho = 1/\text{polylog}(n)$  and  $d = \tilde{O}(n^{\frac{1+\varepsilon}{2}})$  for  $\varepsilon \in (0, 1)$ . Then, the algorithm recovers  $v$  in  $2^{\tilde{O}(n^\varepsilon)}$  time, the same dimension vs. runtime trade-off as established in Theorem 1. Our algorithm in fact works for more general sparse vectors  $v$ ; see Theorem IV.2 and Remark IV.3 for details and discussions on the assumptions of  $v$  that we require.

**Remark I.2.** One can consider the harder planted problem where you are given an arbitrary orthogonal basis of the subspace instead of a “Gaussian basis” like Model 1. In fact, both [25] and [39] started with Model 1, then with some extra work, they showed that the algorithms are robust to exchanging the Gaussian basis for an arbitrary orthogonal basis. We leave this as a future work, though we believe similar techniques (like matrix perturbation analysis) used in [25], [39] may apply to our case as well. On the other hand, all hardness results [10], [39] are proved for Model 1.

## II. TECHNICAL OVERVIEW

*a) Notation.* For an integer  $N$ , we will use  $[N]$  to denote the set  $\{1, 2, \dots, N\}$ . For a vector  $x \in \mathbb{R}^n$ , we use  $\|x\|_p$  to denote its  $\ell_p$  norm, and for any  $S \subseteq [n]$ , we write  $x_S \in \mathbb{R}^{|S|}$  to denote the vector restricted to coordinates in  $S$ . For a matrix  $M \in \mathbb{R}^{n \times d}$ , we use  $\|M\|_2$  to denote its operator (spectral) norm:  $\sup_{x \neq 0} \frac{\|Mx\|_2}{\|x\|_2}$ . Moreover, for any  $S \subseteq [n]$ , we write  $M_S \in \mathbb{R}^{|S| \times d}$  to denote the submatrix of  $M$  obtained by selecting rows according to  $S$ .

*b) Organization.* Section II-A describes the certification algorithm for  $d \leq O(\sqrt{n})$  by [3] using the 2-to-4 norm of  $A$ . Section II-B explains the barrier of going beyond  $\sqrt{n}$  as well as our strategy to bypass it by removing outlier entries of  $Ax$ . This motivates a key ingredient in our analysis, which is the elementary symmetric polynomial  $P_t(x)$  explained in Section II-C. In Section II-D, we make a brief detour to explain the trace moment method and the analysis of [44]. In Section II-E, we upper bound  $P_t(x)$  by bounding the spectral norm of a related random matrix (which we can certify) using the trace method, and explain the crucial idea that symmetrization lowers the spectral norm. Finally, in Section II-F, we give an overview of how we adapt our techniques to the planted problem.

*A. 2-to-4 norm as a proxy for sparsity: certification algorithm of [3]*

Consider a matrix  $A \in \mathbb{R}^{n \times d}$  with i.i.d.  $\mathcal{N}(0, 1)$  entries (where  $d \ll n$ ), and let  $a_1, a_2, \dots, a_n \in \mathbb{R}^d$  be its rows. To certify that  $\text{colspan}(A)$  has small distortion (as defined in Eq. (1)), one needs to certify that  $Ax$  is “sparse” —

$\|Ax\|_2 / \|Ax\|_1 \leq O(1/\sqrt{n})$  — for all  $x \in \mathbb{R}^d$ . It is well known that  $A$  has singular values between  $\sqrt{n}(1 \pm o(1))$  with high probability (see Fact III.5), so  $\|Ax\|_2 \approx \sqrt{n}$  for all unit vectors  $x$ . On the other hand, as intuition, consider a random unit vector  $x$ . Each  $|\langle a_i, x \rangle|$  is roughly  $\Omega(1)$ , hence  $\|Ax\|_1 = \sum_{i=1}^n |\langle a_i, x \rangle| \geq \Omega(n)$ . So, for a random  $x$  we have  $\|Ax\|_2 / \|Ax\|_1 \leq O(1/\sqrt{n})$ , as desired.

The difficult part is to certify a lower bound on  $\|Ax\|_1$  for *all* unit vectors  $x$ . Thus, in many applications, it is more tractable to consider an alternative proxy for sparsity — the 2-to-4 norm. Intuitively, vectors with small 4-norm compared to the 2-norm are considered well-spread. The quantity  $\max_{x \neq 0} \|Ax\|_4 / \|x\|_2$ , denoted  $\|A\|_{2 \rightarrow 4}$ , is called the 2-to-4 norm of  $A$ , and upper bounds on  $\|A\|_{2 \rightarrow 4}$  are called *hypercontractivity* inequalities.

Barak et. al. [3, Theorem 7.1] showed that with high probability,

$$\|A\|_{2 \rightarrow 4}^4 \leq n \cdot \left( 3 + O(1) \cdot \max\left(\frac{d}{\sqrt{n}}, \frac{d^2}{n}\right) \right).$$

Moreover, this can be efficiently certified via a natural SDP relaxation for maximizing  $\|Ax\|_4^4 = \sum_{i=1}^n \langle a_i, x \rangle^4$ , a degree-4 polynomial. Therefore, when  $d \leq O(\sqrt{n})$ , we have that  $\sum_{i=1}^n \langle a_i, x \rangle^4 \leq O(n)$  for all unit vectors  $x$ . Note that this bound matches the case when most  $|\langle a_i, x \rangle|$  are roughly  $\Theta(1)$ , which is the case for a random  $x$ . Combined with standard tools (Lemma III.2 and [10, Proposition 3.4]) and the fact that  $\|Ax\|_2 \approx \sqrt{n}$ , this implies that  $\text{colspan}(A)$  has  $O(1)$  distortion.

We now give a brief overview of the proof by [3], which is our starting point. The main idea is to write  $\|Ax\|_4^4$  as follows,

$$\|Ax\|_4^4 = \sum_{i=1}^n \langle a_i, x \rangle^4 = (x^{\otimes 2})^\top \left( \sum_{i=1}^n a_i^{\otimes 4} \right) (x^{\otimes 2}),$$

where we view  $x^{\otimes 2}$  as a  $d^2$ -dimensional vector and  $\sum_{i=1}^n a_i^{\otimes 4}$  as a  $d^2 \times d^2$  random matrix. The main idea of [3] is that by standard matrix concentration inequalities (like the Hanson-Wright inequality [45] and results from [1]),  $\sum_{i=1}^n a_i^{\otimes 4}$  concentrates around its mean  $\mathbb{E} \sum_{i=1}^n a_i^{\otimes 4} = n\Sigma$  where  $\Sigma = \mathbb{E}[a_i^{\otimes 4}]$ , again viewed as a  $d^2 \times d^2$  matrix.

The naive approach is to bound the operator norm  $\|\Sigma\|_2$ , then we have  $\|Ax\|_4^4 \leq n \cdot \|\Sigma\|_2$ . Unfortunately, there is a rank-1 component  $ww^\top$  in  $\Sigma$  where  $w_{ij} = \delta_{ij}$ , and  $\|ww^\top\|_2 = n$ . The crucial observation is to “shift” the entries of  $\Sigma$  using the symmetry of  $x^{\otimes 2}$  to “break” the rank-1 component. Then, the shifted matrix  $\text{Shift}(\Sigma)$  satisfies  $(x^{\otimes 2})^\top \Sigma (x^{\otimes 2}) = (x^{\otimes 2})^\top \text{Shift}(\Sigma) (x^{\otimes 2}) \leq \|\text{Shift}(\Sigma)\|_2 \leq O(1)$ . This implies that  $\|Ax\|_4^4 \leq O(n)$ , as desired. As we will explain later, this symmetrization technique to decrease the matrix norm is central to our analysis and is also the key idea behind the results of [7], [8], [42].

### B. Going beyond $\sqrt{n}$

When  $d \gg \sqrt{n}$ , we face the immediate problem that  $\|A\|_{2 \rightarrow 4}^4 \leq n$  is no longer true. One can easily see this by considering  $x = a_1/\|a_1\|_2$ . In this case, we have  $\langle a_1, x \rangle = \|a_1\|_2 \approx \sqrt{d}$  since  $a_1$  is a  $d$ -dimensional Gaussian vector, and  $|\langle a_i, x \rangle| \approx \Theta(1)$  for  $i \neq 1$ . Then,  $\|Ax\|_4^4 = \sum_{i=1}^n \langle a_i, x \rangle^4 \geq \Omega(d^2 + n) = \Omega(d^2) \gg n$ .

More generally, for any  $S \subseteq [n]$  of size  $t \ll d$ , the submatrix  $A_S \in \mathbb{R}^{t \times d}$  has maximum singular value  $\approx \sqrt{d}$ , i.e., there is an  $x$  such that  $\|A_S x\|_2^2 \approx d$ . Suppose the vector  $A_S x$  is roughly equally distributed, then we have  $\|A_S x\|_4^4 \geq (\frac{d}{t})^2 t = \frac{d^2}{t}$ , which means that  $\|Ax\|_4^4 \geq \frac{d^2}{t}$ .

**Removing the top entries of  $Ax$ .** These counterexamples give an important insight:  $\|Ax\|_4^4$  seems to be dominated by just a few very large entries in  $Ax$ . In the setting of Theorem 1, we have  $d = n^{\frac{1+\epsilon}{2}}$ , and in the example above,  $\|Ax\|_4^4 \geq \frac{d^2}{t} \gg n$  when  $t \ll n^\epsilon$ . One might guess that after removing the top  $n^\epsilon$  entries of  $Ax$ , the resulting vector has small 4-norm.

We show that this is indeed the case:

**Lemma II.1** (Informal Lemma III.4). *For any unit vector  $x \in \mathbb{R}^d$ , let  $T_x \subseteq [n]$ ,  $|T_x| = n - n^\epsilon$  be the set of indices excluding the top  $n^\epsilon$  entries of  $Ax$ . Then,  $\|A_{T_x} x\|_4^4 \leq O(n)$  with high probability over  $A$ . Moreover, there is an algorithm that certifies this in  $2^{\tilde{O}(n^\epsilon)}$  time.*

Given Lemma II.1, Theorem 1 follows almost immediately. The proof is given in Section III.

The first part that  $\|A_{T_x} x\|_4^4 \leq O(n)$  can in fact be proved via a probabilistic argument. The challenge is to certify it. At a high-level, we need a “proxy” for  $\|Ax\|_4^4$  that is robust to  $t := n^\epsilon$  large entries of  $Ax$  (obviously without knowing where the large entries are because  $T_x$  depends on  $x$ ). In light of this, we turn our attention to the *elementary symmetric polynomials*.

### C. Elementary symmetric polynomials

We define

$$Q_t(z) := t! \sum_{S \subseteq [n]: |S|=t} \prod_{i \in S} z_i^4,$$

and for  $A$  with rows  $a_1, a_2, \dots, a_n \in \mathbb{R}^d$ ,

$$P_t(x) := Q_t(Ax) = t! \sum_{S \subseteq [n]: |S|=t} \prod_{i \in S} \langle a_i, x \rangle^4,$$

where we omit the dependence on  $a_1, \dots, a_n$  for simplicity.

The scaling  $t!$  here is because  $Q_t(z)$  is exactly the “multilinear” terms when one expands out  $\|z\|_4^{4t} = (\sum_{i=1}^n z_i^4)^t$ , and thus our target upper bound is  $O(n)^t$ . Moreover,  $P_t(x)$  can be computed in  $n^{O(t)}$  time.

The following is our key lemma.

**Lemma II.2** (Informal Lemma III.6). *With high probability over  $A$ , there is an algorithm that runs in  $n^{O(t)}$  time and certifies that for all unit vectors  $x \in \mathbb{R}^d$ ,*

$$P_t(x) \leq O\left(n + \frac{d^2}{t} \log^2 n\right)^t.$$

When  $d \leq \tilde{O}(n^{\frac{1+\epsilon}{2}})$  and  $t = n^\epsilon$ , we get  $P_t(x) \leq O(n)^t$ . This is what we expect for a typical  $x$ , where most  $|\langle a_i, x \rangle|$  are  $O(1)$  and  $P_t(x) \approx n^t \cdot O(1)^{4t} = O(n)^t$ .

A careful reader may notice that an upper bound on  $Q_t(z)$  does not imply that the vector  $z$  is dense. For example,  $Q_t(z) = 0$  whenever  $z$  has less than  $t$  nonzero entries. Thus, Lemma II.2 does not immediately imply that  $Ax$  is dense for all  $x$ . However, notice that we are allowed  $n^{O(t)}$  time, so we can exhaustively search over all subsets  $S \subseteq [n]$  of size  $t$  and “check” that no  $Ax$  is concentrated on  $t$  coordinates. This comes down to computing the maximum singular value of  $A_S$  for each  $S$ .

The overall analysis (proof of Lemma II.1 from Lemma II.2) is as follows. For any unit vector  $x \in \mathbb{R}^d$ , since  $T_x$  is the set with the  $t$  largest  $\langle a_i, x \rangle^4$ , and  $\|A_{T_x} x\|_2^2 \leq \sigma_{\max}(A_{T_x})^2 \leq d(1 + o(1))$  verified by exhaustive search, it follows that  $\langle a_i, x \rangle^2 \leq \frac{d}{t}(1 + o(1)) := \tau$  for all  $i \in T_x$ . Then, we consider  $\|A_{T_x} x\|_4^{4t} = (\sum_{i \in T_x} \langle a_i, x \rangle^4)^t$ . For the multilinear term in the expansion of  $\|A_{T_x} x\|_4^{4t}$ , we have

$$t! \sum_{S \subseteq T_x: |S|=t} \prod_{i \in S} \langle a_i, x \rangle^4 \leq t! \sum_{S \subseteq [n]: |S|=t} \prod_{i \in S} \langle a_i, x \rangle^4 = P_t(x) \leq O(n)^t.$$

Note that here we changed the summation of  $S \subseteq T_x$  to  $S \subseteq [n]$  (since  $T_x \subseteq [n]$  and all terms are non-negative). Crucially, we get an upper bound that does not depend on  $T_x$ , thus allowing us to use Lemma II.2.

For the rest of the terms in  $(\sum_{i \in T_x} \langle a_i, x \rangle^4)^t$ , we use combinations of Lemma II.2 and  $\langle a_i, x \rangle^2 \leq \tau$  to bound, which ultimately proves Lemma II.1.

### D. Overview of the trace moment method

In this section, we make a detour and give an overview of the trace moment method. Readers familiar with the analysis of random matrix norm bounds by Tao [44] may skip directly to Section II-E. The starting point of the trace moment method is the inequality

$$\frac{1}{N} \cdot \text{tr}(M^\ell) \leq \|M\|_2^\ell \leq \text{tr}(M^\ell),$$

which holds for any  $N \times N$  symmetric matrix  $M$  and any even integer  $\ell$ . Expanding  $\text{tr}(M^\ell)$  and taking expectation (when  $M$  is a random matrix), we get

$$\mathbb{E}[\|M\|_2^\ell] \leq \mathbb{E}[\text{tr}(M^\ell)] = \mathbb{E} \sum_{i_1, i_2, \dots, i_\ell \in [N]} M_{i_1, i_2} M_{i_2, i_3} \cdots M_{i_\ell, i_1}.$$

Notice that the summation is over closed walks on  $[N]$ :  $i_1 \rightarrow i_2 \rightarrow \cdots \rightarrow i_\ell \rightarrow i_1$ . We can view these as closed walks on the complete graph with  $N$  vertices.

We now briefly explain Tao's analysis [44] on bounding  $\mathbb{E}[\text{tr}(M^\ell)]$  for the (symmetric) random sign matrix, where each entry is uniformly random  $\pm 1$ . The most important observation is that if a closed walk uses any edge (in the  $N$ -vertex complete graph) an *odd* number of times, then the expectation is zero. Thus, we only need to consider closed walks that use each edge an even number of times, each of which contributes 1 in the trace.

