A Randomized Algorithm for Sparse PCA based on the Basic SDP Relaxation

Alberto Del Pia [*] Dekun Zhou [†]

July 12, 2025

Abstract

Sparse Principal Component Analysis (SPCA) is a fundamental technique for dimensionality reduction, and is NP-hard. In this paper, we introduce a randomized approximation algorithm for SPCA, which is based on the basic SDP relaxation. Our algorithm has an approximation ratio of at most the sparsity constant with high probability, if called enough times. Under a technical assumption, which is consistently satisfied in our numerical tests, the average approximation ratio is also bounded by $\mathcal{O}(\log d)$, where d is the number of features. We show that this technical assumption is satisfied if the SDP solution is low-rank, or has exponentially decaying eigenvalues. We then present a broad class of instances for which this technical assumption holds. We also demonstrate that in a covariance model, which generalizes the spiked Wishart model, our proposed algorithm achieves a near-optimal approximation ratio. We demonstrate the efficacy of our algorithm through numerical tests on real-world datasets.

Key words: Sparse PCA, Randomized algorithm, Semidefinite Programming

1 Introduction

The Principal Component Analysis (PCA) problem involves finding a linear combination of d features that captures the maximum possible variance in a given $d \times d$ data matrix A. Formally, the problem is defined as

$$\max \ x^\top A x \quad \text{s.t.} \ \ \|x\|_2 = 1. \tag{PCA}$$

PCA is a widely used statistical technique for reducing the dimensionality of large datasets, and it has been successfully applied to a broad range of topics, including neuroscience, meteorology, psychology, genetics, finance, and pattern recognition. For a comprehensive overview of the applications of PCA, we refer interested readers to [26].

Despite its usefulness, the interpretation of a solution to PCA is limited since the principal component (PC) is often a linear combination of all d features. To address this issue, the Sparse Principal Component Analysis problem (SPCA) is introduced. SPCA aims to find a *sparse* linear combination of features while capturing the maximum variance. SPCA is formally defined as:

$$\max \ x^\top A x \quad \text{s.t.} \ \ \|x\|_2 = 1, \ \ \|x\|_0 \le k. \tag{SPCA}$$

[*] Department of Industrial and Systems Engineering & Wisconsin Institute for Discovery, University of Wisconsin-Madison. E-mail: `delpia@wisc.edu`

[†] Department of Industrial and Systems Engineering & Wisconsin Institute for Discovery, University of Wisconsin-Madison. E-mail: `dzhou44@wisc.edu`

Here k is the *sparsity constant,* a positive integer that sets an upper bound on the number of non-zero entries in the d-dimensional vector x. Compared with PCA, using SPCA for dimensionality reduction yields more interpretable components, lowers downstream memory and computational costs, and helps prevent overfitting. SPCA has a wide range of real-world applications, such as identifying influential single-nucleotide polymorphisms in genetics [31], selecting informative object features in computer vision [35], and organizing a large corpus of text data in data science [46].

SPCA is an NP-hard problem in general [33], making it computationally challenging in practice. In fact, it is NP-hard to approximate SPCA within a multiplicative ratio of $(1 + \epsilon)$ for some constant $\epsilon > 0$ [11]. Despite the hardness of the problem, many polynomial-time approximation algorithms have been proposed to tackle SPCA. For instance, [37] accelerated SPCA by replacing the input matrix with its rank-m approximation formed from the top m eigenpairs, resulting an algorithm with a running time $\mathcal{O}(d^{m+1} \log d)$ and an approximation ratio $1/(1 - \delta_m)$, with $\delta_m \leq \lambda_{m+1} \cdot \min\{d/(k\lambda_1), 1/\max_{i\in[d]} A_{ii}\}$. This algorithm has the advantage that if the eigenvalues $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_d$ of A satisfy an exponential decay, i.e., $\lambda_{i+1}/\lambda_i \leq c \in (0, 1)$, then δ_m decreases exponentially as m increases. [32] introduced two polynomial-time k-approximation algorithms: the Greedy algorithm and the Local Search algorithm. [12] introduced an ϵ-additive approximation algorithm with a runtime of $\mathcal{O}(d^{\mathrm{poly}(1/\epsilon)})$ based on the previous work [3]. Moreover, [12] also proposed a simple $\min\{\sqrt{k}, d^{1/3}\}$-approximation algorithm for SPCA, which is, to the best of our knowledge, the best known approximation ratio for SPCA algorithms with polynomial runtime.

Our contributions. In this paper, we propose a randomized algorithm based on the *basic* semidefinite programming (SDP) relaxation to SPCA,

$$\max \ \mathrm{tr}(AW) \quad \text{s.t.} \ \ \mathrm{tr}(W) = 1, \ \|W\|_1 \leq k, \ W \succeq 0. \tag{SPCA-SDP}$$

Our algorithm is not only efficient, but also improves upon the best known polynomial-time benchmark of $\min\{\sqrt{k}, d^{1/3}\}$ across a variety of practically motivated instances.

Efficient SDP-based Randomized Rounding with Provable Guarantees. Our algorithm transforms an (approximate) optimal solution W^* of SPCA-SDP into a feasible solution for SPCA, in time $\mathcal{O}(d \log d)$. Despite its efficiency, our algorithm achieves an approximation ratio that is upper bounded by k with high probability when run for $\Omega(d/k)$ many independent times, matching the guarantees of the Greedy algorithm and the Local Search algorithm in [32]. The average approximation ratio can be further upper bounded by $\mathcal{O}(\log d)$ under mild assumption on k and a technical assumption related to sum of square roots of diagonal entries of W^*. We note that in the regime where $k = \Omega(\log^2 d)$, and where the technical assumption is true, our algorithm admits an average approximation error strictly better than the best-known polynomial-time guarantee $\min\{\sqrt{k}, d^{1/3}\}$. We further show that this technical assumption is true if the rank of W^* is fixed, or W^* admits exponentially decaying eigenvalues. We further identify two classes of instances that guarantee a rank-one optimal solution, and hence satisfy the technical assumption: (i) Rank-one input matrices whose non-zero entries are bounded below by a certain value; (ii) General input matrices with a sufficient eigenvalue gap, a uniform contraction condition, and small entries outside a specific index set. To the best of our knowledge, we provide the first deterministic classes of instances in which SPCA-SDP admits rank-one optimal solutions. We also perform extensive numerical experiments, demonstrating the effectiveness of our proposed algorithm in real-world datasets. Oftentimes, our algorithm can find a solution which is as good as the best solution other existing methods studied in [12, 37, 32, 4, 21] can find. In terms of the runtime for SPCA-SDP, although general-purpose SDP solvers scale poorly, we find that the GPU implementation of an approximation algorithm [45] proves remarkably efficient: it finds W^* on $d = 2000$ instances of SPCA-SDP in under six seconds.

Adversarial-Robust Recovery Guarantees. We show that SPCA-SDP is robust to adversarial perturbations in a general covariance model. Specifically, we consider a scenario where the input matrix A is under adversarial attack, i.e., $A = (B + M)^\top (B + M)$, where B represents the data matrix that has i.i.d. rows sampled from a covariance model having a sparse spike, and M represents an adversarial perturbation. We show that SPCA-SDP still yields an optimal solution close to the sparse spike when the sample size is sufficiently large. This generalizes the findings of [23] and provides further insight into the strong computational performance of SPCA-SDP. Under this model, our algorithm achieves an approximation ratio near one, indicating that even if the technical assumption for our randomized algorithm is not met, SPCA-SDP and our algorithm can still perform exceptionally well across diverse inputs.

Organization of this paper. In Section 2, we provide an overview of related work. In Section 3, we introduce our randomized algorithm for solving SPCA and provide theoretical results on approximation guarantees. In Section 4, we define a general statistical model that considers adversarial perturbation, demonstrate the robustness of SPCA-SDP against such perturbations, and provide an approximation bound for our algorithm within this model. In Section 5, we present numerical experiments conducted on various real-world datasets to evaluate the performance of our algorithm and compare with other algorithms. We defer proofs in Sections 3 and 4 to Sections 6 and 7. For the remainder of the section, we introduce the notation used in the paper.

Notation. Sets, vectors, and matrices: For any positive integer d, we define $[d] := \{1, 2, \dots, d\}$. Let x be a d-vector. The *support* of x is the set $\operatorname{Supp}(x) := \{i \in [d] : x_i \neq 0\}$. Given an index set $\mathcal{I} \subseteq [d]$, denote by $x_\mathcal{I}$ the sub-vector of x indexed by $\mathcal{I}$, and we write $x_i := x_{\{i\}}$. For $1 \le p \le \infty$, we denote the *p-norm* of x by $\|x\|_p$. The 0-*(pseudo)norm* of x is $\|x\|_0 := |\operatorname{Supp}(x)|$. We say that x is *k-sparse* if $\|x\|_0 \le k$. Let M be a $m \times n$ matrix. Given two index sets $\mathcal{I} \subseteq [m]$, $\mathcal{J} \subseteq [n]$, we denote by $M_{\mathcal{I},\mathcal{J}}$ the submatrix of M consisting of the entries in rows $\mathcal{I}$ and columns $\mathcal{J}$. Let X be an $m \times m$ symmetric positive semidefinite matrix, i.e., $X \succeq 0$, denote by $Y := \sqrt{X}$ the *matrix square root of* X, i.e., $Y = Y^\top \succeq 0$ and $X = YY$. For $1 \le p, q \le \infty$, the *p-to-q norm* of M is defined as $\|M\|_{p\to q} := \max_{\|x\|_p = 1} \|Mx\|_q$. The *2-norm* of M is defined by $\|M\|_2 = \|M\|_{2\to 2}$. The *1-norm* of M is defined by $\|M\|_1 = \sum_{i,j} |M_{ij}|$. The *infinity norm* of M is defined by $\|M\|_\infty := \max_{i,j} |M_{ij}|$. The *Frobenius norm* of M is defined as $\|M\|_F := \sqrt{\sum_{i,j} |M_{ij}|^2}$.

Approximation ratio and ϵ-approximate solution: Let w^* be an optimal solution to a maximization problem $\mathcal{P}$ with objective function f and input D. We say a (randomized) algorithm $\mathcal{A}$ is an approximation algorithm to $\mathcal{P}$ with an *approximation ratio* r, if $\mathcal{A}$ can output a random solution $\bar{w}$ with input D such that $\mathbb{E} f(\bar{w}) \ge 1/r \cdot f(w^*)$. Sometimes we will also say that $\mathcal{A}$ is an *r-approximation algorithm* for brevity. We say a solution $\tilde{w}$ is an *ϵ-approximate solution* to $\mathcal{P}$ if $\tilde{w}$ is feasible to $\mathcal{P}$ such that $f(\tilde{w}) \ge f(w^*) - \epsilon$.

2 Related work

In this section, we discuss the literature related to our work. First, we discuss results related to the basic SDP relaxation, SPCA-SDP. It is known that SDP can be solved in polynomial time up to an arbitrary accuracy, by means of the ellipsoid algorithm and interior point methods [39, 30]. The basic SDP relaxation SPCA-SDP was initially proposed in [15] and has been extensively researched since then. The literature includes studies on its performance under various statistical models and its approximability. The statistical performance of SPCA-SDP has been thoroughly investigated, with the assumption that $A = B^\top B$ and B being an $n \times d$ matrix. For example, [1] demonstrated that the *sparse spike*, which is the sparse maximal eigenvector, can be recovered in a particular covariance model, known as the Wishart spiked model, when the number of samples n is above the threshold $\Omega(k \log d)$. Then, [28] showed that SPCA-SDP is unable to recover the sparse spike if $k = \Omega(\sqrt{n})$ in the model discussed in [1]. In a more general spiked covariance model, [44] showed that SPCA-SDP can recover the sparse spike but at a slightly higher sample complexity $\Omega(k^2 \log d)$. Additionally, [23] demonstrated that SPCA-SDP is robust to adversarial perturbations in the Wishart spiked model. Moreover, a streamline of work [5, 6] investigated the information theoretical limits of SPCA-SDP recovering the sparse spike in certain covariance models. Regarding approximation results, an approximation algorithm based on SPCA-SDP was developed in [13]. It is worth noting that authors of [13] acknowledged that their theoretical guarantees may not be indicative of the outstanding practical performance of their algorithm. Nevertheless, they provided compelling empirical evidence of its efficacy by showcasing impressive computational results on diverse real-world datasets. A worst-case approximation bound of SPCA-SDP was studied in [12], revealing that there exists an instance that results in an approximation ratio that is quasi-quasi-polynomial in d.

Then, we discuss efficient algorithms for SPCA in the literature. To the best of our knowledge, there are currently five other main categories of methods for finding (approximate) solutions to SPCA, except via its basic SDP relaxation. Firstly, various existing methods solve SPCA by relaxing the sparsity constraint with a convex constraint. These methods include providing practical algorithms to maximize the objective value in an ℓ_1 ball [19], or by solving stronger SDP relaxations [27]. Secondly, methods based on integer programming have been developed to solve SPCA exactly or approximately, including using integer programs to obtain dual bounds, for either single sparse PC [20] or multiple but row-sparse PCs [21], solving mixed-integer SDPs or mixed-integer linear programs [32], using branch-and-bound algorithm to obtain certifiable (near) optimality [4, 7], and combining integer programs with geometric approach to obtain multiple sparse PCs [8]. Thirdly, polynomial-time algorithms for a fixed rank input matrix A are proposed, either for a single sparse PC [37], or for multiple but row-sparse PCs [16]. It should be noted that the complexity of these algorithms are oftentimes exponential in $\text{rank}(A)$. Fourthly, polynomial running-time approximation algorithms for SPCA have been developed, including finding a low rank approximation of A and then solving SPCA exactly [37], deriving an approximation algorithm via basis truncation [12], and via basic SDP relaxation [13]. The fifth category includes methods for solving SPCA in certain statistical models via different approaches, including covariance thresholding method [18] and sub-exponential algorithms [22]. A recent work [17] introduces a plug-and-play framework to provide speedup to these existing algorithms in the categories by first finding block-diagonal approximation to the input matrix A and then solving SPCA sub-problems inside each block. For a more comprehensive review of the literature in each of the categories mentioned above, interested readers are referred to the cited papers.

Note that some of the aforementioned work have established approximation results using non-convex optimization methods, which may necessitate an exponential runtime. The non-convexity comes from the fact that the authors solve a maximization problem of a convex objective. For instance, the authors of [20] have demonstrated that by relaxing SPCA within an ℓ_1 ball, an upper bound for SPCA with constant approximation ratio can be achieved. In [21], the authors have extended these findings to finding m sparse principal components with a global support, with an approximation ratio $\mathcal{O}(\sqrt{\log m})$.

3 A randomized algorithm for SPCA

In this section, we present our randomized approximation algorithm for SPCA. For brevity, we denote an ϵ-approximate optimal solution to SPCA-SDP as W^* and an optimal solution to SPCA as x^*, for the rest of this paper.

We begin by presenting the motivation. Although SPCA-SDP has nice properties, such as the result in [32], where the authors demonstrate that the objective value of an optimal solution to SPCA-SDP is at most k times that of SPCA, there are several limitations that hinder the practical application of SPCA-SDP. For instance, as pointed out in [18], that there are also some theoretical limitations of SPCA-SDP. Specifically, (i) W^* is not guaranteed to be a rank-one matrix in general; and (ii) in some cases where W^* is rank-one, denoted as $W^* = v^*(v^*)^\top$, but oftentimes the zero-(pseudo)norm of v, $\|v\|_0$, is larger than k. This raises a natural question: is there a way to transform W^* into a feasible solution for SPCA with high quality?

In [13], the authors partially address this question by providing a vector z, obtained by finding the best rank-one approximation $uu^\top$ to W^*, keeping the $\mathcal{O}(k^2/\epsilon^2)$ largest components (in absolute value) in u, and setting the other entries to zero. They obtain that $z^\top Az \geq 1/\alpha \cdot (x^*)^\top Ax^* - \epsilon$, where $\alpha \geq 1$ is the ratio $\operatorname{tr}(AW^*)/u^\top Au$. However, this algorithm has two major issues: (a) z is generally not a k-sparse vector and $\|z\|_0$ could be much larger than k; and (b) there is no clear theoretical bound on α, making it difficult for users to predict the worst-case quality of z.

In this section, we present Algorithm 1 as an approximation algorithm for SPCA-SDP with the aim of obtaining a high-quality k-sparse vector from W^* and addressing the issues discussed above. The main idea behind Algorithm 1 is to treat the diagonal entries in W^* and A as probability masses that determine whether or not to include the corresponding entry in the support of a vector x.

Algorithm 1 Randomized Algorithm for SPCA

```
Require: A matrix $A \in \mathbb{R}^{d \times d}$, a positive semidefinite matrix $W \in \mathbb{R}^{d \times d}$, and a positive integer $k$
Ensure: A vector $z \in \mathbb{R}^d$, such that $\|z\|_2 = 1$ and $\|z\|_0 \leq k$ with high probability
 1: for $i = 1$ to $d$ do
 2:     $a_i \leftarrow \sqrt{W_{ii}}$
 3: for $i = 1$ to $d$ do
 4:     $p_i \leftarrow \min\{1, 2/3 \cdot ka_i / \sum_{j=1}^{d} a_j + 1/12 \cdot kA_{ii} / \operatorname{tr}(A)\}$
 5:     Sample independently $\epsilon_i \leftarrow 1$ with probability $p_i$, and $\epsilon_i \leftarrow 0$ with probability $1 - p_i$
 6: $S \leftarrow \{i \in [d] : \epsilon_i = 1\}$, $z \leftarrow$ zero vector in $\mathbb{R}^d$
 7: if $A \succeq 0$ and $|S| < k$ then
 8:     $S \leftarrow S \cup T$, with $T \cap S = \emptyset$ and $|T| = k - |S|$
 9: $z_S \leftarrow \arg\max_{\|y\|_2=1} y^\top A_{S,S} y$
10: return $z$
```

In Algorithm 1, lines 7-9 aim to find a better solution. Here, the method of determining set $T \subseteq [d] \backslash S$ on line 8 is omitted, as it does not affect the theoretical bound discussed in the next section. Note that, except for line 8, Algorithm 1 simply requires a runtime at most $\mathcal{O}(d + k^2 \log k)$ (assuming line 9 is solved through randomized Lanczos method [38]). In Section 5, we introduce a very simple heuristic to determine T in time $\mathcal{O}(d \log d)$.

In practice, one can call Algorithm 1 several times, with the intension to obtain a better solution. The operational procedures are presented in Algorithm 2. Approximation guarantee for these algorithms will be provided in the subsequent sections.

Note that lines 1-2 in Algorithm 2 give a greedy heuristic and guarantee that Algorithm 2 finds one feasible solution to SPCA. It is very powerful in some statistical models, as we will see in Section 4.

Algorithm 2 Multi-run Randomized Algorithm for SPCA

```
Require: A matrix $A \in \mathbb{R}^{d\times d}$, a positive semidefinite matrix $W \in \mathbb{R}^{d\times d}$, a positive integer $k \le d$,
    the number of calling $N$ of Algorithm 1
Ensure: A unit vector $z \in \mathbb{R}^d$, such that $\|z\|_0 \le k$ with high probability
$S_0 \leftarrow$ the set of indices in $[d]$ that corresponds to the $k$ largest diagonal entries in $W$
$z_0 \leftarrow \arg\max_{\|y\|_2=1} y^\top A_{S_0,S_0} y$
for $i = 1$ to $N$ do
    Obtain $z_i$ using Algorithm 1 with input $(A, W, k)$
return the best feasible solution to SPCA among $\{z_i\}_{i=0}^N$
```

3.1 Approximation guarantees

In this section, we establish approximation bounds for Algorithms 1 and 2. We first show that, if $N = \Omega(d/k)$, Algorithm 2 is a k-approximation algorithm with probability at least 99%.

Theorem 1. *Let x^* be an optimal solution to SPCA with input pair (A, k), where we assume $A \in \mathbb{R}^{d\times d}$ is positive semidefinite, and k is a positive integer such that $k \le d$. Let W^* be an ϵ-approximate optimal solution to SPCA-SDP with input pair (A, k). Let z be the output of Algorithm 2 with input tuple (A, W^*, k, N).*

Then, one could obtain an approximate solution z to SPCA such that $z^\top Az \ge (x^*)^\top Ax^*/k$ with probability at least $1 - \exp\{-kN/(12d)\} - \exp\{-ckN\}$ for some absolute constant $c > 0$.