**Encoding closed walks.** The main idea is to *combinatorially* count all such "even" closed walks by providing an *encoding*. Following the terminology of [44], we refer to a step  $i_k \rightarrow i_{k+1}$  in the walk as a *leg*. We call a leg "fresh" if it traverses an edge that we haven't seen before, and "return" if it traverses a previously used edge. Since each edge must be traversed at least twice, we can have at most  $\ell/2$  fresh legs. We will view a fresh leg as creating an "active" edge, while a return leg closes an active edge; all edges must be closed in the end. Now, we can encode a closed walk as follows: (1) pick a starting vertex in  $[N]$ , (2) label each leg as "fresh" or "return" such that there are at most  $\ell/2$  fresh legs, (3) for a fresh leg we have  $N$  choices to choose the destination, and for now, assume that each return leg has a unique choice so that we don't need to specify a destination. Then, we upper bound  $\mathbb{E}[\text{tr}(M^\ell)]$  by upper bounding the number of encodings:

$$\mathbb{E}[\text{tr}(M^\ell)] \leq N \cdot 2^\ell \cdot N^{\ell/2} = N \cdot (2\sqrt{N})^\ell.$$

Then, setting  $\ell \gg \log N$ , by Markov's inequality, with probability  $1 - \frac{1}{\text{poly}(N)}$  we have

$$\|M\|_2 \leq \text{tr}(M^\ell)^{1/\ell} \leq N^{O(1/\ell)} \cdot 2\sqrt{N} \leq (2 + o(1))\sqrt{N}.$$

This is the desired upper bound with the correct constant factor 2.

**Unforced return legs.** There is an important assumption that we made in the above analysis: each return leg has a unique choice. This is obviously not true for all walks, but we will prove that this is true for *most* walks. In [44], a return leg with one unique choice is called *forced*, and ones with multiple choices are called *unforced*. We need to prove that the closed walks with any unforced return legs are negligible. The key observation is that if a return leg is unforced at vertex  $v$ , i.e., there are multiple active edges incident to  $v$ , it must be the case that there were previous fresh legs that went back to  $v$ . Such legs are called *non-innovative*, as they create new active edges but not new vertices, and as a result, each non-innovative leg requires only a factor  $\ell$  — the length of the walk — to specify its destination (as opposed to  $N$  for fresh legs).

The main idea is to *charge* the extra information required for unforced return legs to the non-innovative legs. We need to prove (1) an upper bound on the unforced return legs in terms of the number of non-innovative legs, and (2) the cost of a non-innovative

leg (with extra information) is still  $\ll N$ , negligible compared to a fresh leg. This is a key technical challenge in most prior works on (sharp) norm bounds for more complicated random matrices [28], [33], [44].

#### E. Upper bound on $P_t(x)$

We now give an overview of the proof of Lemma II.2, which is the most technical part.

We start by writing  $\langle a_i, x \rangle^4 = \langle a_i^{\otimes 4}, x^{\otimes 4} \rangle = \langle \text{Shift}(a_i^{\otimes 4}), x^{\otimes 4} \rangle$ , where  $\text{Shift}$  is formally defined in Definition III.7 and is also done in [3] to remove the large rank-1 component in the matrix (see Remark III.13 for a lower bound without this). Then,

$$P_t(x) = \left\langle t! \sum_{S \subseteq [n]: |S|=t} \bigotimes_{i \in S} \text{Shift}(a_i^{\otimes 4}), x^{\otimes 4t} \right\rangle \\ = (x^{\otimes 2t})^\top M (x^{\otimes 2t}),$$

where we view  $x^{\otimes 2t}$  as a vector of dimension  $d^{2t}$ , and

$$M := t! \sum_{S \subseteq [n]: |S|=t} \bigotimes_{i \in S} \text{Shift}(a_i^{\otimes 4}) \in \mathbb{R}^{d^{2t} \times d^{2t}}.$$

Here, we view  $a_i^{\otimes 4}$  as a  $d^2 \times d^2$  matrix.

The natural approach to bound  $P_t(x)$  is to bound the spectral norm of  $M$ , since  $P_t(x) \leq \|M\|_2 \cdot \|x^{\otimes 2t}\|_2^2 = \|M\|_2$  for any unit vector  $x$ . Unfortunately, it can be shown that  $\|M\|_2 \geq \Omega(d)^{2t} \gg n^t$  (see Remark III.12). In fact, using  $\|M\|_2$  to bound  $P_t(x)$  does not improve on [3] at all.

**Symmetrization lowers spectral norm.** To remedy this, observe that the flattened vector  $x^{\otimes 2t}$  in the quadratic form is highly symmetric. Specifically, let  $\Pi \in \mathbb{R}^{d^{2t} \times d^{2t}}$  be *any* permutation matrix that maps an index  $J = (j_1, j_2, \dots, j_{2t}) \in [d]^{2t}$  to  $(j_{\pi(1)}, j_{\pi(2)}, \dots, j_{\pi(2t)})$  for some permutation  $\pi$  over  $2t$  elements. Then, we have that

$$\Pi x^{\otimes 2t} = x^{\otimes 2t}.$$

We will use  $\hat{\mathcal{S}}_{2t}$  to denote the collection of all such matrices  $\Pi$ . It follows that for any  $\Pi, \Pi' \in \hat{\mathcal{S}}_{2t}$ ,

$$(x^{\otimes 2t})^\top M (x^{\otimes 2t}) = (x^{\otimes 2t})^\top (\Pi M \Pi') (x^{\otimes 2t}).$$

In particular,

$$(x^{\otimes 2t})^\top M (x^{\otimes 2t}) = (x^{\otimes 2t})^\top \mathbb{E}_{\Pi, \Pi' \sim \hat{\mathcal{S}}_{2t}} [\Pi M \Pi'] (x^{\otimes 2t}) \leq \|\tilde{M}\|_2,$$

where  $\tilde{M} = \mathbb{E}_{\Pi, \Pi' \sim \hat{\mathcal{S}}_{2t}} [\Pi M \Pi']$ .

As  $\tilde{M}$  is the symmetrized version of  $M$ , one may expect that the spectral norm decreases. Indeed, this is also the main idea behind the results of [7], [8], [42]. Our key technical result is that the symmetrization lowers the spectral norm by a factor of roughly  $t^t$ , and we prove it using the trace moment method.

**Lemma II.3** (Informal Lemma III.9). *Let  $\ell = \log n$ . Then,*

$$\mathbb{E} \text{tr}(\tilde{M}^\ell) \leq d^{2t} \cdot O\left(n + \frac{d^2}{t} \log^2 n\right)^{t^\ell}.$$

Here, we need  $\ell = \log n$  (as opposed to  $\log(\dim(\tilde{M})) = 2t \log d$ ) since we only need  $d^{2t/\ell} \leq O(1)^t$ , and saving this  $t$  factor turns out to be important in our analysis. From the discussion above, Lemma II.2 follows immediately since  $P_t(x) = (x^{\otimes 2t})^\top \tilde{M}(x^{\otimes 2t}) \leq \|\tilde{M}\|_2 \leq \text{tr}(\tilde{M}^\ell)^{1/\ell}$ . We also remark that Lemma II.3 is tight; see Remark III.12.

**Trace method for  $\tilde{M}$ .** First, note that  $\mathbb{E} \text{tr}(\tilde{M}^\ell) = \mathbb{E} \text{tr} \mathbb{E}_{\Pi_1, \Pi'_1, \dots, \Pi_\ell, \Pi'_\ell} [\Pi_1 M \Pi'_1 \cdots \Pi_\ell M \Pi'_\ell]$ . The trace is the sum of length- $\ell$  closed walks  $I_1 \rightarrow I_2 \rightarrow \cdots \rightarrow I_\ell \rightarrow I_1$  on the indices (in  $[d]^{2t}$ ). For each sequence  $\pi = (\Pi_1, \Pi'_1, \dots, \Pi_\ell, \Pi'_\ell)$ , this corresponds to a *labeling* of a specific *structure* defined by  $\pi$ , viewed as a graph containing circle and square vertices (see Definitions III.10 and III.11, and Figure 1 for examples). The circle and square vertices receive labels in  $[d]$  and  $[n]$  respectively. We require that each labeled edge (an element in  $[n] \times [d]$ ) appears an even number of times, otherwise the expectation is zero.

We next define an encoding of the labeling such that (1) every valid labeling for a given structure can be encoded, and (2) a labeling is uniquely determined given an encoding and a structure (if decoding succeeds). Similar to the analysis in Section II-D, we mark each leg as “Fresh”, “Return”, “Non-innovative” or “High-multiplicity”. We also introduce an extra “Paired” type since in our case a pair of two circle vertices lead to the same square vertex. In addition, we need to handle potential unforced return legs by charging extra information to the non-innovative and high-multiplicity legs. These are all carefully done in Section III-C.

**Decoding success probability.** We now formalize the idea that symmetrization reduces the spectral norm as follows,

$$\begin{aligned} \mathbb{E} \text{tr}(\tilde{M}^\ell) &\leq \mathbb{E}_\pi \sum_{\sigma: \text{labelings}} \mathbf{1}(\sigma \text{ valid for } \pi) \\ &\leq \sum_{\zeta: \text{encoding}} \Pr[\text{Decode}(\zeta, \pi) \text{ succeeds}]. \end{aligned}$$

We upper bound the probability of the decoder succeeding (for any encoding  $\zeta$ ) in Lemma III.20. At a high level, for any structure (Figure 1), the circle vertices are paired and connected to a square vertex. During decoding, each square vertex receives two labels from the two legs (coming from previous circles), and the decoding fails if there is a conflict. Now, when we randomize the structure  $\pi$ , which we can view as rewiring the edges in the structure and pairing up circle vertices within each block, the decoding succeeds only if all circle vertices are paired up in the correct way. For each block, there are  $2t$  circle vertices, and there are  $(2t-1)!! \geq \Omega(t)^t$  number of ways to group them into pairs. Thus, for all  $\ell$  blocks, we get a factor  $t^{\ell}$  improvement. We defer the details to Section III-D, and the proof of Lemma II.3 is completed in Section III-E.

#### F. Finding planted sparse vector

It turns out that our upper bound on  $P_t(x)$  can be used for the related planted problem. More specifically, the proof of Lemma II.2 shows a stronger statement: the upper bound on  $P_t(x)$  exhibits a degree- $4t$  Sum-of-Squares (SoS) proof, meaning that any pseudo-distribution  $\mu$  satisfies that  $\tilde{\mathbb{E}}_\mu[P_t(x)]$  is small (see Section IV-B for background on SoS proofs and pseudo-distributions).

On the other hand, suppose  $\tilde{A} = AR$  is drawn from Model 1 with a planted sparse vector  $v$  such that  $Q_t(v)$  is large, and let  $\tilde{P}_t(x) := Q_t(\tilde{A}x)$ . Then, we must have  $\max_\mu \tilde{\mathbb{E}}_\mu[\tilde{P}_t(x)] \geq Q_t(v)$ , where the maximum is over all pseudo-distributions satisfying the unit sphere constraint. This is because taking  $x = r_1 := R^\top e_1$  (the first row of  $R$ ) gives  $\tilde{P}_t(r_1) = Q_t(v)$ . Moreover, the pseudo-distribution that maximizes  $\tilde{P}_t(x)$  can be computed in  $n^{O(t)}$  time using the SoS algorithm.

Thus, the main intuition is that this pseudo-distribution  $\mu$  must have significant support on vectors close to  $r_1$ , otherwise  $\tilde{\mathbb{E}}_\mu[\tilde{P}_t(x)]$  cannot be large. We formalize this in Lemma IV.12 and prove that  $\tilde{\mathbb{E}}_\mu[\langle r_1, x \rangle^{2t}] \geq e^{-o(t)} = (1 - o(1))^t$ .

Now, we use a rounding algorithm of [5] (Lemma IV.10) to obtain a list of  $2^{O(t)}$  unit vectors with the guarantee that one of them, say  $\hat{x}$ , is close to  $r_1$ , which means that  $\tilde{v} = \tilde{A}\hat{x}$  is close to  $v$ . The next challenge is to identify a “good”  $\hat{x}$  among this list. The natural idea is to take the vector which is the “most compressed”, i.e., the one whose top  $\rho n$  entries have the largest norm. Lemma IV.6 proves that this indeed works, which completes the proof.

### III. CERTIFYING SPREAD OF A RANDOM SUBSPACE

Recall from Eq. (1) that the distortion of a subspace  $X$  is defined as  $\Delta(X) = \sup_{x \in X: x \neq 0} \frac{\sqrt{n}\|x\|_2}{\|x\|_1}$ . In the following, we state an equivalent notion of spreadness.

**Definition III.1** (Spreadness property of a subspace). A subspace  $X \subseteq \mathbb{R}^n$  is  $(t, \epsilon)$ -spread if for every  $x \in X$  and every  $S \subseteq [n]$  with  $|S| \leq t$ , we have

$$\|x_S\|_2 \geq \epsilon \cdot \|x\|_2.$$

In other words,  $X$  is  $(t, \epsilon)$ -spread if any vector  $x \in X$  still has large ( $\geq \epsilon$  fraction) norm after removing any  $t$  coordinates. The relationship between the distortion and spread of a subspace was proved in [22]:

**Lemma III.2** (Lemma 2.11 of [22]). *Let  $X \subseteq \mathbb{R}^n$  be a subspace.*

- (1) *If  $X$  is  $(t, \epsilon)$ -spread, then  $\Delta(X) \leq \sqrt{\frac{n}{t}} \cdot \epsilon^{-2}$ .*
- (2) *Conversely,  $X$  is  $\left(\frac{n}{2\Delta(X)^2}, \frac{1}{4\Delta(X)}\right)$ -spread.*

In light of Lemma III.2, to prove that  $\Delta(X) \leq O(1)$ , it suffices to prove that  $X$  is  $(\Omega(n), \Omega(1))$ -spread.

We now state our main result for certification.

**Theorem III.3** (Formal version of Theorem 1). Let  $d, n \in \mathbb{N}$  and  $\varepsilon \in (0, 1)$  such that  $d = n^{\frac{1+\varepsilon}{2}} / \log n$  and  $n^\varepsilon \log^6 n \leq n$ . Let  $A \sim \mathcal{N}(0, 1)^{n \times d}$ . Then, there is a certification algorithm that runs in  $n^{O(n^\varepsilon)}$  time and, with probability  $1 - \frac{1}{\text{poly}(n)}$  over  $A$ , certifies that  $\text{colspan}(A)$  is  $(\alpha n, 1/2)$ -spread, where  $\alpha$  is a universal constant.

We remark that our algorithm works not just for Gaussian matrices but also matrices with more general random variables; see Lemma III.9 for the requirements.

#### A. Removing the top entries of $Ax$

As discussed in Section II-B, the barrier of the certification algorithm of [3] at  $d \gg \sqrt{n}$  is that if  $x$  is highly correlated with some row  $a_i$  in  $A$  (i.e.,  $|\langle a_i, x \rangle|$  is large), then  $\|Ax\|_4^4$  can be much larger than  $n$ . Our intuition is that the number of such large entries in  $Ax$  must be very small. Thus, if we remove the top few entries of  $Ax$ , then the 4-norm is at most  $O(n)$ , the desired bound.

The next lemma states that this is indeed the case and that we can certify this.