Then, we show that the approximation ratio of Algorithm 1 is also upper bounded by a $\mathcal{O}(\log d/k)$ multiple of the *sum of square roots* (SSR) of diagonal entries of W^*, i.e., $\sum_{i=1}^d \sqrt{W^*_{ii}}$. Due to the intricate technical details, we present an informal statement of the result as follows. A formal statement can be found in Section 6.1. Additionally, the formal theorem there is more comprehensive, demonstrating that Algorithm 1 remains effective even when A is indefinite.

Theorem 2 (Informal version of Theorem 13)**.** *Let x^* be an optimal solution to SPCA with input pair (A, k), where we assume $A \in \mathbb{R}^{d\times d}$ is positive semidefinite with $\|A\|_2 = 1$, and k is a positive integer such that $k \le d$. Let W^* be an ϵ-approximate optimal solution to SPCA-SDP with input pair (A, k). Denote z to be the output of Algorithm 1 with input tuple (A, W^*, k). Then, there exists a random event $\mathcal{R} \subseteq \{\|z\|_0 \le k\}$ such that $\mathbb{P}(\mathcal{R}) \ge 1 - \exp\{-ck\} - 2d^{-3}$ for an absolute constant $c > 0$, and such that when $ck \ge 3\log(d/k) + \log\log d$, one has*

$$C\log d \cdot \Big[1 + \frac{9(\sum_{i=1}^d \sqrt{W^*_{ii}})^2}{4k}\Big] \cdot \mathbb{E}[z^\top Az|\mathcal{R}] \ge \Big[1 - \mathcal{O}(\frac{1}{\log d})\Big](x^*)^\top Ax^* - \epsilon,$$

for some absolute constant $C > 0$.

Remark. *In this remark, we discuss the approximation ratio of Algorithm 1 in Theorem 2. On one hand, in Theorem 2, we obtain a worst-case multiplicative ratio $\mathcal{O}(d\log d/k)$ due to the fact that $\mathrm{tr}(W^*) = 1$ and Cauchy-Schwarz inequality. On the other hand, it is worth noting that when*

$$\mathrm{SSR} := \sum_{i=1}^d \sqrt{W^*_{ii}} \le c_0 \cdot \sqrt{k} \tag{1}$$

for some absolute constant $c_0 > 0$, Algorithm 1 can obtain a multiplicative ratio $\mathcal{O}(\log d)$. When $k = \Omega(\log^2 d)$, implying $k \ge 3\log(d/k) + \log\log d$, this $\mathcal{O}(\log d)$ guarantee strictly surpasses the $\min\{\sqrt{k}, d^{1/3}\}$-approximation of [12], which, to our knowledge, is the smallest ratio previously known for any polynomial-time SPCA algorithm. We note that, while (1) might not always hold, *it is easily checkable once SPCA-SDP is solved, which could be done in polynomial time. In Section 3.2, we provide further discussions about assumptions on W^* such that* (1) *holds, and in Section 3.3, we provide classes of instances where* (1) *holds for $c_0 = 1$. Furthermore, as we will see in Section 5,* (1) *oftentimes holds in our numerical tests (in fact, $c_0 \leq 2.21$ for 80% of the instances).*

Remark. *In this remark, we point out that Theorem 2 in fact holds true for any $W^* \succeq 0$ such that $\operatorname{tr}(W^*) = 1$ and $\operatorname{tr}(AW^*) \geq (x^*)^\top Ax^* - \epsilon$. This implies that our algorithm extends to any SDP relaxation stronger than SPCA-SDP. For instance, our rounding scheme applies to the tighter relaxation of [27] and preserves the guarantees of Theorem 2. We nevertheless focus on SPCA-SDP for two main reasons. From a theoretical perspective, stronger relaxations do not necessarily yield better SPCA solutions in the input families studied in Sections 3.3 and 4, the latter being a standard model in the statistics literature [1, 28, 44]. From a computational perspective, SPCA-SDP can be (approximately) solved in seconds with a GPU implementation of the CGAL method [45], whereas most stronger relaxations might not be compatible with CGAL and thus scale poorly.*

Note that Theorem 2 gives only an expected approximation guarantee for Algorithm 1. In practice, running Algorithm 2 with a large enough N and returning the best result gives, with high probability, an approximation ratio matching that expectation. The necessary sample size could be found directly from Hoeffding's inequality (see Theorem 4.12 in [34]).

3.2 Assumptions on W^* yielding small sum of square roots

In this section, we discuss assumptions on W^* that yield small SSR, i.e., $\sum_{i=1}^d \sqrt{W^*_{ii}}$, where W^* is an (approximate) optimal solution to SPCA-SDP. In the first proposition, we will show that if SPCA-SDP admits a sparse optimal solution W^*, then the SSR is upper bounded by the support of diagonal entries of W^*. Then, we generalize this result to general low-rank optimal solutions, and show that the SSR is upper bounded by $\sqrt{rk}$, where $r := \operatorname{rank}(W^*)$. In this case, Algorithm 1 gives an average approximation ratio of order $\mathcal{O}(r \log d)$. We also note that this result could be very helpful in practice–it is possible to obtain a fixed-rank approximate solution to SPCA-SDP via CGAL [45] with a fixed number of iterations and a low-rank initial primal solution, or to obtain a local low-rank solution to SPCA-SDP via the first-order approach proposed by Burer and Monteiro [9, 10, 14], or via accelerated first-order methods [43]. Finally, we show that under certain assumptions, SSR is small even when W^* is of full rank. We give our first proposition stating that a sparse W^* has a small SSR:

Proposition 3. *Let $W \succeq 0$, and assume that $\operatorname{tr}(W) = 1$. Define the diagonal support of W as $\operatorname{DSupp}(W) := \{i \in [d] : W_{ii} \neq 0\}$. Then, $\mathrm{SSR} = \sum_{i=1}^d \sqrt{W_{ii}} \leq \sqrt{|\operatorname{DSupp}(W)|}$.*

Proof. By Cauchy-Schwarz inequality, $\sum_{i=1}^d \sqrt{W_{ii}}$ is at most $\sqrt{|\operatorname{DSupp}(W)|}$. □

Note that a solution W^* with sparse diagonal support is by definition a low-rank matrix. We now provide an upper bound for SSR to general low-rank feasible solutions to SPCA-SDP:

Proposition 4. *Let $W \succeq 0$, and assume that $\operatorname{tr}(W) = 1$, $\|W\|_1 \leq k$, and $\operatorname{rank}(W) = r$. Then, $\mathrm{SSR} = \sum_{i=1}^d \sqrt{W_{ii}} \leq \sqrt{rk}$.*

Proof. Suppose that $W = YY^\top$ with $Y \in \mathbb{R}^{d\times r}$. Write $Y^\top = (y_1, y_2, \ldots, y_d)$, with $y_i \in \mathbb{R}^r$. It is clear that $W_{ij} = y_i^\top y_j$ for any $i, j \in [d]$. For ease of notation, we define $r_i := \|y_i\|_2$ and $u_i := y_i/r_i$. Then, we see that $\mathrm{SSR} = \sum_{i=1}^d \sqrt{W_{ii}} = \sum_{i=1}^d r_i$ and $\|W\|_1 = \sum_{i,j=1}^d r_i r_j |u_i^\top u_j|$.

We use a probabilistic viewpoint to lower bound $\|W\|_1$ by SSR^2/r. Let $I, J \in [d]$ be two i.i.d. random variables with $\mathbb{P}(I = i) = r_i/\mathrm{SSR}$, and it is clear that

$$\|W\|_1 = \mathrm{SSR}^2\mathbb{E}|u_I^\top u_J| \geq \mathrm{SSR}^2\mathbb{E}|u_I^\top u_J|^2 = \mathrm{SSR}^2\mathbb{E}(u_I^\top u_J)(u_J^\top u_I) = \mathrm{SSR}^2 \operatorname{tr}\left((\mathbb{E}u_I u_I^\top)^2\right),$$

where the inequality follows by the fact that $|u_I^\top u_J| \leq 1$. Define the matrix $G := \mathbb{E}u_I u_I^\top \in \mathbb{R}^{r\times r}$, and let $\lambda_i(G)$ be the i-th largest eigenvalue of G, we obtain that

$$\operatorname{tr}(G^2) = \sum_{i=1}^r \lambda_i(G)^2 = \frac{r \cdot \sum_{i=1}^r \lambda_i(G)^2}{r} \geq \frac{\left(\sum_{i=1}^r \lambda_i(G)\right)^2}{r} = \frac{1}{r}$$

via Cauchy-Schwarz inequality. □

Finally, we show that, if the eigenvalues of W^* decays exponentially, then $\mathrm{SSR} = \mathcal{O}(k \log d)$:

Proposition 5. *Suppose that there exists a constant $q \in (0, 1)$ such that the i-th largest eigenvalue of W, i.e., $\lambda_i(W)$, satisfies that $\lambda_i(W) \leq q^{i-1} \cdot \lambda_1(W)$ for all $i \in [d]$. Then, there exists an absolute constant $C > 0$ such that $\sum_{i=1}^d \sqrt{W_{ii}} \leq C\sqrt{k \log d/ \log(1/q)}$.*

Proof. Let $W \succeq 0$ and assume that $\operatorname{tr}(W) = 1$ and $\|W\|_1 \leq k$. Let the singular value decomposition of W be $\sum_{i=1}^d \lambda_i(W)u_i u_i^\top$. Define the orthogonal matrix $U := (u_1, u_2, \ldots, u_d)$, and thus $U^{-1} = U^\top$ and hence $UU^\top = \sum_{i=1}^d u_i u_i^\top = I_d$. In other words, we have that $\sum_{i=1}^d (u_i)_j^2 = 1$ for any $i \in [d]$. Let r be the smallest integer such that $q^r \leq 1/d^2$, and thus $r = \lceil 2 \log d/ \log(1/q)\rceil$. Write $W_1 := \sum_{i=1}^r \lambda_i(W)u_i u_i^\top$, and $W_2 := W - W_1 = \sum_{i=r+1}^d \lambda_i(W)u_i u_i^\top$. It is clear that

$$\sum_{j=1}^d \sqrt{W_{jj}} = \sum_{j=1}^d \sqrt{(W_1)_{jj} + (W_2)_{jj}} \leq \sum_{j=1}^d \left(\sqrt{(W_1)_{jj}} + \sqrt{(W_2)_{jj}}\right).$$