**Lemma III.4.** Let  $d, n \in \mathbb{N}$  and  $\varepsilon \in (0, 1)$  such that  $d = n^{\frac{1+\varepsilon}{2}} / \log n$  and  $n^\varepsilon \log^6 n \leq n$ . Let  $a_1, \dots, a_n \in \mathbb{R}^d$  be random vectors with i.i.d.  $\mathcal{N}(0, 1)$  entries. Then, there is a certification algorithm that runs in  $n^{O(n^\varepsilon)}$  time and, with probability  $1 - \frac{1}{\text{poly}(n)}$  over  $a_1, \dots, a_n$ , certifies the following:

- (1) For any  $S \subseteq [n]$  of size  $|S| = n^\varepsilon$ ,  $\sum_{i \in S} \langle a_i, x \rangle^2 \leq (1 + o_n(1)) \cdot d$  for all unit vectors  $x \in \mathbb{R}^d$ .
- (2) For any unit vector  $x \in \mathbb{R}^d$ ,

$$\sum_{i \in T_x} \langle a_i, x \rangle^4 \leq O(n).$$

where  $T_x \subseteq [n]$ ,  $|T_x| = n - n^\varepsilon$  is the set of indices excluding the top  $n^\varepsilon$  with the largest  $\langle a_i, x \rangle^2$ .

Proving (1) is straightforward by applying the following standard matrix concentration result and a union bound.

**Fact III.5** (See e.g. [45]). Let  $m \leq n$ , and let  $A \in \mathbb{R}^{n \times m}$  be a matrix with i.i.d.  $\mathcal{N}(0, 1)$  entries. Then, for every  $t \geq 0$ , with probability at least  $1 - 2e^{-t^2/2}$ , we have  $\sqrt{n} - \sqrt{m} - t \leq \sigma_{\min}(A) \leq \sigma_{\max}(A) \leq \sqrt{n} + \sqrt{m} + t$ .

For (2), first note that (1) implies that  $\langle a_i, x \rangle^2 \leq \frac{d}{n^\varepsilon}(1 + o_n(1))$  for all  $i \in T_x$ . Then, we consider  $(\sum_{i \in T_x} \langle a_i, x \rangle^4)^t$ . For starters, let's focus on the term  $t! \sum_{S \subseteq T_x: |S|=t} \prod_{i \in S} \langle a_i, x \rangle^4$ , i.e., the terms where the indices are distinct. The main observation is that we can upper bound this by  $P_t(x) := t! \sum_{S \subseteq [n]: |S|=t} \prod_{i \in S} \langle a_i, x \rangle^4$ , where we replace  $T_x$  with  $[n]$ . Note that this quantity crucially does not depend on  $T_x$ .

The bulk of our proof is then to prove the following lemma:

**Lemma III.6.** Let  $t \leq d \leq n$  be integers such that  $t \log^6 n \leq n + \frac{d^2}{t} \log^2 n$ . Let  $a_1, \dots, a_n \in \mathbb{R}^d$  be random vectors with

i.i.d.  $\mathcal{N}(0, 1)$  entries. Then, there is a certification algorithm that runs in  $n^{O(t)}$  time and, with probability  $1 - \frac{1}{\text{poly}(n)}$  over  $a_1, \dots, a_n$ , certifies that for all unit vectors  $x \in \mathbb{R}^d$ ,

$$P_t(x) := t! \sum_{S \subseteq [n]: |S|=t} \prod_{i \in S} \langle a_i, x \rangle^4 \leq O\left(n + \frac{d^2}{t} \log^2 n\right)^t.$$

We will prove Lemma III.6 at the end of Section III-B. We first use Lemma III.6 to prove Lemma III.4.

*Proof of Lemma III.4 from Lemma III.6.* Let  $t := n^\varepsilon$ , and let  $A$  be the  $n \times d$  matrix with  $a_1, a_2, \dots, a_n$  as columns. The certification algorithm is as follows,

- (i) For each  $S \subseteq [n]$  with  $|S| = t$ , verify that  $\sigma_{\max}(A_S) \leq (1 + o_n(1))\sqrt{d}$ , where  $A_S \in \mathbb{R}^{t \times d}$  is the submatrix of  $A$  obtained by choosing rows according to  $S$ .
- (ii) For  $s = 1, 2, \dots, t$ , use the algorithm in Lemma III.6 to certify that

$$P_s(x) := s! \sum_{S \subseteq [n]: |S|=s} \prod_{i \in S} \langle a_i, x \rangle^4 \leq O\left(\frac{d^2}{s} \log^2 n\right)^s \quad \text{for all unit vectors } x \in \mathbb{R}^d. \quad (2)$$

First, since  $t \leq d / \log^2 n$ , by Fact III.5, with probability  $1 - 2^{-2t \log n}$  we have  $\sigma_{\max}(A_S) \leq \sqrt{d} + O(\sqrt{t \log n}) \leq (1 + o_n(1))\sqrt{d}$ . Then, a union bound over all  $\binom{n}{t} \leq 2^{t \log n}$  choices of  $S$  shows that this holds for all  $S$ . This certifies (1) of Lemma III.4.

To certify (2), first note that  $\bar{T}_x = [n] \setminus T_x$  is the set of indices with the top  $t$  largest values of  $\langle a_i, x \rangle^2$ , and by (1) we have  $\sum_{i \in \bar{T}_x} \langle a_i, x \rangle^2 = \|A_{\bar{T}_x} x\|_2^2 \leq d(1 + o_n(1))$ . Thus, it follows that

$$\langle a_i, x \rangle^2 \leq \tau := \frac{d}{t}(1 + o_n(1)) \quad \text{for all } i \in T_x. \quad (3)$$

The parameters of  $d, n$  and  $t$  satisfy  $n = \frac{d^2}{t} \log^2 n$  and  $t \log^6 n \leq n + \frac{d^2}{t} \log^2 n$ , the requirements for Lemma III.6. Thus, we can certify Eq. (2) in  $n^{O(t)}$  time for all  $s \leq t$ .

We now proceed to bound  $\sum_{i \in T_x} \langle a_i, x \rangle^4$ . We will use Eq. (2) and (3) to bound the  $t$ -th power:

$$\begin{aligned} \left( \sum_{i \in T_x} \langle a_i, x \rangle^4 \right)^t &= \sum_{i_1, \dots, i_t \in T_x} \prod_{k \in [t]} \langle a_{i_k}, x \rangle^4 \\ &= \sum_{S \subseteq T_x} \sum_{i_1, \dots, i_t \in S: \text{supp}(i_1, \dots, i_t) = S} \prod_{k \in [t]} \langle a_{i_k}, x \rangle^4, \end{aligned}$$

where we group the terms according to the support of the indices. For any  $i_1, \dots, i_t \in T_x$  with  $\text{supp}(i_1, \dots, i_t) = S$ , since  $\langle a_i, x \rangle^4 \leq \tau^2$  for  $i \in T_x$  (Eq. (3)), we have

$$\prod_{k \in [t]} \langle a_{i_k}, x \rangle^4 \leq \tau^{2(t-|S|)} \prod_{j \in S} \langle a_j, x \rangle^4.$$

Moreover, for any  $S \subseteq T_x$  of size  $s \leq t$ , we claim that the number of ordered indices  $(i_1, \dots, i_t)$  with support  $S$  is at most  $t! \cdot \binom{t-1}{t-s}$ . To see this, we can construct  $(i_1, \dots, i_t)$  by

first choosing  $t-s$  elements from  $S$  with replacement, for which there are  $\binom{s+(t-s)-1}{t-s} = \binom{t-1}{t-s}$  choices (alternatively, this is the number of ways of throwing  $t$  balls into  $s$  bins so that no bin is non-empty), and then there are  $t!$  ways to permute the indices. Thus,

$$\begin{aligned} \left( \sum_{i \in T_x} \langle a_i, x \rangle^4 \right)^t &\leq \sum_{s=1}^t \sum_{S \subseteq T_x: |S|=s} t! \binom{t-1}{t-s} \tau^{2(t-s)} \prod_{i \in S} \langle a_i, x \rangle^4 \\ &\leq \sum_{s=1}^t \frac{t!}{s!} \binom{t-1}{t-s} \tau^{2(t-s)} \\ &\quad \cdot s! \sum_{S \subseteq [n]: |S|=s} \prod_{i \in S} \langle a_i, x \rangle^4. \end{aligned}$$

Note that we changed the summation of  $S \subseteq T_x$  to  $S \subseteq [n]$  (since  $T_x \subseteq [n]$  and all terms are non-negative). Crucially, we get an upper bound that does not depend on  $T_x$ , thus allowing us to use Eq. (2). Since  $\frac{d^2}{s} \log^2 n \geq \frac{d^2}{t} \log^2 n = n$  and  $\tau = O(\frac{d}{t})$ , the above is bounded by

$$\begin{aligned} &\sum_{s=1}^t \frac{t!}{s!} \binom{t-1}{t-s} \cdot O\left(\frac{d}{t}\right)^{2(t-s)} O\left(\frac{d^2}{s} \log^2 n\right)^s \\ &\leq O\left(\frac{d^2}{t} \log^2 n\right)^t \sum_{s=1}^t \binom{t-1}{t-s} \left(\frac{t}{s}\right)^s \\ &\leq O(n)^t \sum_{s=1}^t \binom{t-1}{t-s} \left(\frac{t}{s}\right)^s \\ &\leq O(n)^t \cdot 2^{2t}. \end{aligned}$$

Thus, we have certified that  $(\sum_{i \in T_x} \langle a_i, x \rangle^4)^t \leq O(n)^t$ , completing the proof.  $\square$

With Lemma III.4, the proof of Theorem III.3 is straightforward.

*Proof of Theorem III.3 from Lemma III.4.* Let  $x \in \mathbb{R}^d$  be any unit vector, let  $y = Ax$ , and let  $S \subseteq [n]$  be any subset of size  $\leq \alpha n$ . We would like to certify that  $\|y_{\bar{S}}\|_2 \geq \frac{1}{2} \|y\|_2$ . Let  $T_x \subseteq [n]$  be as defined in Lemma III.4, i.e., the set of indices excluding the top  $n^\epsilon$  largest  $\langle a_i, x \rangle^2$ . By (1) of Lemma III.4, we can certify that  $\|y_{T_x}\|_2^2 = \|A_{T_x} x\|_2^2 \leq d(1 + o_n(1))$  since  $|T_x| = n^\epsilon$  by definition. Moreover, by Fact III.5 we know that  $\sigma_{\min}(A) \geq \sqrt{n} - \sqrt{d} - o_n(1)$  with high probability, and since  $d = o(n)$ , we have  $\|y\|_2^2 \geq n(1 - o_n(1))$ . This means that  $\|y_{T_x}\|_2^2 \leq o_n(1) \|y\|_2^2$ .

Let  $S' := S \cap T_x = S \setminus \bar{T}_x$ . Then, we have

$$\|y_{S'}\|_2^2 \geq \|y\|_2^2 - \|y_{\bar{S}}\|_2^2 - \|y_{\bar{T}_x}\|_2^2 \geq (1 - o_n(1)) \|y\|_2^2 - \|y_{\bar{S}}\|_2^2$$

Next, by (2) of Lemma III.4,  $\sum_{i \in S'} y_i^4 \leq \sum_{i \in T_x} y_i^4 \leq Bn$  for some constant  $B$ . Thus, by Cauchy-Schwarz,

$$\begin{aligned} \|y_{S'}\|_2^2 &= \sum_{i \in S'} y_i^2 \leq \sqrt{|S'|} \sum_{i \in S'} y_i^4 \leq \sqrt{\alpha n} \cdot \sqrt{Bn} \\ &\leq \sqrt{\alpha B} \cdot \|y\|_2^2 \cdot (1 + o_n(1)). \end{aligned}$$

Thus, we can set the constant  $\alpha$  such that  $\|y_{\bar{S}}\|_2^2 \geq (1 - \sqrt{\alpha B} - o_n(1)) \|y\|_2^2 \geq \frac{1}{4} \|y\|_2^2$ .  $\square$

### B. Symmetrization lowers spectral norm

Note that  $\langle a_i, x \rangle^4 = \langle a_i^{\otimes 4}, x^{\otimes 4} \rangle$ , where we may view  $a_i^{\otimes 4}$  as a 4-th order tensor or a  $d^2 \times d^2$  matrix. We start by redistributing the entries of  $a_i^{\otimes 4}$ .

**Definition III.7.** Given a  $d^2 \times d^2$  matrix  $A$  indexed by tuples  $(j_1, j_2), (j_3, j_4) \in [d]$ , we define  $\text{Shift}(A)$  to be the  $d^2 \times d^2$  matrix such that  $\text{Shift}(A)_{(j_1, j_2), (j_3, j_4)}$  equals

$$\begin{cases} 0 & \text{if } j_1 = j_2 \notin \{j_3, j_4\} \text{ or } j_3 = j_4 \notin \{j_1, j_2\} \\ \frac{3}{2} A_{(j_1, j_2), (j_3, j_4)} & \text{if } j_1 \neq j_2, j_3 \neq j_4, \text{ and } |\{j_1, j_2, j_3, j_4\}| \leq 3 \\ A_{j_1, j_2, j_3, j_4} & \text{otherwise.} \end{cases}$$

For example  $\text{Shift}(A)_{(1,1), (2,2)} = 0$  and  $\text{Shift}(A)_{(1,2), (1,2)} = \frac{3}{2} A_{(1,2), (1,2)}$ . This is also done in [3] to remove a large rank-1 component in  $\Sigma = \mathbb{E}[a_i^{\otimes 4}]$  (recall Section II-A) so that  $\text{Shift}(\Sigma)$  has norm  $O(1)$  even though  $\|\Sigma\|_2 \geq n$ . In Remark III.13, we will see that without this, the spectral norm bound is false.

**Proposition III.8.** For any  $a, x \in \mathbb{R}^d$ ,  $\langle a, x \rangle^4 = \langle \text{Shift}(a^{\otimes 4}), x^{\otimes 4} \rangle$ .

*Proof.* Expanding  $\langle a, x \rangle^4$ :  $\sum_{j_1, j_2, j_3, j_4} a_{j_1} a_{j_2} a_{j_3} a_{j_4} x_{j_1} x_{j_2} x_{j_3} x_{j_4}$ , where we can group the terms according to the multiset  $\{j_1, j_2, j_3, j_4\}$ . The only entries that differ between  $a^{\otimes 4}$  and  $\text{Shift}(a^{\otimes 4})$  are the ones where the multiset  $\{j_1, j_2, j_3, j_4\}$  is of the following form:

- $j_1 = j_2 \neq j_3 = j_4$ : there are 6 such terms in  $a^{\otimes 4}$ , and there are 4 in  $\text{Shift}(a^{\otimes 4})$ , each scaled by  $\frac{3}{2}$ .
- $j_1 = j_2 \neq j_3 \neq j_4$ : there are 12 such terms in  $a^{\otimes 4}$ , and there are 8 in  $\text{Shift}(a^{\otimes 4})$ , each scaled by  $\frac{3}{2}$ .

This shows that  $\langle a, x \rangle^4 = \langle \text{Shift}(a^{\otimes 4}), x^{\otimes 4} \rangle$ .  $\square$

To prove Lemma III.6, we start by writing  $t! \sum_{|S|=t} \prod_{i \in S} \langle a_i, x \rangle^4 = t! \sum_{|S|=t} \prod_{i \in S} \langle \text{Shift}(a_i^{\otimes 4}), x^{\otimes 4} \rangle$  as a quadratic form:

$$\begin{aligned} &t! \sum_{S \subseteq [n]: |S|=t} \prod_{i \in S} \langle \text{Shift}(a_i^{\otimes 4}), x^{\otimes 4} \rangle \\ &= \left\langle t! \sum_{S \subseteq [n]: |S|=t} \bigotimes_{i \in S} \text{Shift}(a_i^{\otimes 4}), x^{\otimes 4t} \right\rangle = (x^{\otimes 2t})^\top M (x^{\otimes 2t}), \end{aligned} \quad (4)$$

where we view  $x^{\otimes 2t}$  as a vector of dimension  $d^{2t}$ , and

$$M := t! \sum_{S \subseteq [n]: |S|=t} \bigotimes_{i \in S} \text{Shift}(a_i^{\otimes 4}) \in \mathbb{R}^{d^{2t} \times d^{2t}}. \quad (5)$$

Here, we view  $\text{Shift}(a_i^{\otimes 4})$  as a  $d^2 \times d^2$  matrix.

The natural approach to bound Eq. (4) is to bound the spectral norm of  $M$ . Unfortunately, it can be shown that  $\|M\|_2 \geq \Omega(d)^{2t}$ . As explained in Section II-E, we resolve this by exploiting the symmetry of  $x^{\otimes 2t}$ . Let  $\Pi \in$



$\mathbb{R}^{d^{2t} \times d^{2t}}$  be any permutation matrix that maps an index  $J = (j_1, j_2, \dots, j_{2t}) \in [d]^{2t}$  to  $(j_{\pi(1)}, j_{\pi(2)}, \dots, j_{\pi(2t)})$  for some permutation  $\pi$  over  $2t$  elements. Then, we have that  $\Pi x^{\otimes 2t} = x^{\otimes 2t}$ .

We will use  $\hat{\mathcal{S}}_{2t}$  to denote the collection of all such matrices  $\Pi$ . It follows that

$$(x^{\otimes 2t})^\top M(x^{\otimes 2t}) = (x^{\otimes 2t})^\top \tilde{M}(x^{\otimes 2t}) \leq \|\tilde{M}\|_2, \quad (6)$$

where  $\tilde{M} = \mathbb{E}_{\Pi, \Pi' \sim \hat{\mathcal{S}}_{2t}}[\Pi M \Pi']$ .

Our key technical result is that the symmetrization lowers the spectral norm by a factor of roughly  $t^t$ , and we prove it using the trace moment method, a standard technique for upper bounding spectral norm of matrices.

**Lemma III.9.** *Let  $a_1, \dots, a_n \in \mathbb{R}^d$  be random vectors with independent entries such that  $\mathbb{E}[a_{ij}^k] = 0$  for all odd  $k$ ,  $\mathbb{E}[a_{ij}^2] \leq 1$ ,  $\mathbb{E}[a_{ij}^4] \leq \mu_4$ , and  $|a_{ij}| \leq \sqrt{C \log n}$  almost surely for  $\mu_4 \geq 1$  and constant  $C > 0$ . Let  $t \leq d \leq n$  be integers such that  $t \log^6 n \leq \mu_4 n + \frac{d^2}{t} \log^2 n$ . Let  $\tilde{M}$  be the  $d^{2t} \times d^{2t}$  matrix defined in Eq. (6). Let  $\ell \leq \log n$ . Then,*

$$\mathbb{E}_A[\text{tr}(\tilde{M}^\ell)] \leq d^{2t} \cdot O\left(\mu_4 n + \frac{d^2}{t} \log^2 n\right)^{t\ell}.$$

In Remark III.12, we will show that the upper bound in Lemma III.9 is tight up to log factors.

We can immediately complete the proof of Lemma III.6 using Lemma III.9.

*Proof of Lemma III.6 from Lemma III.9.* Since  $a_1, \dots, a_n \in \mathbb{R}^d$  are vectors with  $\mathcal{N}(0, 1)$  entries, we have that  $\mathbb{E}[a_{ij}^4] = 3$ , and moreover, with probability  $1 - \frac{1}{\text{poly}(n)}$  we have that  $|a_{ij}| \leq \sqrt{C \log n}$  for some constant  $C > 0$  for all  $i \in [n]$ ,  $j \in [d]$ . Thus, we can now condition on this event; the conditioned variables still satisfy  $\mathbb{E}[a_{ij}^k] = 0$  for odd  $k$ ,  $\mathbb{E}[a_{ij}^2] \leq 1$  and  $\mathbb{E}[a_{ij}^4] \leq 3$ , i.e., the conditions in Lemma III.9.

Set  $\ell = \log n$ . By Lemma III.9 and Markov's inequality, with probability  $1 - \exp(-t\ell) \geq 1 - n^{-t}$ , we have that  $\text{tr}(\tilde{M}^\ell) \leq d^{2t} \cdot O(n + \frac{d^2}{t} \log^2 n)^{t\ell}$ . Then,  $\|\tilde{M}\|_2 \leq \text{tr}(\tilde{M}^\ell)^{1/\ell} \leq O(n + \frac{d^2}{t} \log^2 n)^t$ , using the fact that  $d^{2t/\ell} \leq 2^{2t}$ .

We can construct the matrix  $\tilde{M} \in \mathbb{R}^{d^{2t} \times d^{2t}}$  and calculate its spectral norm in  $n^{O(t)}$  time. Since  $(x^{\otimes 2t})^\top \tilde{M}(x^{\otimes 2t}) \leq \|\tilde{M}\|_2 \cdot \|x^{\otimes 2t}\|_2^2$  and  $\|x^{\otimes 2t}\|_2 = 1$  for all unit vectors  $x$ , this completes the proof.  $\square$

### C. Encoding of walks in the trace

In this section, we prove Lemma III.9. Expanding  $\mathbb{E}_A \text{tr}(\tilde{M}^\ell)$  gives

$$\mathbb{E}_A \text{tr} \mathbb{E}_{\Pi_1, \Pi'_1, \Pi_2, \Pi'_2, \dots, \Pi_\ell} [\Pi_1 M \Pi'_1 \Pi_2 M \Pi'_2 \cdots \Pi_\ell M \Pi'_\ell].$$

We will use  $\pi$  to denote the sequence of permutation matrices  $(\Pi_1, \Pi'_1, \Pi_2, \Pi'_2, \dots, \Pi_\ell) \in (\hat{\mathcal{S}}_{2t})^{2\ell}$ .

The trace power of a matrix is often viewed as a sum of closed walks. Indeed,  $\text{tr}(M^\ell) =$

$\sum_{I_1, I_2, \dots, I_\ell \in [d]^{2t}} M_{I_1, I_2} M_{I_2, I_3} \cdots M_{I_\ell, I_1}$ , which is a weighted sum of length- $\ell$  closed walks  $I_1 \rightarrow I_2 \rightarrow \cdots \rightarrow I_\ell \rightarrow I_1$  on the indices. With a specific permutation  $\pi$ , it is then a sum of closed walks where the indices are permuted accordingly at each step. See Figure 1 for an example.

**Definition III.10** (Structure). We will use a graph with two types of vertices (circle and square) to represent a walk of length  $\ell$  in the trace. The graph is divided into  $\ell$  blocks, each with  $t$  square vertices in the middle and  $4t$  edges between square and circle vertices. We use  $\pi = (\Pi_1, \Pi'_1, \dots, \Pi_\ell, \Pi'_\ell) \in (\hat{\mathcal{S}}_{2t})^{2\ell}$  to denote the *structure* of the graph, where the edges in block  $k$  are connected according to  $\Pi_k, \Pi'_k$ .

For example, Figure 1 shows two structures with  $t = 3$ . In Figure 1a, all permutations are identity. In Figure 1b, we have permutations  $\pi_1, \pi'_1, \dots, \pi_\ell, \pi'_\ell$ ; for example, on the left side of the first block,  $\pi_1(1) = 3, \pi_1(2) = 2, \pi_1(3) = 5$  and so on, i.e., circle  $i$  is connected to square  $\lceil \pi_1(i)/2 \rceil$ .

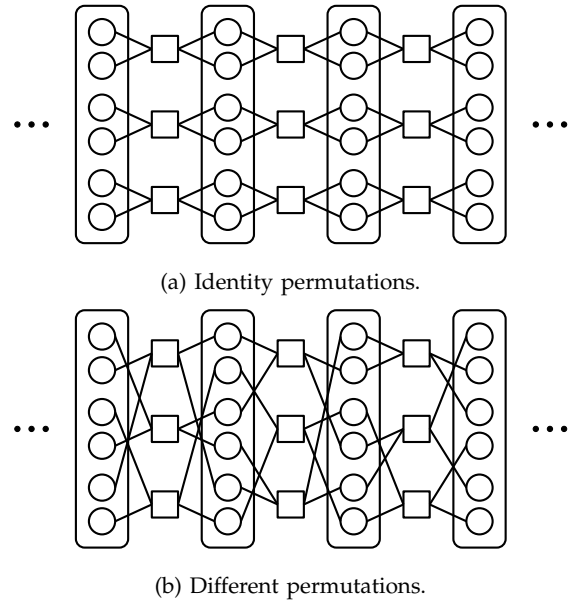


Fig. 1: Examples of different structures of walks in the trace.

**Definition III.11** (Valid labeling of a structure). A *labeling*  $\sigma$  is a map that maps circle vertices to  $[d]$  and square vertices to  $[n]$ . We say that a labeling  $\sigma$  is *valid* for a structure  $\pi$  if

- (1) All labeled edges (i.e., elements in  $[n] \times [d]$ ) appear even number of times.
- (2) The  $t$  square vertices in each block receive distinct labels.
- (3) For each square vertex, if the 2 incident circle vertices on the left (resp. right) have the same labels

$j \in [d]$ , then there must be at least one circle vertex on the right (resp. left) that is labeled  $j$ .

Moreover, we define  $\text{edge-factor}(\sigma, \pi)$  to be the product of factors from the labeled edges, where a labeled edge appearing  $k$  times gets a factor of  $\mathbb{E}[a^k]$ .

Requirement (1) is because  $\mathbb{E}[a_{ij}^k] = 0$  for all odd  $k$ , and because of this, labelings that violate requirement (1) automatically have  $\text{edge-factor}(\sigma, \pi) = 0$ . Requirement (2) is by definition of the  $M$  matrix (Eq. (5)), which is a sum over  $S \subseteq [n]$ . Finally, we can impose requirement (3) on the labelings because  $\text{Shift}(a_i^{\otimes 4})_{(j_1, j_1), (j_2, j_3)} = 0$  for  $j_1 \notin \{j_2, j_3\}$ . Indeed, recall from Definition III.7 that  $\text{Shift}(a_i^{\otimes 4})_{(j_1, j_1), (j_2, j_3)}$  is nonzero only if one or both of  $j_2, j_3$  equal  $j_1$ .

**Remark III.12** (Lower bound on  $\mathbb{E} \text{tr}(\tilde{M}^\ell)$ ). We claim that  $\mathbb{E} \text{tr}(\tilde{M}^\ell) \geq \Omega(\mu_4 n + \frac{d^2}{t})^{t\ell}$ , thus the upper bound in Lemma III.9 is tight up to log factors.

Fix a subset  $S \subseteq [n]$  and consider the labelings where the circle vertices are all distinct (i.e.,  $I_1, I_2, \dots, I_\ell \in [d]^{2t}$  collectively have distinct indices) while the square vertices in each block is  $S$  (up to ordering). There are  $\binom{d}{2t\ell} (2t\ell)! \approx d^{2t\ell}$  choices for the circle vertices (as  $d \gg 2t\ell$ ). Since each circle vertex is distinct, the two adjacent square vertices (in neighboring blocks) must be the same so that each edge appears twice. Note that the square vertices in each block can be reordered. Thus, for the circle-to-square structure in each block, viewing it as a matching between  $2t$  circle vertices, there is exactly one matching that results in an even labeling. The probability is  $\frac{1}{(2t-1)!!} \geq t^{-t}$ . Thus, these labelings contribute a lower bound of  $d^{2t\ell} \cdot t^{-t\ell} = (\frac{d^2}{t})^{t\ell}$ .

Next, consider the labelings where all circle vertices are the same while the square vertices are distinct. There are  $\approx n^{t\ell}$  choices for the square vertices. Moreover, there are  $t\ell$  distinct edges, each appearing 4 times, hence giving a factor of  $\mu_4^{t\ell}$ . Thus, these contribute a lower bound of  $(\mu_4 n)^{t\ell}$ .

**Remark III.13** (A higher lower bound on  $\mathbb{E} \text{tr}(\tilde{M}^\ell)$  without requirement (3)). Fix the identity permutation structure (Figure 1a). Consider the labeling such that for each square vertex, the two circle vertices on each side get the same labels (thus violating requirement (3)). Then, we assign distinct labels to all square vertices and all pairs of circle vertices, for which there are  $\approx d^{t\ell} n^{t\ell}$  choices. After permutation, i.e., randomizing the structure, with probability  $(\frac{1}{(2t-1)!!})^{2\ell} \geq t^{-2t\ell}$  each pair of circle vertices are still paired, in which case all edges appear twice, satisfying requirement (1). Thus, this gives a lower bound of  $\Omega(\frac{dn}{t^2})^{t\ell}$ , which is much larger than the target bound  $O(n + \frac{d^2}{t})^{t\ell}$  in Lemma III.9.

Now, we can bound  $\mathbb{E} \text{tr}(\tilde{M}^\ell)$  as follows,

$$\begin{aligned} & \mathbb{E}_A \text{tr}(\tilde{M}^\ell) \\ & \leq 2^{O(t\ell)} \cdot \mathbb{E}_\pi \sum_{\sigma: \text{labeling}} \mathbf{1}(\sigma \text{ valid for } \pi) \cdot \text{edge-factor}(\sigma, \pi). \end{aligned}$$

Here, the  $2^{O(t\ell)}$  factor takes care of the fact that  $\text{Shift}(a^{\otimes 4})$  has some entries scaled by  $3/2$ .

The crucial step in bounding the above is defining an encoding of the labelings. Following the terminology of [44], we will refer to a circle-to-square or square-to-circle step in the walk (structure) as a *leg*, and we will refer to an element in  $[n] \times [d]$  as an *edge*. In other words, a labeling of a leg in the structure specifies an edge. An example is shown in Figure 2.

**Remark III.14** (Encoding requirement). We would like an encoding scheme that uses as few bits as possible such that (1) every valid labeling for a given structure can be encoded, and (2) a labeling is uniquely determined given an encoding and a structure (if decoding succeeds). In particular, the encoding itself does not need to identify the structure; instead, the decoding algorithm will take both the encoding and the structure to decode. Thus, when provided with the same encoding and two distinct structures, the decoding algorithm can successfully output two different labelings, each valid for its respective structure.

**Definition III.15** (Encoding of a labeling). We define our encoding of a valid labeling for a structure  $\pi$  as follows. First, specify a starting index  $J \in [d]^{2t}$ . Then, for each of the  $4t\ell$  legs ( $2t$  circle to square and  $2t$  square to circle in each block), specify a *type* in  $\{F, P, R, N, H\}$  along with additional information for decoding (namely, **destination** and **return labels** depending on the type):

- **F (Fresh) leg**: the destination is a new vertex (not seen before), thus creating a new edge, marked as *active*. An F leg from circle to square must be paired with a P leg as described next. The new vertex is specified as follows:
  - Circle to square: an element in  $[n]$ .
  - Square to circle: an element in  $[d]$  and a *bucket index*  $b \in [\log n]$  for the circle vertex. The bucket index indicates that the vertex will have degree between  $2^b$  and  $2^{b+1}$  in total.
- **P (Paired) leg**: it must be from a circle to a new square vertex and must be paired with an F leg. There is no need to specify a vertex label because the vertex is determined by the paired F leg. The P legs are further split into two types:
  - **P<sub>1</sub>**: the paired F and P legs come from *different* circle vertices. It marks the edge as active.
  - **P<sub>2</sub>**: the paired F and P legs come from the *same* circle vertex. By requirement (3) of Definition III.11, one or two of the next legs leaving the square

vertex must be an R going back to the circle vertex.

In addition, specify a return label in  $\{1, 2\}$ .

- R (Return) leg: it traverses an incident active edge and marks it as *closed*. The incident edge will be chosen (during decoding) using return labels from previous legs.
- N (Non-innovative) leg: the destination is an old vertex (seen before), but the edge is new and is marked as active.
  - Circle to square: specify a previous square vertex (a label in  $[t\ell]$ ), and then specify a return label in  $[4\ell]$  (because square labels in a layer are distinct).
  - Square to circle: pick a bucket index  $b \in [\log n]$  and choose a vertex from the bucket by specifying an element in  $[\frac{4t\ell}{2^b}]$ . Then, specify a return label in  $[2^{b+1}]$ .
- H (High-multiplicity): it traverses an incident *closed* edge and marks it as *active*, indicating that the edge is traversed an odd number of times.
  - Circle to square: specify an incident edge (a label in  $[4t\ell]$ ), and then specify a return label in  $[\ell]$ .
  - Square to circle: same as N.