By Theorem 4, we obtain that $\sum_{j=1}^d \sqrt{(W_1)_{jj}} \leq \sqrt{rk} = \sqrt{\lceil 2k \log d/ \log(1/q)\rceil}$. For the term involving W_2, we see that $\sqrt{(W_2)_{jj}} = \sqrt{\sum_{i=r+1}^d \lambda_i(W)(u_i)_j^2} \leq \sqrt{1/d^2} = 1/d$. Therefore, we obtain that $\sum_{j=1}^d \sqrt{(W_2)_{jj}} \leq d \cdot \frac{1}{d} = 1$, and thus $\sum_{j=1}^d \sqrt{W_{jj}} \leq \sqrt{\lceil 2k \log d/ \log(1/q)\rceil} + 1$. □

3.3 Sufficient conditions for a rank-one optimal solution

In this section, we provide further understanding for the technical condition (1), by providing sufficient conditions for obtaining a rank-one optimal solution to SPCA-SDP. We note that while conditions of this type have been investigated in the context of general QCQPs [42], in this section we focus on conditions specifically tailored to SPCA-SDP. By Theorem 4, the SSR is upper bounded by $\sqrt{k}$ when there is an rank-one (approximate) optimal solution to SPCA-SDP. To our knowledge, we give the first deterministic classes of instances for SPCA-SDP that guarantee rank-one optimal solutions.

We start by stating a simple fact that, if the input matrix A admits a maximum eigenvector with an ℓ_1-norm upper bounded by $\sqrt{k}$, then SPCA-SDP admits a rank-one optimal solution.

Fact 1. *Let that $A \succeq 0$, and denote by $v_1(A)$ an eigenvector corresponding to the maximum eigenvalue of A with $\|v_1(A)\|_2 = 1$. Assume that $\|v_1(A)\|_1 \leq \sqrt{k}$, then $v_1(A)v_1(A)^\top$ is an optimal solution to SPCA-SDP.*

Proof. Let $W \succeq 0$ be a feasible solution to SPCA-SDP, and assume that its singular value decomposition is $W = \sum_{i=1}^d \lambda_i v_i v_i^\top$. Denote by $\lambda_1(A)$ the largest eigenvalue of A. It is clear that

$$\operatorname{tr}(AW) = \sum_{i=1}^d \lambda_i v_i^\top A v_i \leq \sum_{i=1}^d \lambda_i \cdot \lambda_1(A) = \lambda_1(A),$$

where the last equality follows from the fact that $\mathrm{tr}(W) = 1$. We obtain our desired result by noticing the fact that $\left\|v_1(A)v_1(A)^\top\right\|_1 = \|v_1(A)\|_1^2 \le k$ and $\mathrm{tr}(Av_1(A)v_1(A)^\top) = \lambda_1(A)$. □

In the next example, we provide a class of instances that satisfy the assumptions in Fact 1:

Example. *Suppose that the matrix $A = \lambda I - S \succeq 0$ for some $\lambda \ge 0$ and $S \succeq 0$. Denote by $\{n_i\}_{i=1}^r$ an orthonormal basis of the nullspace of S with dimension $r \le d$, and denote by $N := (n_1, n_2, \dots, n_r)$. For $i \in [d]$, let $N_i \in \mathbb{R}^r$ be the row vector of N. Assume that $\sum_{i=1}^d \|N_i\|_2 \le \sqrt{rk}$, then A admits a top eigenvector v such that $\|v\|_1 \le \sqrt{k}$.*

Proof. It only suffices to show that there exists a unit vector $x \in \mathbb{R}^r$ such that $\sum_{i=1}^d \left|\sum_{j=1}^r N_{ij}x_j\right| \le \sqrt{k}$. Indeed, if such x exists, the vector Nx is by definition a top eigenvector of A with $\|Nx\|_2 = 1$ and $\|Nx\|_1 \le \sqrt{k}$.

We use a probabilistic method to prove the desired result. Let $\{\epsilon_i\}_{i=1}^r$ be i.i.d. Rademacher random variables, i.e., $\mathbb{P}(\epsilon_i = \pm 1) = 1/2$. By Khintchine inequality [24], we obtain that $\mathbb{E}\left[\left|\sum_{j=1}^r N_{ij}\epsilon_j\right|\right] \le \|N_i\|_2$, and therefore $\mathbb{E}\sum_{i=1}^d \left|\sum_{j=1}^r N_{ij}\epsilon_j\right| \le \sum_{i=1}^d \|N_i\|_2 \le \sqrt{rk}$. Hence, we see that there exists a vector $y \in \{\pm 1\}^r$ such that $\sum_{i=1}^d \left|\sum_{j=1}^r N_{ij}y_j\right| \le \sqrt{rk}$. Taking $x = y/\sqrt{r}$ concludes the proof. □

However, in practice, the assumptions in Fact 1 might not hold. In the remainder of the section, we provide other classes of instances that allow the top eigenvector of A to have a larger ℓ_1-norm, yet still guarantees that SPCA-SDP admits a rank-one optimal solution. In the next theorem, we show that if A is the sum of a non-negative multiple of the identity and a rank-one matrix, then SPCA-SDP admits a rank-one solution under mild assumptions:

Theorem 6. *Assume that $A = \lambda I_d + uu^\top$ for some vector $u \in \mathbb{R}^d$ with $m := \|u\|_1 / \|u\|_2 > \sqrt{k}$. Let $T := \mathrm{Supp}(u)$, and assume that*

$$\frac{m - \sqrt{k \cdot \frac{|T|-m^2}{|T|-k}}}{|T|} < \min_{i \in T} |u_i| < \frac{m + \sqrt{k \cdot \frac{|T|-m^2}{|T|-k}}}{|T|}. \tag{2}$$

Then, SPCA-SDP admits a unique optimal solution $w^*(w^*)^\top$ with $\mathrm{Supp}(w^*) = T$.

We note that if A is of a fixed rank r (up to an addition of a scaling of identity), an optimal solution to SPCA could be found in polynomial time [37, 16]. In the next theorem, we generalize Theorem 6 to A with arbitrary rank, but satisfying stronger assumptions on its eigensystem:

Assumption 1. *Suppose that there exists a set $S \subseteq [d]$. Let $A_{S,S}$ be the sub-matrix of A indexed by S with eigenvalues $\lambda_1 > \lambda_2 \ge \lambda_3 \ge \cdots \ge \lambda_{|S|} \ge 0$. For each $i \in [|S|]$, let v_i denote the eigenvector corresponding to λ_i, where $\|v_i\|_2 = 1$. Define the sign vector of v_1 to be $s := \mathrm{sign}(v_1)$, and write $\alpha_i := s^\top v_i$ for $i \in [|S|]$. Finally, we set $D := \sqrt{\frac{4\|v_1\|_1^4 - k\|v_1\|_1^2}{k\sum_{i=2}^{|S|}\alpha_i^2}}$. Then, we assume that*

A1 (Sufficient eigengap). The first eigenvector v_1 satisfies $\|v_1\|_1 > \sqrt{k}$. Furthermore, we have $\left(1 - \frac{D}{1-\gamma} \cdot \frac{\sqrt{k \cdot |S|}}{\|v_1\|_1^2}\right)\lambda_1 \ge (D+1)\lambda_2$ for some $\gamma \in [0, 1)$.

A2 (Uniform contraction). We assume that $\mathrm{Supp}(v_1) = S$. For all $j \in [|S|]$, and for all $\lambda \in [\lambda_1/(D+1), \lambda_1]$, we have $\gamma\|v_1\|_1 \cdot |v_{1,j}| \ge \left|\sum_{i=2}^{|S|} \frac{\lambda_1 - \lambda}{\lambda_i - \lambda}\alpha_i v_{i,j}\right|$.

A3 (Small entries outside S). We have that $\max\left\{\max_{i\in S, j\in S^c} |A_{ij}|, \max_{i,j\in S^c} |A_{ij}|\right\} \leq \frac{\delta}{2\sqrt{k}\|v_1\|_1^2}$*, where $\delta > 0$ is a constant of v_i and λ_i such that* $\frac{\left\|(A_{S,S}-\lambda I_{|S|})^{-1}\operatorname{sign}(v_1)\right\|_1}{\left\|(A_{S,S}-\lambda I_{|S|})^{-1}\operatorname{sign}(v_1)\right\|_2} \geq \sqrt{k}$ *for any $\lambda \in [\lambda_1 - \delta, \lambda_1)$.*

Loosely speaking, in A1, we ask that the input matrix A to have a sufficient gap between λ_1 and λ_2. In A2, we ask the eigensystem of A behaves in a good way such that the right-hand-side admits a uniform contraction bound. The intuition behind this assumption is to make sure that the sign vector of our candidate optimal solution w^* would remain unchanged in a certain interval, where we believe that w^* would exist. The details would be made clear in the proofs in Section 6.2. Lastly, in A3, we ask that the entries of the input matrix A is small enough outside S, and we will prove that the constant $\delta > 0$ indeed exists in Theorem 15. We note that these assumptions would not make SPCA easier, as S might not be the support for an optimal solution to SPCA.

Next, we are ready to state the formal result:

Theorem 7. *Under Assumption 1, SPCA-SDP admits a rank-one optimal solution $W^* = w^*(w^*)^\top$, with $w^* \in \mathbb{R}^d$ and $\operatorname{Supp}(w^*) = S$. Moreover, if the inequality in A1 in Assumption 1 is strict, then W^* is the unique optimal solution to SPCA-SDP.*

For a rank-one optimal solution W^*, we can apply Algorithm 2 for a set number of iterations to obtain a solution to SPCA with an approximation ratio of $\mathcal{O}(\log d)$, as guaranteed by Theorem 2. However, we acknowledge that, when W^* is rank-one, stronger approximation guarantees are achievable, as shown in [19, 21]. The authors demonstrate that solving specific non-convex quadratic programs, equivalent to enforcing a rank-one solution in SPCA-SDP, yields a solution with a constant approximation ratio, using a slightly different sampling rule compared to Algorithm 1. Despite some algorithmic overlap, our approach diverges significantly from theirs in several ways: (i) Algorithm 1 takes a square positive semidefinite input, while [19, 21] operates on vector solution(s); (ii) this distinction also leads to different proof techniques: [19, 21] analyze the size of feasible region, whereas we derive good-quality solutions based on properties of (sub-)Gaussian variables, as will be made clear in Section 6; and (iii) while their work focuses on efficiently solving non-convex programs (except exponential time complexity in the worst case), we aim to develop a polynomial-time algorithm with strong approximation ratios and identify input classes where even better performance can be achieved, as discussed in this and the following section.