Finally, let  $\zeta$  be an encoding with  $h$  number of H legs and  $p_2$  number of  $P_2$  legs. With a slight abuse of notation, we define *edge-factor*( $\zeta$ ) as  $\mu_4^{p_2} (C \log n)^h$ , where  $\mu_4$  and  $C$  are the constants such that our random variables satisfy  $\mathbb{E}[a_{ij}^4] \leq \mu_4$  and  $|a_{ij}| \leq \sqrt{C \log n}$  almost surely.

See Figure 2 for examples of an encoding. We note that the terms “fresh”, “return”, “non-innovative” and “high-multiplicity” in Definition III.15 are adopted from [44]<sup>3</sup> (see Section II-D). The P (Paired) type is introduced to handle the specific structure of our matrix, namely that there are two legs leading to a square.

**Remark III.16** ( $P_1$  and  $P_2$  legs). As each  $P_1$  leg creates a new edge, it should be viewed as an F leg except that it does not need to specify the destination.

On the other hand, a  $P_2$  leg should be viewed as an H leg which is closed immediately when departing from the square. When two circle vertices with the same label  $j \in [d]$  go to a square  $i \in [n]$  via an F and  $P_2$  leg, the edge  $(\boxed{i}, \circledast j)$  is traversed twice. However, by requirement (3), one or two of the next legs leaving the square vertex must be R legs going back to  $j$ , so after this the edge is traversed either 3 or 4 times. Thus, the  $P_2$  leg is effectively an H leg that traverses the edge the *third* time, which introduces an edge factor of  $\mathbb{E}[a_{ij}^4] \leq \mu_4$ . See Remark III.13 for a lower bound without requirement (3).

It is helpful for readers to keep in mind that eventually the dominating terms in the trace calculation will be encodings with mostly F (or  $\{F, P\}$  if leading to a square)

<sup>3</sup> In [28], a non-innovative leg is called a “surprise”.

and R legs. In this case, most R legs are “forced”, i.e., there is only one edge to choose. In fact, the only times an R leg is “unforced” at a vertex  $v$  (i.e., it needs to choose between multiple incident edges) are when there were previous N or H legs going back to  $v$ . When this happens, the destination of the R leg is determined by one return label among those placed by previous N and H legs. We now need to show that the return labels in our encoding are sufficient to guide the “unforced” return legs.

a) *Unforced return legs.*: Again following the terminology in [44], we say that an R (return) leg from a vertex  $v$  is *forced* if it marks the final visit to  $v$  in the walk. Consequently, following this leg, all edges incident to  $v$  are closed for the remainder of the walk. Every circle vertex has exactly 1 forced return, and every square vertex has 2 (the two legs leaving  $v$  in its final appearance). All other return legs are *unforced*.

In Definition III.15, each N and H leg arriving at  $v$  provides 1 return label for  $v$ . The next lemma shows that this is sufficient for all unforced returns departing from  $v$ .

**Lemma III.17** (Bounding unforced returns). *For each vertex  $v$ , the number of unforced return legs departing from  $v$  is at most the number of N and H legs arriving at  $v$ . Consequently, the return labels in Definition III.15 are sufficient for all unforced returns.*

*Proof.* Let  $f_{in}, n_{in}, h_{in}, r_{in}$  be the number of F, N, H, R steps arriving at  $v$ , and  $f_{out}, n_{out}, h_{out}, r_{out}$  be the number of F, N, H, R steps departing from  $v$ . For circle vertices, there may be  $p_{out}$  out-going P legs. We know that (1) the number of in-going and out-going edges are the same:  $f_{in} + n_{in} + h_{in} + r_{in} = f_{out} + p_{out} + n_{out} + h_{out} + r_{out}$ , and (2) all active edges need to be closed:  $r_{in} + r_{out} = f_{in} + f_{out} + p_{out} + n_{in} + n_{out} + h_{in} + h_{out}$ . Combining the two, we get

$$\begin{aligned} r_{in} + r_{out} &= 2(f_{in} + n_{in} + h_{in}) + r_{in} - r_{out} \\ \implies r_{out} &= f_{in} + n_{in} + h_{in}. \end{aligned}$$

For circle vertices,  $f_{in} = 1$  (the first arrival at  $v$ ), while there is 1 forced return from  $v$ . So, there are  $n_{in} + h_{out}$  number of unforced returns from  $v$ .

Now, if  $v$  is a square vertex, there is exactly 1 incoming F and 1 P leg (the first arrival at  $v$ ; all subsequent arrivals are N, H or R by definition), meaning  $f_{in} + p_{in} = 2$ . The same calculation shows that  $r_{out} = f_{in} + p_{in} + n_{in} + h_{in}$ . There are 2 forced returns from  $v$ , so again there are  $n_{in} + h_{out}$  number of unforced returns from  $v$ .  $\square$

We remark that this generalized definition of forced/unforced returns was used in [28] as well to bound walks that have complicated structures. In fact, Lemma III.17 is essentially the Potential-Unforced-Return (PUR) factor as defined in [28].

#### D. Decoding

Given an encoding and a structure, we will decode the labeling step by step from left to right. Each edge is either active or closed during the decoding process, and all edges must be closed in the end (so that requirement (1) of Definition III.11 is satisfied).

**Algorithm 1** (Decoding algorithm Decode).

**Input:** A structure  $\pi \in (\hat{\mathcal{S}}_{2t})^{2\ell}$ , an encoding  $\xi$ .

**Output:** A labeling of the structure, or FAIL.

**Operation:** We will label the walk (the graph determined by  $\pi$ ) one vertex at a time. It is convenient to view the algorithm as dynamically maintaining a separate bipartite graph between labeled square and circle vertices (elements in  $[n]$  and  $[d]$  respectively), where a leg in the walk may create a new vertex and/or edge, and each edge has either “active” or “closed” status. Each vertex stores a list of return labels. Moreover, each circle vertex is assigned a bucket index  $b$ .

- $\{F, P\}$  legs from circle to square: label the square vertex with the specified label, and mark the edges as active. Place down the return label from the P leg.
- F leg from square to circle: label the circle vertex with the specified label, and mark the edge as active. Then, label the circle vertex with the specified bucket index.
- N or H leg: label the vertex by the specified previous vertex, and mark the edge as active. Then, place down the return label.
- R leg: use a return label to choose an active edge incident to the current vertex to close. In the special case that it is a square to circle leg and the previous visit to the square is via  $\{F, P_2\}$  from two circle vertices both labeled  $j$ , then simply set  $j$  as the destination. If there is no return label, there must be only one choice (otherwise output FAIL).
- For every square vertex, unless it is reached via  $\{F, P\}$  legs, it gets two labels from the two legs. Output FAIL if the two labels differ. Also output FAIL if the labels assigned to the  $t$  square vertices in the same block are not distinct.

Figure 2 shows an example where we decode the same encoding with different structures. The first one (Figure 2a) succeeds; in fact, for the part shown in Figure 2a, the resulting labeling satisfies all requirements of a valid labeling (Definition III.11). The second one fails because for the square vertex (marked with an “x”), the

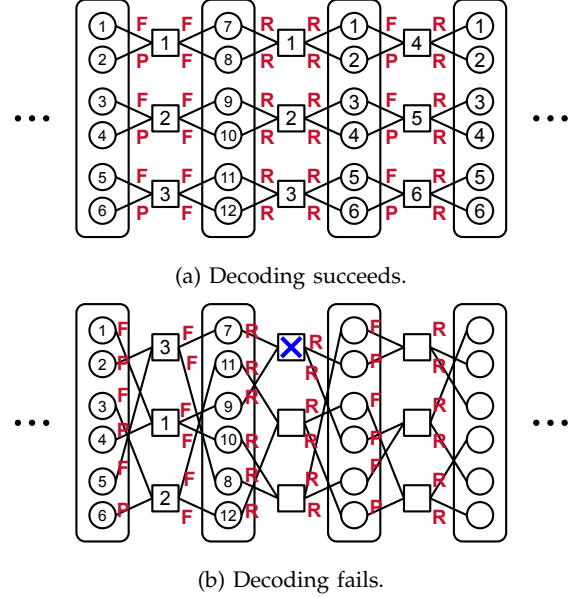


Fig. 2: An example where we decode the same encoding with different structures.

two legs give conflicting labels: the first one labels it 3 to close the edge  $(\boxed{3}, \textcircled{7})$  while the second one labels it 1 to close the edge  $(\boxed{1}, \textcircled{9})$ .

It is crucial that any valid labeling is captured by some encoding, so that we can bound the sum of labelings by the sum of encodings.

**Lemma III.18.** *For any labeling  $\sigma$  which is valid for some structure  $\pi \in (\hat{\mathcal{S}}_{2t})^{2\ell}$ , there is an encoding  $\xi$  such that  $\text{Decode}(\xi, \pi)$  successfully outputs  $\sigma$ . Moreover,  $\text{edge-factor}(\xi) \geq \text{edge-factor}(\sigma, \pi)$ .*

*Proof.* Given a valid labeling for a structure, the  $4t\ell$  leg types and the specification of vertices (a new vertex for F, a previous location for N, and an incident edge for H) are straightforward to determine. For circle vertices, the bucket index  $b \in [\log n]$  is specified such that the total number of edges incident to the vertex is between  $2^b$  and  $2^{b+1}$ . Here  $b \leq \log n$  since  $4t\ell \ll 2^{\log n} = n$ . In particular, the number of circle vertices in bucket  $b$  is at most  $\frac{4t\ell}{2^b}$ .

The return labels are also straightforward. As shown in Lemma III.17, one return label for each N or H is enough to guide all unforced returns. Thus, we can specify the return labels arriving at a vertex  $v$  according to the ordering in which active edges incident to  $v$  are closed. A circle vertex with bucket index  $b$  has at most  $2^{b+1}$  incident edges, whereas a square vertex has at most  $4\ell$  incident edges (due to requirement (2) of Definition III.11 that the square vertices in each block are distinct). For a square vertex  $v$ , two departing R legs are taken to close the final two active edges during the last visit to  $v$ . The return label for the P leg (an element in

$\{1, 2\}$ ) that initially creates  $v$  is decided by the ordering of these final two R legs. Therefore, the ranges of return labels as defined in the encoding are sufficient.

For the edge factors, consider an edge traversed  $2k \geq 2$  times in the walk. If the edge is created by an  $\{F, P_2\}$  pair, then as discussed in Remark III.16, we may view the  $P_2$  leg as the third time the edge is traversed. In this case, the edge will be traversed by 1 F, 1  $P_2$ ,  $(k-2)$  H and  $k$  R legs. Then, the edge factor of  $\xi$  from this edge will be  $\mu_4 \cdot (C \log n)^{k-2}$ . This is a valid upper bound as  $\mathbb{E}[a^{2k}] \leq \mathbb{E}[a^4] \cdot (\sqrt{C \log n})^{2k-4} \leq \mu_4 \cdot (C \log n)^{k-2}$ .

On the other hand, in other cases the edge will be traversed by 1 F,  $(k-1)$  H and  $k$  R legs, giving an edge factor of  $(C \log n)^{k-1}$ . This is also a valid upper bound as  $\mathbb{E}[a^{2k}] \leq \mathbb{E}[a^2] \cdot (\sqrt{C \log n})^{2k-2} \leq (C \log n)^{k-1}$ . Therefore,  $\text{edge-factor}(\xi)$  as defined in Definition III.15 is a valid upper bound on  $\text{edge-factor}(\sigma, \pi)$ .  $\square$

The next lemma states two simple but important relationships between the number of legs of each type in a valid encoding.

**Lemma III.19.** *Let  $\xi$  be an encoding such that  $\text{Decode}(\xi, \pi)$  succeeds for some  $\pi$ . Let  $f_s, p_s, n_s, h_s, r_s$  be the number of F, P, N, H, R legs from circle to square, let  $f_c, n_c, h_c, r_c$  be the number of F, N, H, R legs from square to circle, and let  $n = n_s + n_c$ ,  $h = h_s + h_c$  and  $r = r_s + r_c$ . Then, we must have*

- (1)  $r = 2f_s + f_c + n + h = 2t\ell$ .
- (2)  $n_s + h_s + r_s = f_c + n + h$ .

*Proof.* First, recall from the encoding (Definition III.15) that each P must be paired with an F from circle to square, so  $p_s = f_s$ . Next, each F, P, N and H leg creates an active edge that needs to be closed by an R leg, thus

$$r = f + p_s + n + h = 2f_s + f_c + n + h. \quad (7)$$

Next, the total number of edges is  $4t\ell$ , so  $2f_s + f_c + n + h + r = 4t\ell$ , which means that  $r = 2f_s + f_c + n + h = 2t\ell$ .

Moreover, as the number of edges from circle to square is the same as that from square to circle, we have

$$2f_s + n_s + h_s + r_s = f_c + n_c + h_c + r_c.$$

Combined with Eq. (7), we get  $r_s + r_c = 2(f_c + n_c + h_c) + (r_c - r_s)$ , which means  $r_s = f_c + n_c + h_c$ . Thus,  $n_s + h_s + r_s = f_c + n + h$ .  $\square$

We next prove the crucial lemma upper bounding the success probability of decoding over random permutations of the structure.

**Lemma III.20** (Decoding success probability). *Let  $\xi$  be an encoding, let  $f_c$  be the number of F legs from square to circle, and let  $n, h$  be the number of N, H legs. Then,*

$$\Pr_{\pi \sim (\mathcal{S}_{2t})^{2\ell}} [\text{Decode}(\xi, \pi) \text{ succeeds}] \leq \left(\frac{t}{e}\right)^{-\frac{1}{2}(f_c + n + h)}.$$

*Proof.* Let  $f_s, p_s, n_s, h_s, r_s$  be the number of F, P, N, H, R legs from circle to square, and let  $f_c, n_c, h_c, r_c$  be the number of F, N, H, R legs from square to circle. Consider the circle-to-square part of each block, and view the structure as a perfect matching between  $2t$  circle vertices. The F and P legs must be paired and the  $t$  square vertices must receive distinct labels. Moreover, all N, H, R legs from circle to square provide labels, and recall that Algorithm 1 fails if any square vertex receives two different labels. Therefore, there is exactly one way to pair up all the N, H, R legs.

Suppose there are  $i \leq t$  number of F,  $i$  number of P (because F and P must be paired), and  $2t - 2i$  other legs. There are  $(2t - 1)!! \geq \left(\frac{2t}{e}\right)^t$  number of ways to group them into pairs. However, there are  $i!$  ways to pair the F and P legs, and there is a unique way to pair the other legs. Thus, via Stirling's approximation, the probability that a grouping is valid is

$$\frac{i!}{(2t - 1)!!} \leq i! \left(\frac{2t}{e}\right)^{-t} \leq \left(\frac{t}{e}\right)^{-(t-i)}.$$

Note that  $t - i$  is 1/2 of the number of N, H and R legs from circle to square in the block. Thus, overall, the probability of success is upper bounded by  $(t/e)^{-\frac{1}{2}(n_s + h_s + r_s)}$ . By Lemma III.19, we have  $n_s + h_s + r_s = f_c + n + h$ , thus completing the proof.  $\square$

*E. Proof of Lemma III.9*

We can finally finish the proof of Lemma III.9.

*Proof of Lemma III.9.* We start with writing the trace as a sum of encodings:

$$\begin{aligned} \mathbb{E}_A \text{tr}(\tilde{M}^\ell) &\leq \mathbb{E}_{\pi \sim (\mathcal{S}_{2t})^{2\ell}} \sum_{\sigma: \text{labeling}} \mathbf{1}(\sigma \text{ valid for } \pi) \cdot \text{edge-factor}(\sigma, \pi) \\ &\leq \mathbb{E}_{\pi \sim (\mathcal{S}_{2t})^{2\ell}} \sum_{\sigma: \text{labeling}} \sum_{\xi: \text{encoding}} \mathbf{1}(\text{Decode}(\xi, \pi) = \sigma \text{ \& valid for } \pi) \\ &\quad \cdot \text{edge-factor}(\xi) \\ &\leq \sum_{\xi: \text{encoding}} \Pr_{\pi \sim (\mathcal{S}_{2t})^{2\ell}} [\text{Decode}(\xi, \pi) \text{ succeeds}] \cdot \text{edge-factor}(\xi). \end{aligned}$$

Here, the second inequality follows from Lemma III.18 that every labeling  $\sigma$  has an encoding  $\xi$  such that  $\text{Decode}(\xi, \pi)$  successfully outputs  $\sigma$ , and that  $\text{edge-factor}(\sigma, \pi) \leq \text{edge-factor}(\xi)$ .

We bound the number of encodings as follows:

- For each  $\{F, P\}$  leading to a square, there are  $n$  choices for the new vertex, and 2 choices for the return label of P. In total,  $2n$  choices.
- For each F from square to circle, there are  $d$  choices for the new vertex and  $\log n$  choices for the bucket index. In total,  $d \log n$  choices.
- For each N from circle to square, there are  $t\ell$  choices for a previous square vertex, and  $4\ell$  choices for the return label. In total,  $4t\ell^2$  choices.

- For each N or H from square to circle, there are  $\log n$  choices to choose a bucket index  $b$ , and  $\frac{4t\ell}{2^b}$  choices for a vertex in bucket  $b$ . Then, there are  $2^{b+1}$  choices for a return label. In total,  $\log n \cdot \frac{4t\ell}{2^b} \cdot 2^{b+1} = 8t\ell \log n$  choices.
- For each H from circle to square, there are  $4t\ell$  choices to choose an incident edge, and  $4\ell$  choices for a return label. In total,  $16t\ell^2$  choices.
- **Edge factor:** Each H gets a  $C \log n$  factor, and each  $P_2$  gets a  $\mu_4$  factor. Recall that  $\mu_4$  and  $C$  are the constants such that  $\mathbb{E}[a_{ij}^4] \leq \mu_4$  and  $|a_{ij}| \leq \sqrt{C \log n}$  almost surely.

Fix a leg type pattern and starting index in  $[d]^{2t}$  (labels for the first  $2t$  circle vertices in the structure), and let  $f_s, p_s, n_s, h_s, r_s$  be the number of F, P, N, H, R legs from circle to square, and let  $f_c, n_c, h_c, r_c$  be the number of F, N, H, R legs from square to circle. Moreover, let  $n = n_s + n_c$  and  $h = h_s + h_c$ . Then, the total encodings, weighted by edge-factor, is bounded by

$$\begin{aligned} & n^{f_s} (2\mu_4)^{p_s} (d \log n)^{f_c} (4t\ell^2)^{n_s} (8t\ell \log n)^{n_c} \\ & \cdot (16Ct\ell^2 \log n)^{h_s} (8Ct\ell \log^2 n)^{h_c} \\ & \leq (2\mu_4 n)^{f_s} (d \log n)^{f_c} (8t \log^2 n)^n (16Ct \log^3 n)^h, \end{aligned}$$

where we use the fact that  $p_s = f_s$ ,  $\mu_4 \geq 1$  and  $\ell \leq \log n$ .

Next, by Lemma III.20,  $\Pr_{\pi}[\text{Decode}(\zeta, \pi) \text{ succeeds}] \leq (t/e)^{-\frac{1}{2}(f_c + n + h)}$ . Thus, we get

$$\begin{aligned} & (2\mu_4 n)^{f_s} \left( \sqrt{\frac{e}{t}} d \log n \right)^{f_c} O(\sqrt{t} \log^3 n)^{n+h} \\ & \leq O\left(\mu_4 n + \frac{d^2}{t} \log^2 n\right)^{f_s + \frac{1}{2}f_c} O(\sqrt{t} \log^3 n)^{n+h} \end{aligned}$$

By Lemma III.19,  $2f_s + f_c = 2t\ell - n - h$ . Thus, as long as  $t \log^6 n \leq \mu_4 n + \frac{d^2}{t} \log^2 n$ , the above is bounded by  $O(\mu_4 n + \frac{d^2}{t} \log^2 n)^{t\ell}$ .

Finally, there are at most  $2^{O(t\ell)}$  leg type patterns, and  $d^{2t}$  choices for the starting index  $J \in [d]^{2t}$  in the encoding. This gives the final bound of  $d^{2t} \cdot O(\mu_4 n + \frac{d^2}{t} \log^2 n)^{t\ell}$ .  $\square$

#### IV. PLANTED SPARSE VECTOR

We first restate the planted sparse vector model.

**Model** (Restatement of Model 1). Fix an unknown unit vector  $v \in \mathbb{R}^n$ , and let  $d \leq n \in \mathbb{N}$ . Let  $\tilde{A}$  be a random  $n \times d$  matrix sampled as follows: (1) let  $A$  be the random matrix such that the first column is  $v$  and the other  $d-1$  columns are i.i.d.  $\mathcal{N}(0, \frac{1}{n} \mathbb{I}_n)$  vectors; (2) let  $R \in \mathbb{R}^{d \times d}$  be an arbitrary unknown rotation matrix; (3) set  $\tilde{A} = AR$ .

The task is that given  $\tilde{A}$ , output a unit vector  $\hat{v} \in \mathbb{R}^n$  such that  $\langle \hat{v}, v \rangle^2 \geq 1 - o(1)$ .

We first characterize the sparse vectors that we will plant in Model 1. Motivated by the definition of well-spread subspaces (Definition III.1), we define the notion of compressible vectors as follows,

**Definition IV.1** (Compressible vector). Let  $\rho, \gamma \in (0, 1)$ . We say that a vector  $v \in \mathbb{R}^n$  is  $(\rho, \gamma)$ -compressible if

$$\max_{S \subseteq [n]: |S| \leq \rho n} \|v_S\|_2 = (1 - \gamma) \|v\|_2.$$

In other words,  $v$  is compressible if the top  $\rho n$  coordinates of  $v$  contains  $1 - \gamma$  fraction of its  $\ell_2$ -mass.

Next, recall our notation for the elementary symmetric polynomial (of the 4th powers):

$$Q_t(z) := t! \sum_{S \subseteq [n]: |S|=t} \prod_{i \in S} z_i^4.$$

We now state our main result.

**Theorem IV.2** (Formal version of Theorem 2). *There is an absolute positive constant  $C$  such that for every  $t \leq d \leq n \in \mathbb{N}$  such that  $\frac{ed^2}{n} \log^C n \leq t \leq \frac{d}{\log^C n}$  and  $\rho \in (0, \frac{1}{\log^C n})$ , there is a randomized algorithm with running time  $2^{\tilde{O}(t)}$  with the following guarantee: Given  $\tilde{A} \in \mathbb{R}^{n \times d}$  drawn from Model 1 with a hidden unit vector  $v \in \mathbb{R}^n$  such that  $v$  is  $(\rho, 1/2)$ -compressible and  $Q_t(v)^{1/t} \geq (1 - \frac{1}{\log^C n}) \|v\|_4^4$ , the algorithm outputs a unit vector  $\hat{v} \in \text{colspan}(\tilde{A})$  such that  $\langle \hat{v}, v \rangle^2 \geq 1 - o(1)$  with probability  $1 - o(1)$  over the randomness of the algorithm and the input.*

To interpret the parameters in Theorem IV.2, consider  $d = O(n^{\frac{1+\epsilon}{2}})$  and  $t = \tilde{O}(n^\epsilon)$  (same as in Theorem III.3), and the sparsity of  $v$  is  $\rho = 1/\text{polylog}(n)$ . Then, we can approximately recover  $v$  in  $2^{\tilde{O}(n^\epsilon)}$  time. This establishes the same trade-off between the dimension and runtime as our certification algorithm (Theorem 1).

Our algorithm is described in Algorithm 2, and the proof of Theorem IV.2 is completed at the end of Section IV-C.

**Remark IV.3** (Assumptions on  $v$ ). The assumption that  $Q_t(v)^{1/t} \geq (1 - o(1)) \|v\|_4^4$  is not standard. In particular, this does not hold for any vector  $v$  whose mass concentrates on  $< t$  coordinates. However, for such vectors, one can simply brute-force search over all support of size  $\leq \tilde{O}(t)$ , which takes  $2^{\tilde{O}(t)}$  time.

On the other hand, most distributions of sparse vectors used in previous works satisfy that  $Q_t(v)^{1/t} \geq (1 - o(1)) \|v\|_4^4$  (when the sparsity is not small enough to brute-force search over the support). These include the noiseless and noisy Rademacher-Bernoulli distribution (Definition I.1) considered in [10], [12], [39], [47] and the Rademacher-Gaussian distribution considered in [39].

We note that the algorithms of [25], [39] require very minimal assumptions on  $v$ . In particular, the algorithm of [39] succeeds as long as  $\|v\|_4^4 - \frac{3}{n} \geq \Omega(\frac{1}{n})$  and  $\frac{1}{\|v\|_4^4} \frac{d}{n^{3/2}} \ll 1$ ,<sup>4</sup> so when  $\|v\|_4^4 \approx \frac{1}{\rho n}$ , this is equivalent

<sup>4</sup> Here  $\frac{3}{n}$  is exactly  $\mathbb{E}[\|g\|_4^4]$  for a Gaussian vector  $g \sim \mathcal{N}(0, \frac{1}{n} \mathbb{I}_n)$ . There is also a requirement on  $\|v\|_\infty / \|v\|_4^2$  which we omit, as it is satisfied in most cases. See [39] for the full statement.

to  $\rho d \ll \sqrt{n}$ . We leave as future work the identification of a minimal analytic assumption on  $v$  that ensures successful recovery in the subexponential-time regime.

#### A. Preliminaries for the planted sparse vector model

It is easy to see that a compressible vector  $v$  has large 4-norm:

**Lemma IV.4.** *A  $(\rho, \gamma)$ -compressible vector  $v \in \mathbb{R}^n$  satisfies  $\|v\|_4^4 \geq \frac{(1-\gamma)^4}{\rho n} \|v\|_2^2$ .*

*Proof.* Let  $S^* \subseteq [n]$  be such that  $|S^*| \leq \rho n$  and  $\|v_{S^*}\|_2 \geq (1-\gamma)\|v\|_2$ . Then, by Cauchy-Schwarz,  $(1-\gamma)^4 \|v\|_2^4 \leq \|v_{S^*}\|_2^4 \leq |S^*| \cdot \|v_{S^*}\|_4^4 \leq \rho n \cdot \|v\|_4^4$ .  $\square$

The following lemma is also straightforward given the standard singular value concentration bounds on Gaussian matrices (Fact III.5).

**Lemma IV.5.** *Let  $d \leq n/\text{polylog}(n)$ . Let  $A \in \mathbb{R}^{n \times d}$  be drawn from Model 1 with an arbitrary unit vector  $v$ . Then, with probability  $1 - o(1)$ , we have  $1 - o(1) \leq \sigma_{\min}(A) \leq \sigma_{\max}(A) \leq 1 + o(1)$ .*

*Proof.* Recall that the first column of  $A$  is  $v$ , and let  $\bar{A} \in \mathbb{R}^{n \times (d-1)}$  be the rest of the matrix with i.i.d.  $\mathcal{N}(0, 1/n)$  entries. By Fact III.5, we know that  $\sigma_{\max}(\bar{A}) \leq 1 + o(1)$  with high probability. Moreover, since  $v$  is a unit vector,  $\bar{A}^\top v$  is distributed as a  $(d-1)$ -dimensional vector with  $\mathcal{N}(0, 1/n)$  entries, thus with high probability we have  $\|\bar{A}^\top v\|_2^2 \leq O(d/n) \leq o(1)$ .

For any unit vector  $x = (x_1, \bar{x})$ , we have  $Ax = x_1 v + \bar{A}\bar{x}$ , thus

$$\begin{aligned} \|Ax\|_2^2 &= x_1^2 \|v\|_2^2 + \|\bar{A}\bar{x}\|_2^2 + 2x_1 v^\top \bar{A}\bar{x} \\ &\leq x_1^2 \|v\|_2^2 + \|\bar{A}\bar{x}\|_2^2 + 2|x_1| \|\bar{x}\|_2 \|\bar{A}^\top v\|_2 \\ &\leq (1 + o(1)) (x_1^2 + \|\bar{x}\|_2^2) = (1 + o(1)) \|x\|_2^2. \end{aligned}$$

Similarly, the minimum singular value lower bound follows from the fact that  $\sigma_{\min}(\bar{A}) \geq 1 - o(1)$  and  $\|Ax\|_2^2 \geq x_1^2 \|v\|_2^2 + \|\bar{A}\bar{x}\|_2^2 - 2|x_1| \|\bar{x}\|_2 \|\bar{A}^\top v\|_2$ .  $\square$

The following lemma is important in our algorithm to identify a “good” vector to output.

**Lemma IV.6.** *Let  $n, d \in \mathbb{N}$  such that  $d \leq n/\text{polylog}(n)$ , let  $\rho \leq 1/\text{polylog}(n)$ , and let  $\gamma, \delta \in (0, 1)$ . Let  $v \in \mathbb{R}^n$  be a  $(\rho, \gamma)$ -compressible unit vector, and let  $A \in \mathbb{R}^{n \times d}$  be drawn from Model 1 with planted vector  $v$ . Then, with probability  $1 - o(1)$  over  $A$ , the following holds: let  $y \in \mathbb{R}^d$  be any unit vector and let  $\hat{v} = Ay$ ,*

- (1) *If  $\|y - e_1\|_2 \leq \delta$  or  $\|y + e_1\|_2 \leq \delta$ , then  $\max_{S: |S| \leq \rho n} \|\hat{v}_S\|_2 \geq 1 - \gamma - (1 + o(1))\delta$ .*
- (2) *If  $\|y - e_1\|_2 \geq \delta$  and  $\|y + e_1\|_2 \geq \delta$ , then  $\max_{S: |S| \leq \rho n} \|\hat{v}_S\|_2 \leq (1 - \frac{1}{2}\delta^2)(1 - \gamma) + \frac{1}{\text{polylog}(n)}$ .*

*Proof.* Let  $S^* = \arg \max_{S: |S| \leq \rho n} \|v_S\|_2$ , and recall that  $Ae_1 = v$ .

Suppose  $\|y - e_1\|_2 \leq \delta$ . Then,  $\|\hat{v} - v\|_2 = \|A(y - e_1)\|_2 \leq \sigma_{\max}(A)\delta \leq (1 + o(1))\delta$  with high probability by Lemma IV.5. By the triangle inequality,  $\|\hat{v}_{S^*}\|_2 \geq \|v_{S^*}\|_2 - \|\hat{v}_{S^*} - v_{S^*}\|_2 \geq 1 - \gamma - (1 + o(1))\delta$  since  $\|v_{S^*} - \hat{v}_{S^*}\|_2 \leq \|v - \hat{v}\|_2 \leq (1 + o(1))\delta$ . The same is true if we flip the sign of  $y$  so that  $\hat{v}$  is close to  $-v$ . This proves the first statement.