4 Robustness of basic SDP relaxation within a general covariance model

In this section, we study a statistical model where the input matrix A is of the form $A = (B + M)^\top(B+M)$, where $B \in \mathbb{R}^{n\times d}$ is a data matrix with a certain sparse signal, and M is an adversarial perturbation matrix. Here, n is known to be the number of samples. One of the goals this section is to demonstrate that under this model assumption, W^* closely approximates a sparse signal embedded within the model. Consequently, W^* can be regarded as an effective approximation of the true sparse signal. After that, we show that in this model, our randomized algorithm Algorithm 2 achieves an approximation ratio close to one, suggesting our algorithm is also very effective.

In [23], the authors studied the spiked Wishart model, where M is a zero matrix, and B is a spiked standard Gaussian matrix, i.e., every row of B is an i.i.d. random vector drawn from $\mathcal{N}(0_d, I_d + \beta vv^\top)$, with $\beta > 0$ and v being a k-sparse vector. The authors demonstrated that an optimal solution to SPCA-SDP provides a good approximation to v for a certain sample size n, which is roughly $\mathcal{O}(k \log d + k \|M\|_{1\to 2}^2)$. However, such spiked assumption is often not applicable to real-life scenarios, as it is uncommon for each row of the actual data matrix B to represent a sum of a sparse signal realization $\sqrt{\beta}v$ and independent standard Gaussian noise. This discrepancy serves as the impetus for exploring the performance of SPCA-SDP in more generalized contexts. These contexts are characterized not only by the presence of sub-Gaussian random variables but also by the inclusion of multiple realizations of signals, amongst which a sparse dominant signal is present. Formally, we introduce Model 1:

Model 1. *The input matrix A can be written as $A = (B+M)^\top (B+M)$, where $B \in \mathbb{R}^{n\times d}$ is a random matrix with i.i.d. sub-Gaussian rows with parameter σ^2, and M is a modification matrix such that its maximal column norm is upper bounded by a constant $b > 0$, i.e., $\|M\|_{1\to 2} \le b$. Furthermore, the rows of B have zero means and admit a covariance matrix Σ, such that Σ has a k-sparse maximal eigenvector v associated with eigenvalue λ_1.*

We present in Theorem 9 that our algorithm achieves an approximation ratio close to one given a sufficient number of samples n. This result develops on the following characterization that W^* is close enough to $vv^\top$, implying that SPCA-SDP is robust to adversarial perturbations in Model 1:

Proposition 8. *In Model 1, denote λ_1, λ_2 to be the largest and second largest eigenvalue of Σ, respectively, and assume $\lambda_1 - \lambda_2 > 0$. Let v be the eigenvector associated with λ_1, and denote $a := \min_{i: v_i \neq 0} |v_i|$. Let W^* be an optimal solution to SPCA-SDP. Then, there exists an absolute constant $C^* > 0$ such that when n is greater or equal to*

$$n^* := \max\Big\{C^* \cdot \Big[\frac{k^2\sigma^4 \log d + b^2 k^2 \Big(\sigma^2 + \max \Sigma_{ii}\Big)}{(\lambda_1 - \lambda_2)^2 a^4} + \frac{kb^2}{(\lambda_1-\lambda_2)a^2}\Big], \frac{4}{a^2}, \log d\Big\}, \tag{3}$$

then $\left\|W^* - vv^\top\right\|_\infty \le a^2/2$ holds with probability at least $1 - d^{-10}$.

Remark. *In this remark, we discuss the sample complexity required for SPCA-SDP to recover* Supp(v) *in Model 1, which might be of independent interests. According to Theorem 8, given a fixed signal intensity $\lambda_1 - \lambda_2$, a fixed variance factor σ^2, and a fixed a, if*

$$n = \Omega\left(k^2 \log d + k^2 b^2 \lambda_1 + k b^2 \log d\right), \tag{4}$$

one can find out Supp(v) *with high probability via SPCA-SDP. We note that the term $\Omega(k^2 \log d)$ in* (4) *is due to the recovery of* Supp(v) *in Model 1 without any adversarial perturbation, which is consistent with the findings of [44]. The term $\Omega(k^2 b^2 \lambda_1 + k b^2 \log d)$ in* (4) *reflects the additional number of samples required to recover* Supp(v) *under adversarial perturbations. Finally, we note that such sample complexity is insufficient to imply* (1) *via the use of Theorem 8 directly.*

We are ready to develop the main theorem in this section. The core message is that the greedy heuristic (lines 1 - 2) in Algorithm 2 finds a solution that achieves an approximation ratio near one.

Theorem 9. *In Model 1, denote λ_1, λ_2 to be the largest and second largest eigenvalue of Σ, respectively, and assume $\lambda_1 - \lambda_2 > 0$. Let W^* be an optimal solution to SPCA-SDP. Denote n^* the number defined in* (3). *Then, for $l \ge 1$, suppose that one has $n \ge l \cdot n^*$, then for any number of iterations $N \ge 0$, Algorithm 2 with input (A, W^*, k, N) has an approximation ratio of at most $1 + 2/(8\sqrt{l} - 1)$ with probability at least $1 - d^{-10}$.*

5 Numerical tests

In this section, we present our numerical results. Our main objective is to evaluate the performance of Algorithm 2 on real-world datasets and compare computational performance with existing state-of-the-art algorithms that run in *polynomial time* and have an approximation guarantee. It should be noted that, while comparisons are made, they are limited to a selected subset of existing algorithms, rather than an exhaustive review of all available methods. For further comparisons of SPCA-SDP with other algorithms, readers are directed to the recent study in [13]. We conducted tests on several real-world datasets, with the dimension of matrix A ranging from 79 to 2000. For most available SDP solvers, including Mosek [2] and SCS [36], face weak scalability, making it hard for them to handle semidefinite programs with dimensions exceeding 1000. As a result, we use CGAL [45] to obtain approximation solutions to SPCA-SDP. It is worth noting that, in Algorithm 1, we simply take the set T (on line 8) with $|T| = k - |S|$ to be the index set in $[d]\backslash S$ that yields largest W_{ii}'s (with ties broken arbitrarily).

Introduction to CGAL. The conditional gradient augmented Lagrangian framework (CGAL) [45] is an iterative algorithm that (approximately) solves the following problem:

$$f(x^*) := \min \ f(x) \quad \text{s.t. } x \in \mathcal{X}, \ Cx \in \mathcal{K},$$

where f is a convex and L-smooth function, C is a linear mapping, $\mathcal{X}$ is a convex compact set and $\mathcal{K}$ is a convex set. In the m-th iteration, CGAL is guaranteed to find x_m such that $|f(x_m) - f(x^*)| \leq \mathcal{O}(m^{-1/2})$ and $\operatorname{dist}(Cx_m, \mathcal{K}) \leq \mathcal{O}(m^{-1/2})$. In the tests, number of iterations in CGAL is set to 100 and the parameter λ_0 set to 1, and we initialize x_0 to be the zero matrix—which guarantees that $\operatorname{rank}(x_m) \leq m$. We find that this algorithm could be efficiently implemented on GPU.

Hardware. We conducted all tests on a personal computer with 8 Cores i7-9700K 3.60GHz CPU, 64 GB of memory, and NVIDIA GEFORCE RTX 2080 SUPER with 8 GB of GPU memory.

Baselines. We compare polynomial-time algorithms studied in [37, 12, 32, 4]. It is worth noting that Chan's algorithm studied in [12] finds a $\min\{\sqrt{k}, d^{1/3}\}$-approximate solution to SPCA. To the best of our knowledge, this is the best known approximation ratio that can be obtained in polynomial time for general SPCA. We also compare the performance of Algorithm 2, among the Greedy algorithm and the Local Search algorithm studied in [32], and the Low-Rank method studied in [37]. The Greedy algorithm and the Local Search algorithm both find k-approximate solutions in polynomial time, while the Low-Rank method [37] finds a solution with approximation ratio depending on the decay of the eigenvalues of the input matrix A. For computational efficiency, we apply the Low-Rank method with the rank-2 approximation to A.

We also compare with the Branch-and-Bound (BB) algorithm from [4] and MSPCA algorithm from [21], while the latter finds SPCA both a lower bound (LB) using a heuristic and an upper bound (UB) using MILP. Although BB and MSPCA frequently find good solutions within our time limit, both algorithms have worst-case running times that could grow exponentially with k, so we treat their results as supplementary to the core comparison among polynomial-time methods. BB has no approximation guarantee, and MSPCA, while returning an upper bound within a constant factor of the optimum, still provides no polynomial runtime bound. By contrast, Greedy, Local Search, Low-Rank, and Algorithm 2 combine provable approximation ratios with guaranteed polynomial complexity, making them the efficient choice for large-scale instances (see Table 1).

Summary of results. In Figure 1, we report the following *Chan-normalized Gap* for the five algorithms, and use Chan's Algorithm as a baseline:

$$\text{Chan-normalized Gap} := \frac{\text{Obj}_{\text{alg}} - \text{Obj}_{\text{Chan}}}{\text{Obj}_{\text{Chan}}} \times 100\%,$$

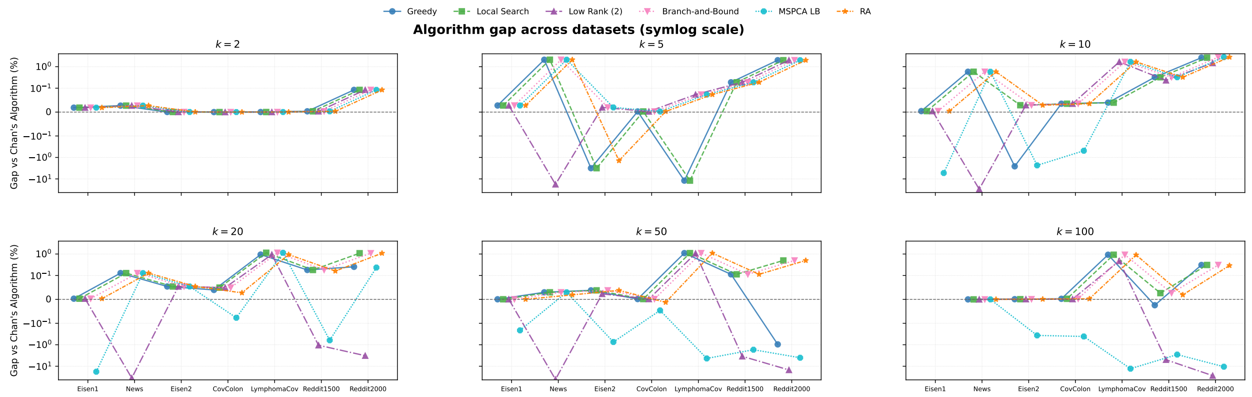

Figure 1: **Chan-normalized gaps (higher = better).** Each panel fixes the sparsity level k and plots the Chan-normalized gaps of the six algorithms across the seven benchmark datasets, ordered from the smallest to the largest dimension (left to right). Our Randomized Algorithm achieves the best objective on 31 of the 41 instances (75%) and never falls below Chan by more than 1.5%. Note that we do not report the result for $k = 100$ on the Eisen1 dataset as its dimension is smaller than 100.

where Obj_{Chan} is the SPCA objective value found by Chan's algorithm, and Obj_{alg} is that of the algorithm under test. Across 41 benchmark instances, and up to a tolerance of 10^{-3} in objective values, Algorithm 2 (RA) attains the best objective in 31 cases and matches or exceeds the competitors as follows:

- Greedy algorithm: 85% of instances (20% strictly better);
- Local Search algorithm: 80% (10% strictly better);
- Low-Rank method: 90% (39% strictly better);
- Chan's algorithm: 95% (78% strictly better);
- BB: 75% (2% strictly better);
- MSPCA LB: 90% (44% strictly better).

Relative to Chan's method, RA shows an average gap of 0.34% (max 2.80%, min -1.42%). The average Chan-normalized gaps for the Greedy algorithm, the Local Search algorithm, the Low-Rank method, BB, and MSPCA LB are -0.16%, -0.03%, -4.16%, 0.38%, and -2.43%, respectively. Note that RA obtains the highest average Chan-normalized gap among all polynomial-time algorithms. In six instances (14% of all instances) RA gains over 1% compared to Chan's Algorithm. We also notice that, although the gaps for Greedy, Local Search, and RA shown in Figure 1 are oftentimes close to each other, RA is able to find better solutions when they fail. For example, for the instance $k = 5$ on the LymphomaCov dataset, both Greedy algorithm and Local Search algorithm find solutions with gaps around -11.97%, yet RA finds a solution with a gap 0.07%. Although BB often attains equal or better objective values compared to all four polynomial-time methods, its Chan-normalized gap exceeds that of RA by only 0.04% on average and at most 1.43%.

In Figure 2, we report the runtimes of different algorithms. It highlights that RA maintains these accuracy gains at practical speeds: the median runtime is merely 2.57 seconds, and maximum runtime is 11.95 seconds. It should be noted that the sampling of $N = 3000$ solutions only takes less than 3.5 seconds for 75% of the instances and the maximum sampling time among the 41 instances is 9.06 seconds. RA is approximately 312 times faster than BB on average, 1512 times faster in the best case ($k = 20$ on Eisen2 dataset). Moreover, RA is approximately 15 times faster than Low-Rank method on average, 67 times faster in the best case ($k = 100$ on LymphomaCov dataset). MSPCA's runtime covers both its lower and upper bound computations and is included only for completeness. On all 20 instances with $k \geq 20$, RA is 2.5 times faster than the Local Search algorithm on average, and 11 times faster in the best case ($k = 100$ on Reddit1500 dataset). Although RA is in general slower than the Greedy algorithm and Chan's algorithm, it offers an accuracy–efficiency trade-off, coupling better objectives with practical scalability for large SPCA instances, as shown in Figure 1.

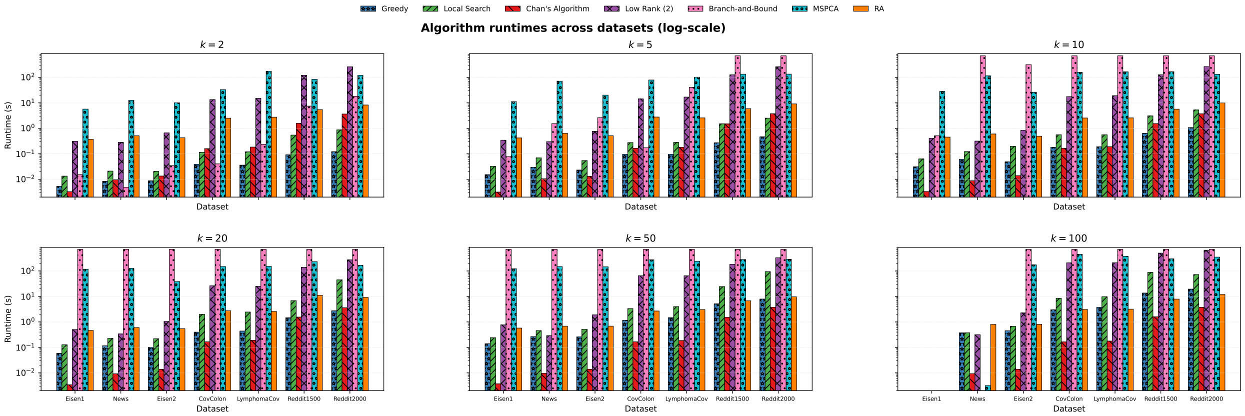

Figure 2: **Wall-clock runtimes (log scale).** The same experiments as Figure 1 but reporting runtime. RA terminates in under 10 seconds on 90% of instances. It is roughly 312 times faster than BB and 15 times faster than the Low-Rank method, while only one order of magnitude slower than the Greedy and Chan's algorithm. When $k \geq 20$, the runtime of RA is on average 2.5 times faster than the Local Search algorithm. We do not report the result for $k = 100$ on the Eisen1 dataset as its dimension is less than 100.

It is important to note that, across all instances the assumption (1) holds with $c_0 \leq 2.16$ on average, the 80-th percentile for c_0 is merely 2.21, while the 90-th percentile is 5.29. As a result, RA invariably achieves an $\mathcal{O}(\log d)$-approximation.

Detailed results. Comprehensive comparisons among the methods are detailed in Table 1. We use "Obj" to denote the objective value of a solution found by a certain algorithm, "Time" to denote the runtime of an algorithm. Note that we run Algorithm 2 with $N = 3000$, and report the largest objective value among those feasible solutions found in the column "Obj" belonging to "Algorithm 2". For reproducibility, we set the random seed to 42. In the table, c_0 stands for $\sum_{i=1}^{d} \sqrt{W_{ii}^*}/\sqrt{k}$, measuring how far SSR is from $\sqrt{k}$. The column "Total Time" stands for the runtime of CGAL + Algorithm 2. We highlight the objective values that are largest among all algorithms.

Table 1: Complete numerical results for all 41 instances. Objective values and runtimes for each algorithm are reported. The time limit for Branch-and-Bound algorithm is set to 700 seconds. In the columns belonging to "CGAL", objective values for SPCA-SDP are reported, and c_0 stands for $\mathrm{SSR}/\sqrt{k}$. In the columns belonging to "Algorithm 2", "Time" stands for the runtime of Algorithm 2 only, while "Total Time" stands for the runtime of CGAL + Algorithm 2.