Suppose  $\|y - e_1\|_2 \geq \delta$  and  $\|y + e_1\|_2 \geq \delta$ . Denote  $y = (y_1, \bar{y})$ , where  $\bar{y} \in \mathbb{R}^{d-1}$  and  $\|\bar{y}\|_2^2 = 1 - y_1^2$ . Then, we have  $\delta^2 \leq (1 - y_1)^2 + \|\bar{y}\|_2^2 = 2 - 2y_1$  and  $\delta^2 \leq (1 + y_1)^2 + \|\bar{y}\|_2^2 = 2 + 2y_1$ , which means that  $|y_1| \leq 1 - \delta^2/2$ . For any  $S \subseteq [n]$  with  $|S| \leq \rho n$ , we have  $\sigma_{\max}(\bar{A}_S) \leq \tilde{O}(\sqrt{d/n} + \sqrt{\rho}) \leq 1/\text{polylog}(n)$  by Fact III.5 and a union bound over all  $S$  (here we need  $\rho, d/n \leq \frac{1}{\text{polylog}(n)}$ ). Then, since  $\hat{v}_S = y_1 v_S + \bar{A}_S \bar{y}$ , by the triangle inequality,  $\|\hat{v}_S\|_2 \leq |y_1| \|v_S\|_2 + \|\bar{A}_S \bar{y}\|_2 \leq (1 - \frac{1}{2}\delta^2) \|v_{S^*}\|_2 + \frac{1}{\text{polylog}(n)}$ , which proves the second statement.  $\square$

#### B. Background on Sum-of-Squares

In this section, we give an overview of the Sum-of-Squares (SoS) framework. We refer the reader to the monograph [19] and the lecture notes [6] for a detailed exposition of the SoS method and its usage in algorithm design.

a) *Pseudo-distributions.*: Pseudo-distributions are generalizations of probability distributions and are represented by their *pseudo-expectation* operators. Formally, a degree- $t$  pseudo-distribution  $\mu$  over variables  $x_1, \dots, x_n$  corresponds to a linear operator  $\tilde{\mathbb{E}}_\mu$  that maps polynomials of degree  $\leq t$  to real numbers and satisfies  $\tilde{\mathbb{E}}_\mu[1] = 1$  and  $\tilde{\mathbb{E}}_\mu[p^2] \geq 0$  for every polynomial  $p(x_1, \dots, x_n)$  of degree  $\leq t/2$ .

Let  $\mathcal{A} = \{f_1 \geq 0, f_2 \geq 0, \dots, f_m \geq 0\}$  be a system of  $m$  polynomial inequality constraints. We say that  $\mu$  satisfies the system of constraints  $\mathcal{A}$  at degree  $t$  if for every sum-of-squares polynomial  $h$  and any  $T \subseteq [m]$  such that  $\deg(h) + \sum_{i \in T} \deg(f_i) \leq t$ ,  $\tilde{\mathbb{E}}_\mu[h \cdot \prod_{i \in T} f_i] \geq 0$ . In particular, if  $\mathcal{A}$  contains an equality constraint  $f = 0$ , then  $\tilde{\mathbb{E}}_\mu[f \cdot p] = 0$  for any polynomial  $p$  with  $\deg(f) + \deg(p) \leq t$ . Specific to our application, we say that a pseudo-distribution  $\mu$  satisfies the *unit sphere* constraints if  $\tilde{\mathbb{E}}_\mu[\|x\|_2^2 \cdot p(x)] = \tilde{\mathbb{E}}_\mu[p(x)]$  for every  $p$  of degree  $\leq t - 2$ .

Unlike true distributions, there is a  $n^{O(t)}$ -time weak separation oracle for degree- $t$  pseudo-distributions, which allows us to efficiently optimize over pseudo-distributions that satisfy a given set of polynomial constraints (approximately) via the ellipsoid method — this is called the Sum-of-Squares algorithm. Therefore, given a polynomial  $p$  (with the  $\ell_1$ -norm of the coefficients being  $\|p\|_1$ ) over  $x_1, \dots, x_n$ , a degree- $t$  pseudo-distribution satisfying the unit sphere constraint that maximizes  $p$  within an additive  $\epsilon \|p\|_1$  error can be found in time  $n^{O(t)} \text{polylog}(n/\epsilon)$ .

b) *Sum-of-Squares proofs.*: Let  $f_1, f_2, \dots, f_m$  and  $g$  be multivariate polynomials in  $x$ . A *sum-of-squares* proof that the constraints  $\mathcal{A} = \{f_1 \geq 0, \dots, f_m \geq 0\}$  imply  $g \geq 0$  consists of sum-of-squares polynomials  $(p_S)_{S \subseteq [m]}$  such that  $g(x) = \sum_{S \subseteq [m]} p_S(x) \prod_{i \in S} f_i(x)$ . The *degree* of such an SoS proof equals the maximum of the degree of  $p_S \prod_{i \in S} f_i$  over all  $S$  appearing in the sum above. We write

$$\mathcal{A} \Big|_{\frac{x}{t}} \{g(x) \geq 0\}$$

where  $t$  is the degree of the SoS proof.

The following fact is the crucial connection between SoS proofs and pseudo-distributions.

**Fact IV.7.** Suppose  $\mathcal{A} \Big|_{\frac{x}{t}} \{g(x) \geq 0\}$  for some polynomial constraints  $\mathcal{A}$  and a polynomial  $g$ . Let  $\mu$  be any pseudo-distribution of degree  $\geq t$  satisfying  $\mathcal{A}$ . Then,  $\tilde{\mathbb{E}}_\mu[g] \geq 0$ .

In other words, for polynomials  $f, g$ , in order to prove that  $\tilde{\mathbb{E}}_\mu[f] \leq \tilde{\mathbb{E}}_\mu[g]$  for any degree- $t$  pseudo-distribution  $\mu$  satisfying  $\mathcal{A}$ , it suffices to show an SoS proof that  $\mathcal{A} \Big|_{\frac{x}{t}} \{f(x) \leq g(x)\}$ .

We will need the following SoS versions of Cauchy-Schwarz and Hölder's inequalities.

**Fact IV.8** (Fact 3.9 and 3.11 in [40]). For any  $\delta > 0$ ,

$$\begin{aligned} & \bullet \left| \frac{x,y}{2} \right\{ xy \leq \frac{\delta}{2} x^2 + \frac{1}{2\delta} y^2 \}. \\ & \bullet \left| \frac{x,y}{4} \right\{ x^3 y \leq \frac{3\delta}{4} x^4 + \frac{1}{4\delta^3} y^4 \}. \end{aligned}$$

Using Fact IV.8, we can prove the following lemma, which is basically the SoS version of Jensen's inequality applied to the  $x^4$  function.

**Lemma IV.9.** For any  $\delta > 0$ ,

$$\left| \frac{x,y}{4} \right\{ (x+y)^4 \leq (1+\delta)^3 \left( x^4 + \frac{y^4}{\delta^3} \right) \}.$$

*Proof.* We expand  $(x+y)^4 = x^4 + 4x^3y + 6x^2y^2 + 4xy^3 + y^4$  and apply Fact IV.8 to the middle 3 terms:  $(x+y)^4 \leq x^4 + (3\delta x^4 + \frac{1}{\delta^3} y^4) + (3\delta^2 x^4 + \frac{3}{\delta^2} y^4) + (\delta^3 x^4 + \frac{3}{\delta} y^4) + y^4 = (1+\delta)^3 x^4 + (1+\frac{1}{\delta})^3 y^4$ .  $\square$

c) *Rounding.*: In the following, we state an important technique for rounding pseudo-distributions from [5].

**Lemma IV.10** (Theorem 5.1 of [5]). For every even  $t \in \mathbb{N}$  and  $\varepsilon \in (0,1)$ , there exists a randomized algorithm with running time  $n^{O(t)}$  and success probability  $2^{-t/\text{poly}(\varepsilon)}$  for the following problem: Fix an unknown unit vector  $q \in \mathbb{R}^n$ . Given a degree- $t$  pseudo-distribution  $\mu$  over  $x \in \mathbb{R}^n$  satisfying the constraint  $\|x\|_2^2 = 1$  such that  $\tilde{\mathbb{E}}_\mu[\langle q, x \rangle^t] \geq e^{-\varepsilon t}$ , output a unit vector  $\hat{x} \in \mathbb{R}^n$  with  $\langle q, \hat{x} \rangle \geq 1 - O(\varepsilon)$ .

C. Algorithm for planted sparse vector

**Algorithm 2** (Recover hidden sparse vector).

**Input:** A matrix  $\tilde{A} \in \mathbb{R}^{n \times d}$  drawn from Model 1,

parameters  $t \in \mathbb{N}$  and  $\rho \in (0,1)$ .

**Output:** A unit vector  $\hat{v} \in \text{colspan}(\tilde{A})$ .

**Operation:**

- 1) Solve the degree- $4t$  SoS relaxation of the following program over  $x \in \mathbb{R}^d$ ;

$$\begin{aligned} \max \quad & \tilde{P}_t(x) := t! \sum_{S \subseteq [n]: |S|=t} \prod_{i \in S} \langle \tilde{a}_i, x \rangle^4 \\ \text{s.t.} \quad & \|x\|_2^2 = 1 \end{aligned}$$

where  $\tilde{a}_1, \dots, \tilde{a}_n$  are the rows in  $\tilde{A}$ .

- 2) Repeat the algorithm of Lemma IV.10  $N = 2^{\tilde{O}(t)}$  times and obtain unit vectors  $\hat{x}^{(1)}, \hat{x}^{(2)}, \dots, \hat{x}^{(N)}$ .
- 3) Let  $\hat{v}^{(i)} = \tilde{A} \hat{x}^{(i)}$ , and let  $j = \arg \max_{i \in [N]} \max_{S \subseteq [n]: |S| \leq \rho n} \|\hat{v}_S^{(i)}\|_2$ , i.e., the vector whose top  $\rho n$  entries have the largest norm. Output  $\hat{v}^{(j)} / \|\hat{v}^{(j)}\|_2$ .

The main ingredient in the analysis is Lemma III.6. Below, we state it as a degree- $4t$  SoS proof.

**Lemma IV.11** (SoS version of Lemma III.6). Let  $t \leq d \leq n$  be integers such that  $t \log^6 n \leq n + \frac{d^2}{t} \log^2 n$ . Let  $A \sim \mathcal{N}(0, \frac{1}{n})^{n \times d}$  with rows  $a_1, \dots, a_n \in \mathbb{R}^d$ . Then, with probability  $1 - \frac{1}{\text{poly}(n)}$  over  $a_1, \dots, a_n$ ,

$$\left| \frac{y}{4t} \right\{ P_t(y) := t! \sum_{S \subseteq [n]: |S|=t} \prod_{i \in S} \langle a_i, y \rangle^4 \leq O\left(\frac{1}{n} + \frac{d^2}{n^2 t} \log^2 n\right)^t \|y\|_2^{4t} \}.$$

Note the scaling of  $1/n$  here because we assume the entries to be  $\mathcal{N}(0, 1/n)$  as opposed to  $\mathcal{N}(0, 1)$  in Lemma III.6.

Using Lemma IV.11, we prove our main lemma.

**Lemma IV.12.** Assume the same setting as Theorem IV.2. Let  $r_1 \in \mathbb{R}^d$  be the first row of the unknown rotation matrix  $R$ . Further, let  $\mu$  be the degree- $4t$  pseudo-distribution over  $x \in \mathbb{R}^n$  with the unit sphere constraint that maximizes  $\tilde{P}_t(x)$ . Then,  $\tilde{\mathbb{E}}_\mu[\langle r_1, x \rangle^{2t}] \geq e^{-O(t/\log n)}$ .

*Proof.* Recall from Model 1 that  $\tilde{A} = AR$  and  $v$  is the first column of  $A$ . Denote  $y = Rx$  such that  $\langle \tilde{a}_i, x \rangle = \langle a_i, Rx \rangle = \langle a_i, y \rangle$  and  $\tilde{P}_t(x) = P_t(y)$ . Note also that  $y_1 = \langle r_1, x \rangle$ . Our goal is to prove that  $\tilde{\mathbb{E}}_\mu[y_1^{2t}]$  is large.

For each row  $a_i$  in  $A$ , we write  $a_i = (v_i, \bar{a}_i)$ , and for simplicity denote  $w_i := \langle \bar{a}_i, \bar{y} \rangle$  such that  $\langle a_i, y \rangle = y_1 v_i + w_i$ . We first apply Lemma IV.9 with  $\delta := \frac{1}{\log^3 n}$  to each  $(y_1 v_i + w_i)^4$ :

$$\left| \frac{y}{4t} \right\{ P_t(y) \leq t! (1+\delta)^{3t} \sum_{S \subseteq [n]: |S|=t} \prod_{i \in S} \left( y_1^4 v_i^4 + \frac{w_i^4}{\delta^3} \right) \}.$$

Next, we expand the above by splitting  $S$  into disjoint



$S'$  and  $S''$ :

$$t!(1+\delta)^{3t} \sum_{s=0}^t \frac{y_1^{4(t-s)}}{\delta^{3s}} \sum_{\substack{S' \subseteq [n]: \\ |S'|=t-s}} \prod_{i \in S'} v_i^4 \sum_{\substack{S'' \subseteq [n] \setminus S': \\ |S''|=s}} \prod_{j \in S''} w_j^4. \quad (8)$$

Recall that  $Q_s(z) := s! \sum_{S \subseteq [n]: |S|=s} \prod_{i \in S} z_i^4$ . For any  $s > 0$  and  $S' \subseteq [n]$  with  $|S'| = t-s$ , by Lemma IV.11, with high probability we have

$$\begin{aligned} \frac{|4s|}{y} \left\{ s! \sum_{\substack{S'' \subseteq [n] \setminus S': \\ |S''|=s}} \prod_{j \in S''} w_j^4 \leq Q_s(w) \right. \\ \left. \leq O\left(\frac{1}{n} + \frac{d^2}{n^2 s} \log^2 n\right)^s \cdot \|\bar{y}\|_2^{4s} \right\}. \end{aligned}$$

Note that in the first inequality, we upper bound the summation over  $S'' \subseteq [n] \setminus S'$  by the summation over  $S \subseteq [n]$ , resulting in  $Q_s(w)$ .

Next,  $(t-s)! \sum_{|S|=t-s} \prod_{i \in S} v_i^4 = Q_{t-s}(v) \leq \|v\|_4^{4(t-s)}$ . Moreover, by Lemma IV.4, we have  $\|v\|_4^4 \geq \Omega(\frac{1}{\rho n})$  since  $v$  is a  $(\rho, 1/2)$ -compressible unit vector. Then, from Eq. (8) we have

$$\begin{aligned} \frac{y}{4t} \left\{ P_t(y) \leq (1+\delta)^{3t} \sum_{s=0}^t \binom{t}{s} \frac{y_1^{4(t-s)}}{\delta^{3s}} \cdot Q_{t-s}(v) Q_s(w) \right. \\ \leq (1+\delta)^{3t} \sum_{s=0}^t \binom{t}{s} y_1^{4(t-s)} \|v\|_4^{4(t-s)} \\ \cdot O\left(\frac{1}{\delta^3 n} + \frac{d^2 \log^2 n}{\delta^3 n^2 s}\right)^s \|\bar{y}\|_2^{4s} \\ \leq (1+\delta)^{3t} \|v\|_4^{4t} \sum_{s=0}^t \binom{t}{s} y_1^{4(t-s)} \\ \cdot O\left(\frac{\rho}{\delta^3} + \frac{\rho d^2 \log^2 n}{\delta^3 n s}\right)^s \|\bar{y}\|_2^{4s} \left. \right\}. \quad (9) \end{aligned}$$

We now take the pseudo-expectation  $\tilde{\mathbb{E}}_\mu$  of both sides above. For simplicity, we denote  $\alpha := \frac{\rho d^2 \log^2 n}{\delta^3 n t} \leq \frac{1}{\log^C n}$  for some large enough constant  $C$  (since  $t \geq \frac{\rho d^2}{n} \text{polylog}(n)$ ). Moreover,  $\mu$  is the pseudo-distribution that maximizes  $\tilde{\mathbb{E}}_\mu[\tilde{P}_t(x)]$ , and in particular the distribution supported on  $x = r_1$  is feasible, so  $\tilde{\mathbb{E}}_\mu[\tilde{P}_t(x)] \geq \tilde{P}_t(r_1) = P_t(e_1) = Q_t(v)$ . Moreover, by assumption we have  $Q_t(v)^{1/t} \geq (1-\eta)\|v\|_4^4$  with  $\eta = \frac{1}{\log^C(n)} \leq \delta$ . Thus, since  $\|\bar{y}\|_2^2 = 1 - y_1^2$  and  $\eta \leq \delta \leq o(1)$ , from Eq. (9) it follows that

$$\begin{aligned} e^{-O(\delta)t} &\leq \frac{(1-\eta)^t}{(1+\delta)^{3t}} \\ &\leq \sum_{s=0}^t \binom{t}{s} O\left(\frac{\rho}{\delta^3} + \frac{\alpha t}{s}\right)^s \tilde{\mathbb{E}}_\mu \left[ y_1^{4(t-s)} (1 - y_1^2)^{2s} \right]. \end{aligned}$$

Thus, there must be an  $s^* \in \{0, 1, \dots, t\}$  such that

$$\begin{aligned} \binom{t}{s^*} O\left(\frac{\rho}{\delta^3} + \frac{\alpha t}{s^*}\right)^{s^*} \tilde{\mathbb{E}}_\mu \left[ y_1^{4(t-s^*)} (1 - y_1^2)^{2s^*} \right] &\geq \frac{1}{t} e^{-O(\delta)t} \\ &\geq e^{-O(\delta)t}. \end{aligned} \quad (10)$$

Here we use the fact that  $\delta t \gg \log t$ .