Dataset	d	k	Greedy		Local Search		Chan's		Low Rank		Branch-and-Bound		MSPCA			CGAL			Algorithm 2		
			Obj	Time	Obj	Time	Obj	Time	Obj	Time	Obj	Time	LB	UB	Time	Time	SDP Obj	c_0	Obj	Time	Total Time
Eisen1	79	2	**7.583**	0.005	**7.583**	0.013	7.582	0.003	**7.583**	0.304	**7.583**	0.015	**7.583**	7.825	5.675	0.077	7.820	1.320	**7.583**	0.296	0.372
Eisen1	79	5	**14.076**	0.015	**14.076**	0.032	14.072	0.003	**14.076**	0.343	**14.076**	0.078	**14.076**	14.272	11.131	0.071	14.623	1.023	**14.076**	0.349	0.420
Eisen1	79	10	**17.335**	0.031	**17.335**	0.063	**17.335**	0.003	**17.335**	0.399	**17.335**	0.508	16.415	17.399	27.938	0.073	17.712	1.006	**17.335**	0.380	0.453
Eisen1	79	20	**17.719**	0.058	**17.719**	0.127	**17.719**	0.003	**17.719**	0.503	**17.719**	700.000	14.530	17.799	117.359	0.046	18.131	0.967	**17.719**	0.417	0.463
Eisen1	79	50	**18.069**	0.140	**18.069**	0.241	**18.069**	0.004	**18.069**	0.760	**18.069**	700.001	18.030	18.082	121.533	0.044	18.131	0.612	**18.069**	0.530	0.574
News	100	2	2750.539	0.008	2750.539	0.021	2749.795	0.010	**2750.540**	0.282	**2750.540**	0.005	**2750.540**	2751.281	12.496	0.091	2745.233	1.521	**2750.540**	0.419	0.510
News	100	5	**3557.755**	0.030	3557.754	0.069	3483.899	0.010	2861.836	0.300	**3557.755**	1.531	**3557.755**	3673.218	70.845	0.094	3679.519	1.126	**3557.755**	0.543	0.637
News	100	10	4549.526	0.061	4549.526	0.123	4523.123	0.009	3171.891	0.316	**4549.527**	700.000	**4549.527**	4727.807	114.517	0.090	4756.311	1.043	**4549.527**	0.511	0.601
News	100	20	5696.468	0.116	**5696.469**	0.229	5689.103	0.009	3674.691	0.336	**5696.469**	700.000	**5696.469**	5935.193	125.960	0.085	6050.354	1.011	**5696.469**	0.517	0.602
News	100	50	7017.849	0.266	7017.850	0.457	7015.768	0.010	4126.741	0.290	**7017.851**	700.000	**7017.851**	7147.107	148.708	0.095	7334.629	1.000	7017.103	0.587	0.682
News	100	100	**7367.672**	0.370	**7367.672**	0.373	**7367.672**	0.009	**7367.672**	0.316	7367.667	0.000	**7367.672**	7367.672	0.003	0.049	7367.672	0.755	**7367.672**	0.749	0.799
Eisen2	118	2	**3.976**	0.009	**3.976**	0.020	**3.976**	0.013	**3.976**	0.655	**3.976**	0.034	**3.976**	3.989	10.164	0.107	4.167	2.103	**3.976**	0.320	0.427
Eisen2	118	5	6.425	0.023	6.425	0.054	6.635	0.013	**6.636**	0.763	**6.636**	2.622	**6.636**	6.824	20.064	0.109	7.117	1.892	6.541	0.404	0.512
Eisen2	118	10	11.412	0.048	**11.718**	0.196	11.715	0.014	**11.718**	0.835	**11.718**	315.274	11.441	11.844	26.214	0.108	11.970	1.432	**11.718**	0.385	0.492
Eisen2	118	20	**19.323**	0.100	**19.323**	0.220	19.312	0.014	**19.323**	1.058	**19.323**	700.000	**19.323**	19.597	38.332	0.106	19.559	1.155	**19.323**	0.439	0.544
Eisen2	118	50	**26.035**	0.257	**26.035**	0.515	26.025	0.014	26.031	1.883	**26.035**	700.000	25.831	26.280	145.161	0.109	27.083	1.003	**26.035**	0.575	0.684
Eisen2	118	100	**27.573**	0.451	**27.573**	0.674	**27.573**	0.014	**27.573**	2.250	**27.573**	700.000	27.471	27.597	172.536	0.060	27.679	0.821	**27.573**	0.749	0.810
CovColon	500	2	**715.395**	0.038	**715.395**	0.113	**715.395**	0.160	**715.395**	13.481	**715.395**	0.040	**715.395**	817.226	33.006	1.955	4628.146	13.003	**715.395**	0.553	2.508
CovColon	500	5	**1646.454**	0.096	**1646.454**	0.277	1646.412	0.163	**1646.454**	14.172	**1646.454**	0.172	**1646.454**	1664.632	78.784	2.132	5110.411	7.487	**1646.454**	0.639	2.771
CovColon	500	10	**2641.229**	0.181	**2641.229**	0.557	2640.302	0.166	**2641.229**	17.591	**2641.229**	700.000	2627.404	2735.330	156.236	1.878	5819.470	5.374	**2641.229**	0.667	2.545
CovColon	500	20	4255.287	0.398	**4255.694**	1.962	4253.598	0.166	**4255.694**	26.237	**4255.694**	700.000	4250.320	4420.541	149.264	2.052	7291.615	3.502	4254.765	0.693	2.745
CovColon	500	50	**7977.493**	1.138	**7977.493**	3.342	7977.381	0.166	**7977.493**	65.019	**7977.493**	700.000	7973.697	8232.498	273.509	1.881	12175.880	1.889	7976.408	0.803	2.683
CovColon	500	100	**12307.385**	3.010	**12307.385**	8.528	12307.080	0.163	12307.228	208.050	**12307.385**	700.003	12256.380	12511.080	446.110	1.957	16957.169	1.437	**12307.385**	1.125	3.081
LymphomaCov	500	2	**2064.868**	0.036	**2064.868**	0.119	2064.864	0.184	**2064.868**	15.086	**2064.868**	0.233	**2064.868**	2652.215	173.575	2.119	3223.160	10.030	**2064.868**	0.625	2.744
LymphomaCov	500	5	3782.621	0.094	3782.621	0.279	4297.340	0.180	**4300.497**	16.836	**4300.497**	40.169	**4300.497**	4375.991	101.530	1.831	5080.984	4.933	**4300.497**	0.734	2.565
LymphomaCov	500	10	5911.412	0.189	5911.412	0.556	5909.077	0.191	**6008.741**	19.093	**6008.741**	700.000	**6008.741**	6857.551	163.512	1.797	6893.303	3.311	6008.317	0.769	2.566
LymphomaCov	500	20	9063.961	0.439	**9082.158**	2.449	8979.248	0.188	9063.961	24.743	**9082.158**	700.001	**9082.158**	9916.811	153.780	1.811	10077.055	2.276	9063.960	0.770	2.580
LymphomaCov	500	50	14546.930	1.471	**14546.931**	3.997	14388.111	0.186	14541.948	64.599	**14546.931**	700.002	13763.608	15338.658	241.648	2.139	15541.962	1.389	**14546.931**	0.916	3.055
LymphomaCov	500	100	**19339.870**	3.696	**19339.870**	9.875	19162.191	0.178	19253.991	210.344	**19339.870**	700.000	16668.065	19260.237	370.315	1.983	20941.087	1.129	19336.097	1.188	3.171
Reddit1500	1500	2	**920.074**	0.091	**920.074**	0.543	920.044	1.552	**920.074**	118.997	**920.074**	7.457	**920.074**	942.504	84.123	2.474	978.858	1.201	**920.074**	2.951	5.424
Reddit1500	1500	5	**980.974**	0.269	**980.974**	1.499	979.129	1.478	**980.974**	125.956	**980.974**	700.010	**980.974**	1018.085	133.564	2.177	1080.998	1.105	**980.974**	3.740	5.917
Reddit1500	1500	10	**1045.743**	0.631	**1045.743**	3.056	1042.357	1.534	1044.777	126.630	**1045.743**	700.002	**1045.743**	1077.079	167.336	2.347	1146.300	1.061	**1045.743**	3.310	5.657
Reddit1500	1500	20	**1105.286**	1.463	**1105.286**	6.859	1103.282	1.552	1091.657	136.850	**1105.286**	700.007	1096.453	1132.024	231.820	5.545	1193.533	1.024	1105.068	5.627	11.172
Reddit1500	1500	50	**1172.655**	5.057	**1172.655**	24.446	1171.323	1.529	1130.934	181.040	**1172.655**	700.012	1151.347	1171.751	280.928	2.546	1230.595	1.007	**1172.655**	4.187	6.734
Reddit1500	1500	100	1200.142	13.617	**1200.751**	87.973	1200.435	1.586	1140.451	501.478	**1200.751**	700.012	1165.997	1187.112	297.662	2.345	1241.312	1.000	1200.663	5.527	7.871
Reddit2000	2000	2	**1254.755**	0.121	**1254.755**	0.863	1253.590	3.643	**1254.755**	257.563	**1254.755**	17.952	**1254.755**	1353.007	118.676	2.568	1352.753	1.526	**1254.755**	5.671	8.239
Reddit2000	2000	5	**1397.358**	0.463	**1397.358**	2.511	1369.780	3.753	**1397.358**	261.937	**1397.358**	700.008	**1397.358**	1543.256	134.227	2.903	1545.370	1.238	**1397.358**	6.195	9.099
Reddit2000	2000	10	1521.308	1.073	1521.308	5.237	1482.320	3.705	1504.450	265.079	**1523.823**	700.012	**1523.823**	1731.351	130.650	2.848	1733.370	1.130	**1523.823**	7.130	9.978
Reddit2000	2000	20	1670.471	2.738	1684.394	44.551	1666.240	3.617	1612.458	271.172	**1684.395**	700.012	1670.153	1961.127	164.186	2.723	2026.850	1.076	1684.394	6.459	9.182
Reddit2000	2000	50	2289.037	7.926	2322.820	94.621	2311.241	3.751	1967.200	331.002	**2322.821**	700.007	2218.043	2315.127	288.524	2.831	2502.577	1.016	2322.820	6.892	9.723
Reddit2000	2000	100	2544.370	19.426	2544.370	72.566	2536.516	3.685	1841.806	654.783	**2544.371**	700.007	2269.553	2338.926	349.681	2.898	2600.108	1.002	2543.924	9.057	11.955
Average			3230.839	1.601	3232.479	9.273	3228.698	0.820	3043.605	96.081	3247.551	453.322	3150.170	3373.851	143.690	1.443	4011.721	2.169	3246.916	2.058	3.501

6 Proofs in Section 3

In this section, we present the proofs that we owe in Section 3.

6.1 Proofs in Section 3.1

In this section, we prove Theorems 1 and 2. To do this, we need to introduce several new notation and random events.

Firstly, we show that Algorithm 1 produces a k-sparse vector with a high probability.

Proposition 10. *Let S be the set in Algorithm 1. Define a random event*

$$\mathcal{A} := \{|S| \leq k\}. \tag{5}$$

Then, $\mathcal{A}$ holds true with probability at least $1 - \exp\{-ck\}$ for some absolute constant $c > 0$.

The proof of this property relies on the well-known multiplicative Chernoff bound:

Lemma 11 ([34], Theorem 4.4 and the remark after Corollary 4.6)**.** *Let $X = \sum_{i=1}^{d} X_i$, where X_i's are independent Bernoulli (i.e., 0-1) random variables. Let L_u be a number such that $L_u \geq \mathbb{E}X$. Then, for any $\delta > 0$, we have*

$$\mathbb{P}\left(X \geq (1+\delta)L_u\right) \leq \left(\frac{e^{\delta}}{(1+\delta)^{1+\delta}}\right)^{L_u}.$$

Proof of Theorem 10. We use Theorem 11 to prove this statement. It is equivalent to showing that $\sum_{i=1}^{d} \epsilon_i \leq k$ with probability at least $1 - \exp\{-ck\}$ for some $c > 0$, where ϵ_i's are the Bernoulli random variables in Algorithm 1. Define $L_u := 3/4 \cdot k$, it is clear that $\mathbb{E}\sum_{i=1}^{d} \epsilon = \sum_{i=1}^{d} p_i \leq L_u$. By Theorem 11, we pick $\delta := 1/3$, and obtain the desired result with $c = -1/4 - \log(3/4) > 0.037$. □

Now, we are ready to prove Theorem 1.