Let  $\beta := \frac{1}{\log^2 n} \ll 1/2$ . If  $s^* \leq \beta t$ , then  $\binom{t}{s^*} O\left(\frac{\rho}{\delta^3} + \frac{\alpha t}{s^*}\right)^{s^*} \leq t^{O(s^*)} \leq e^{O(\beta t \log n)}$  since  $t \leq n$ . Then, as  $4(t-s^*) \geq 2t$ , we have

$$\begin{aligned} \tilde{\mathbb{E}}_\mu[y_1^{2t}] &\geq \tilde{\mathbb{E}}_\mu \left[ y_1^{4(t-s^*)} (1 - y_1^2)^{2s^*} \right] \\ &\geq e^{-O(\delta + \beta \log n)t} \geq e^{-O(t/\log n)}. \end{aligned}$$

On the other hand, we claim that  $s^*$  cannot be larger than  $\beta t$ . If  $s^* > \beta t$ , then

$$\begin{aligned} \binom{t}{s^*} O\left(\frac{\rho}{\delta^3} + \frac{\alpha t}{s^*}\right)^{s^*} &\leq O\left(\frac{1}{\beta} \left(\frac{\rho}{\delta^3} + \frac{\alpha}{\beta}\right)\right)^{s^*} \\ &\leq e^{-\beta t} < e^{-O(\delta)t}, \end{aligned}$$

since our parameters satisfy  $\rho, \alpha \leq \frac{1}{\log^C n}$  for some large enough constant  $C$  so that  $\frac{1}{\beta} \left(\frac{\rho}{\delta^3} + \frac{\alpha}{\beta}\right) \ll 1$ , and the last inequality follows from  $\delta = \frac{1}{\log^3 n} \ll \beta$ . This contradicts Eq. (10). Thus, it must be that  $s^* \leq \beta t$ , and we have  $\tilde{\mathbb{E}}_\mu[y_1^{2t}] \geq e^{-O(t/\log n)}$ , completing the proof.  $\square$

With Lemmas IV.6 and IV.12 and Lemma IV.10 (the rounding algorithm of [5]) in hand, we can now prove that Algorithm 2 succeeds in recovering  $v$ , completing the proof of Theorem IV.2.

*Proof of Theorem IV.2.* Let  $r_1 \in \mathbb{R}^d$  be the first row of the unknown rotation matrix  $R$  from Model 1, and note that  $r_1$  is a unit vector. By Lemma IV.12, the pseudo-distribution  $\mu$  obtained from step (1) of Algorithm 2 satisfies that  $\tilde{\mathbb{E}}_\mu[\langle r_1, x \rangle^{2t}] \geq e^{-O(t/\log n)}$ .

By repeating the algorithm in Lemma IV.10  $2^{\tilde{O}(t)}$  times, with high probability at least one of the unit vectors  $\hat{x}$  satisfies  $\langle \hat{x}, r_1 \rangle \geq 1 - O(\delta)$  where  $\delta := \frac{1}{\log n}$ , which means that  $\|R(\hat{x} - r_1)\|_2 = \|R\hat{x} - e_1\|_2 \leq O(\delta)$ . Since  $v$  is a  $(\rho, \gamma)$ -compressible unit vector (for  $\gamma$  bounded away from 1), by (1) of Lemma IV.6, it follows that  $\hat{v} = \tilde{A}\hat{x}$  satisfies  $\max_{S: |S| \leq \rho n} \|\hat{v}_S\|_2 \geq 1 - \gamma - O(\delta)$ .

Thus, in step (3) of Algorithm 2, we will choose an  $\hat{x}^*$  from the list such that  $\hat{v}^* = \tilde{A}\hat{x}^*$  satisfies  $\max_{S: |S| \leq \rho n} \|\hat{v}_S^*\|_2 \geq 1 - \gamma - O(\delta)$ . By (2) of Lemma IV.6, we have that either  $\|\hat{x}^* - r_1\|_2$  or  $\|\hat{x}^* + r_1\|_2 \leq o(1)$ . Since the singular values of  $\tilde{A}$  are all  $1 \pm o(1)$  by Lemma IV.5, it follows that  $\hat{v}^*$  is  $o(1)$ -close to  $\pm v$  and  $\|\hat{v}^*\|_2 = 1 \pm o(1)$ . This completes the proof.  $\square$

## REFERENCES

- [1] Radosław Adamczak, Alexander E Litvak, Alain Pajor, and Nicole Tomczak-Jaegermann. Sharp bounds on the rate of convergence of the empirical covariance matrix. *Comptes Rendus. Mathématique*, 349(3-4):195–200, 2011.
- [2] Arash A Amini and Martin J Wainwright. High-dimensional analysis of semidefinite relaxations for sparse principal components. In *2008 IEEE international symposium on information theory*, pages 2454–2458. IEEE, 2008.
- [3] Boaz Barak, Fernando GSL Brandao, Aram W Harrow, Jonathan Kelner, David Steurer, and Yuan Zhou. Hypercontractivity, Sum-of-Squares Proofs, and their Applications. In *Proceedings of the forty-fourth annual ACM symposium on Theory of computing*, pages 307–326, 2012.
- [4] Boaz Barak, Jonathan A Kelner, and David Steurer. Rounding Sum-of-Squares Relaxations. In *Proceedings of the forty-sixth annual ACM symposium on Theory of computing*, pages 31–40, 2014.
- [5] Boaz Barak, Jonathan A Kelner, and David Steurer. Dictionary Learning and Tensor Decomposition via the Sum-of-Squares Method. In *Proceedings of the forty-seventh annual ACM symposium on Theory of computing*, pages 143–151, 2015.
- [6] Boaz Barak and David Steurer. Proofs, beliefs, and algorithms through the lens of sum-of-squares. *Course notes*: <http://www.sumofsquares.org/public/index.html>, 2016.
- [7] Vijay Bhattiprolu, Mrinalkanti Ghosh, Venkatesan Guruswami, Euiwoong Lee, and Madhur Tulsiani. Weak Decoupling, Polynomial Folds, and Approximate Optimization over the Sphere. In *2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 1008–1019. IEEE, 2017.
- [8] Vijay Bhattiprolu, Venkatesan Guruswami, and Euiwoong Lee. Sum-of-squares certificates for maxima of random tensors on the sphere. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM*, volume 81 of *LIPIcs*, pages 31:1–31:20. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2017.
- [9] Emmanuel J Candes and Terence Tao. Decoding by linear programming. *IEEE transactions on information theory*, 51(12):4203–4215, 2005.
- [10] Hongjie Chen and Tommaso d’Orsi. On the well-spread property and its relation to linear regression. In *Conference on Learning Theory*, pages 3905–3935. PMLR, 2022.
- [11] Laurent Demanet and Paul Hand. Recovering the Sparsest Element in a Subspace. *arXiv preprint arXiv:1310.1654*, 2013.
- [12] Ilias Diakonikolas and Daniel Kane. Non-Gaussian Component Analysis via Lattice Basis Reduction. In *Conference on Learning Theory*, pages 4535–4547. PMLR, 2022.
- [13] Yunzi Ding, Dmitriy Kunisky, Alexander S Wein, and Afonso S Bandeira. The Average-Case Time Complexity of Certifying the Restricted Isometry Property. *IEEE Transactions on Information Theory*, 67(11):7355–7361, 2021.
- [14] Yunzi Ding, Dmitriy Kunisky, Alexander S Wein, and Afonso S Bandeira. Subexponential-time algorithms for sparse PCA. *Foundations of Computational Mathematics*, pages 1–50, 2023.
- [15] David L Donoho. Compressed Sensing. *IEEE Transactions on information theory*, 52(4):1289–1306, 2006.
- [16] Tommaso d’Orsi, Pravesh K Kothari, Gleb Novikov, and David Steurer. Sparse PCA: Algorithms, Adversarial Perturbations and Certificates. In *2020 IEEE 61st Annual Symposium on Foundations of Computer Science (FOCS)*, pages 553–564. IEEE, 2020.
- [17] Tommaso d’Orsi, Chih-Hung Liu, Rajai Nasser, Gleb Novikov, David Steurer, and Stefan Tiegel. Consistent Estimation for PCA and Sparse Regression with Oblivious Outliers. *Advances in Neural Information Processing Systems*, 34:25427–25438, 2021.
- [18] T Figiel, J Lindenstrauss, and VD Milman. The dimension of almost spherical sections of convex bodies. *Acta Mathematica*, 139:53–94, 1977.
- [19] Noah Fleming, Pravesh Kothari, and Toniann Pitassi. Semialgebraic Proofs and Efficient Algorithm Design. *Foundations and Trends® in Theoretical Computer Science*, 14(1-2):1–221, 2019.
- [20] Andrei Yurevich Garnaev and Efim Davydovich Gluskin. The widths of a Euclidean ball. In *Doklady Akademii Nauk*, volume 277, pages 1048–1052. Russian Academy of Sciences, 1984.
- [21] Venkatesan Guruswami, Pravesh K Kothari, and Peter Manohar. Algorithms and certificates for Boolean CSP refutation: smoothed is no harder than random. In *Proceedings of the 54th Annual ACM SIGACT Symposium on Theory of Computing*, pages 678–689, 2022.
- [22] Venkatesan Guruswami, James R Lee, and Alexander Razborov. Almost Euclidean subspaces of  $\ell_1^N$  via expander codes. *Combinatorica*, 30(1):47–68, 2010.
- [23] Venkatesan Guruswami, James R Lee, and Avi Wigderson. Euclidean sections of  $\ell_1^N$  with sublinear randomness and error-correction over the reals. In *International Workshop on Approximation Algorithms for Combinatorial Optimization*, pages 444–454. Springer, 2008.
- [24] Samuel Hopkins. Statistical inference and the sum of squares method. *PhD thesis, Cornell University*, 2018.
- [25] Samuel B Hopkins, Tselil Schramm, Jonathan Shi, and David Steurer. Fast spectral algorithms from sum-of-squares proofs: tensor decomposition and planted sparse vectors. In *Proceedings of the forty-eighth annual ACM symposium on Theory of Computing*, pages 178–191, 2016.
- [26] Jun-Ting Hsieh, Pravesh K Kothari, and Sidhanth Mohanty. A simple and sharper proof of the hypergraph Moore bound. In *Proceedings of the 2023 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 2324–2344. SIAM, 2023.
- [27] Jun-Ting Hsieh, Pravesh K Kothari, Lucas Pesenti, and Luca Trevisan. New SDP Roundings and Certifiable Approximation for Cubic Optimization. In *Proceedings of the 2024 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 2337–2362. SIAM, 2024.
- [28] Jun-Ting Hsieh, Pravesh K Kothari, Aaron Potechin, and Jeff Xu. Ellipsoid Fitting up to a Constant. In *50th International Colloquium on Automata, Languages, and Programming (ICALP 2023)*. Schloss-Dagstuhl-Leibniz Zentrum für Informatik, 2023.
- [29] Piotr Indyk. Stable Distributions, Pseudorandom Generators, Embeddings, and Data Stream Computation. *Journal of the ACM (JACM)*, 53(3):307–323, 2006.
- [30] Piotr Indyk. Uncertainty principles, extractors, and explicit embeddings of  $\ell_2$  into  $\ell_1$ . In *Proceedings of the thirty-ninth annual ACM symposium on Theory of computing*, pages 615–620, 2007.
- [31] Piotr Indyk and Stanislaw Szarek. Almost-Euclidean Subspaces of via Tensor Products: A Simple Approach to Randomness Reduction. In *International Workshop on Randomization and Approximation Techniques in Computer Science*, pages 632–641. Springer, 2010.
- [32] Iain M Johnstone and Arthur Yu Lu. On consistency and sparsity for principal components analysis in high dimensions. *Journal of the American Statistical Association*, 104(486):682–693, 2009.
- [33] 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.
- [34] Boris S Kashin and Vladimir N Temlyakov. A Remark on Compressed Sensing. *Mathematical notes*, 82:748–755, 2007.
- [35] Boris Sergeevich Kashin. Diameters of some finite-dimensional sets and classes of smooth functions. *Izvestiya Rossiiskoi Akademii Nauk. Seriya Matematicheskaya*, 41(2):334–351, 1977.
- [36] Pascal Koiran and Anastasios Zouzias. Hidden Cliques and the Certification of the Restricted Isometry Property. *IEEE transactions on information theory*, 60(8):4999–5006, 2014.
- [37] Dmitriy Kunisky, Alexander S Wein, and Afonso S Bandeira. Notes on computational hardness of hypothesis testing: Predictions using the low-degree likelihood ratio. *arXiv preprint arXiv:1907.11636*, 2019.
- [38] Shachar Lovett and Sasha Sodin. Almost Euclidean sections of the  $N$ -dimensional cross-polytope using  $O(N)$  random bits. *Communications in Contemporary Mathematics*, 10(04):477–489, 2008.
- [39] Cheng Mao and Alexander S Wein. Optimal Spectral Recovery of a Planted Vector in a Subspace. *arXiv preprint arXiv:2105.15081*, 2021.
- [40] Ryan O’Donnell and Yuan Zhou. Approximability and proof complexity. In *Proceedings of the twenty-fourth annual ACM-SIAM symposium on Discrete algorithms*, pages 1537–1556. SIAM, 2013.
- [41] Qing Qu, Ju Sun, and John Wright. Finding a sparse vector in a subspace: Linear sparsity using alternating directions. *Advances in Neural Information Processing Systems*, 27, 2014.

- [42] Prasad Raghavendra, Satish Rao, and Tselil Schramm. Strongly refuting random CSPs below the spectral threshold. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2017, Montreal, QC, Canada, June 19-23, 2017*, pages 121–131. ACM, 2017.
- [43] Daniel A Spielman, Huan Wang, and John Wright. Exact recovery of sparsely-used dictionaries. In *Conference on Learning Theory, JMLR Workshop and Conference Proceedings*, 2012.
- [44] Terence Tao. Topics in random matrix theory. *Graduate Studies in Mathematics*, 2012.
- [45] Roman Vershynin. High-Dimensional Probability. *University of California, Irvine*, 2020.
- [46] Alexander S Wein, Ahmed El Alaoui, and Cristopher Moore. The Kikuchi hierarchy and tensor PCA. In *2019 IEEE 60th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 1446–1468. IEEE, 2019.
- [47] Ilias Zadik, Min Jae Song, Alexander S Wein, and Joan Bruna. Lattice-Based Methods Surpass Sum-of-Squares in Clustering. In *Conference on Learning Theory*, pages 1247–1248. PMLR, 2022.