Proof of Theorem 1. Define $i^* := \arg\max_{i \in [d]} A_{ii}$, and it suffices to show that $z^\top A z \geq A_{i^* i^*} = \|A\|_\infty$ with high probability. Indeed, by Hölder's inequality, for any feasible solution W to SPCA-SDP, we have $\operatorname{tr}(AW) \leq \|A\|_\infty \cdot \|W\|_1 \leq k\|A\|_\infty = kA_{i^* i^*}$, where the last equality follows from the fact that for any $i \neq j$, one have $A_{ij}^2 \leq A_{ii}A_{jj} \leq \max\{A_{ii}, A_{jj}\}^2$ due to $A \succeq 0$.

It is clear that the probability of the index i^* not being chosen in the support of a solution z_i obtained at the i-th iteration of Algorithm 2 is exactly $\mathbb{P}\left(\{i^* \text{ is not chosen in } z_i\}\right) = 1 - p_{i^*}$. Therefore, the probability that the support of at least one feasible solution contains i^* is given by

$$\begin{aligned}
&1 - \mathbb{P}\left(\bigcap_{i=1}^{N} (\{i^* \text{ is not chosen in } z_i\} \cup \{z_i \text{ is not feasible}\})\right) \geq 1 - (1 - p_{i^*})^N - (\exp\{-ck\})^N \\
&\geq 1 - \exp\{-Np_{i^*}\} - \exp\{-ckN\} \geq 1 - \exp\left\{-N \cdot \frac{kA_{i^* i^*}}{12\operatorname{tr}(A)}\right\} - \exp\{-ckN\} \\
&\geq 1 - \exp\left\{-N \cdot \frac{k}{12d}\right\} - \exp\{-ckN\},
\end{aligned}$$

where the first inequality follows from Theorem 10 with the same $c > 0$, and at the last equality we use the fact that $dA_{i^* i^*} \geq \operatorname{tr}(A)$. □

To prove Theorem 2, we need an additional useful probabilistic property of the uniform random vector g that we will be used in the proof.

Proposition 12. *Let $g \in \mathbb{R}^d$ be a random vector such that $g_i \overset{i.i.d.}{\sim}$ Uniform$(-\sqrt{3}, \sqrt{3})$. Assume a matrix $W \succeq 0$, and denote $U := \sqrt{W}$, and let $U = (u_1, u_2, \ldots, u_d)$. Define the random event as follows:*

$$\mathcal{B} := \left\{ \max_{i \in [d], \|u_i\|_2 > 0} \frac{|u_i^\top g|^2}{\|u_i\|_2^2} \leq C \log d \right\} \tag{6}$$

Then, $\mathcal{B}$ holds true for some absolute constant $C > 0$, with probability at least $1 - 2d^{-3}$.

Proof of Theorem 12. WLOG we assume that $u_i \neq 0_d$ for every $i \in [d]$. Since each g_i is a bounded random variable, we see that $g_i \sim \mathcal{SG}(3)$, i.e., $\mathbb{E}\exp\{tg_i\} \leq \exp\{3t^2/2\}$. Moreover, as each g_i has i.i.d. entries, it is clear that $u_i^\top g \sim \mathcal{SG}(3\|u_i\|_2^2)$, and hence by standard Chernoff bound (see, e.g., Section 4.2 in [34]), for an arbitrary $s > 0$, $\mathbb{P}\left(|u_i^\top g| > s\right) \leq 2\exp\left\{-s^2/(6\|u_i\|_2^2)\right\}$. By union bound, for any $t > 0$, we obtain that

$$\mathbb{P}\left(\max_{i \in [d]} \frac{|u_i^\top g|}{\|u_i\|_2} > t\right) \leq \sum_{i=1}^d \mathbb{P}\left(\frac{|u_i^\top g|}{\|u_i\|_2} > t\right) \leq \sum_{i=1}^d 2\exp\left\{-\frac{t^2}{6}\right\} = 2d\exp\left\{-\frac{t^2}{6}\right\}.$$

Setting $t := 2\sqrt{6}\sqrt{\log d}$ concludes the proof. □

Now, we are ready to give a formal statement of Theorem 2:

Theorem 13. *Let the notation be the same as in Theorem 2, but only assuming that $A \in \mathbb{R}^{d \times d}$ is symmetric with $\|A\|_2 = 1$. Denote random events $\mathcal{A}$ and $\mathcal{B}$ as defined in (5) and (6), respectively, and denote $c > 0$ and $C > 0$ the absolute constants in Theorem 10 and Theorem 12, respectively.*

1. *Suppose that the input matrix A has non-negative diagonal entries, then one has*

$$C \log d \left[1 + \frac{9\left(\sum_{i=1}^d \sqrt{W_{ii}^*}\right)^2}{4k}\right] \mathbb{E}\left[z^\top A z | \mathcal{A} \cap \mathcal{B}\right] \geq \left[1 - \mathcal{O}\left(\frac{1}{d}\right)\right] (x^*)^\top A x^* - \epsilon - e^{-ck + 2\log\left(\frac{2d}{k}\right)}.$$

2. *Suppose, in addition, that the input matrix A is positive semidefinite, and that $ck \geq 3\log(d/k) + \log\log d$. Then,*

$$C \log d \left[1 + \frac{9\left(\sum_{i=1}^d \sqrt{W_{ii}^*}\right)^2}{4k}\right] \mathbb{E}\left[z^\top A z | \mathcal{A} \cap \mathcal{B}\right] \geq \left[1 - \mathcal{O}\left(\frac{1}{\log d}\right)\right] (x^*)^\top A x^* - \epsilon.$$

It is clear that Theorem 13 indeed implies Theorem 2, by setting $\mathcal{R} := \mathcal{A} \cap \mathcal{B}$, and it is clear that the probability that $\mathcal{R}$ occurs is lower bounded by $1 - \exp\{-ck\} - 2d^{-3}$ by Theorems 10 and 12.

We are now ready to prove Theorem 13.

Proof of Theorem 13. In the proof, denote by $\sqrt{W^*} = (u_1, u_2, \ldots, u_d)$. We first point out a fact regarding Algorithm 1: After executing line 6, i.e., setting the set S to be the set $\{i \in [d] : \epsilon_i = 1\}$, if one draws a random vector $g \in \mathbb{R}^d$ such that its entries are all i.i.d. Uniform in $[-\sqrt{3}, \sqrt{3}]$, and defines a random vector $x \in \mathbb{R}^d$ such that $x_i = 0$ for $i \in S^c$, and $x_i = u_i^\top g / p_i$ for $i \in S$, then one can see that $z^\top A z \geq x^\top A x / \|x\|_2^2$. Indeed, this is implied by the fact that Supp$(x) \subseteq$ Supp(z), and z is chosen to the vector with larger SPCA objective on line 9.

The ideas behind the rest of the proof are as follows: The expected objective value $x^\top Ax$ is greater or equal to $\mathrm{tr}(AW^*)$. This, coupled with the facts that $z^\top Az \geq x^\top Ax/\left\|x\right\|_2^2$, $\mathrm{tr}(AW^*) \geq (x^*)^\top Ax^* - \epsilon$, and upper bound of $\mathrm{tr}(AW^*)$, lead to approximation guarantees.

Expected value of $x^\top Ax$. Assuming that $p_i > 0$ for all $i \in [d]$. By definition,

$$x^\top Ax = \sum_{i,j=1}^{d} a_{ij} x_i x_j = \sum_{i,j=1}^{d} a_{ij} \cdot \left(\frac{u_i^\top g}{p_i} \cdot \epsilon_i\right) \cdot \left(\frac{u_j^\top g}{p_j} \cdot \epsilon_j\right).$$

For $i, j \in [d]$ and $i \neq j$, it holds that $\mathbb{E}x_i x_j = \mathbb{E}(u_i^\top g)(u_j^\top g) = u_i^\top \left(\mathbb{E}gg^\top\right) u_j = u_i^\top I_d u_j = u_i^\top u_j$, and for $i = j \in [d]$, we have that $\mathbb{E}x_i^2 = \mathbb{E}(u_i^\top g)^2/p_i = u_i^\top \left(\mathbb{E}gg^\top\right) u_i/p_i = \left\|u_i\right\|_2^2/p_i$. Since $U = \sqrt{W}$, we have that $U^2 = U^\top U = W$, and hence $W_{ij} = u_i^\top u_j$. Combining all above facts, we obtain that

$$\mathbb{E}x^\top Ax = \sum_{i=1}^{d} \frac{a_{ii}}{p_i} \left\|u_i\right\|_2^2 + \sum_{1 \leq i \neq j \leq d} a_{ij} u_i^\top u_j = \mathrm{tr}\left(AW\right) + \sum_{i=1}^{d} \left(\frac{1}{p_i} - 1\right) a_{ii} \left\|u_i\right\|_2^2 \geq \mathrm{tr}(AW^*).$$

In the case where there exist some $p_i = \min\{1, 2/3 \cdot k \left\|u_i\right\|_2 / \sum_{j=1}^d \left\|u_j\right\|_2 + 1/12 \cdot kA_{ii}/\,\mathrm{tr}(A)\} = 0$, it is clear that $x_i = 0$ and $W_{ii} = \left\|u_i\right\|_2^2 = 0$. $W \succeq 0$ implies that $W_{ij} = W_{ji} = 0$ for all $j \in [d]$, and thus $\mathbb{E}x_i x_j = 0 = W_{ij}$, and thus $\mathbb{E}x^\top Ax \geq \mathrm{tr}(AW^*)$ still holds.

Upper bound of $\mathrm{tr}(AW^*)$. By law of total expectation, we obtain that

$$\begin{aligned} \mathbb{E}x^\top Ax &= \mathbb{P}(\mathcal{A}) \cdot \mathbb{E}\left[x^\top Ax|\mathcal{A}\right] + \mathbb{P}(\mathcal{A}^c) \cdot \mathbb{E}\left[x^\top Ax|\mathcal{A}^c\right] \\ &\leq \mathbb{P}(\mathcal{A}) \cdot \mathbb{E}\left[x^\top Ax|\mathcal{A}\right] + \exp\{-ck\} \cdot \left\|A\right\|_2 \cdot \mathbb{E}\left[x^\top x|\mathcal{A}^c\right] \\ &= \mathbb{P}(\mathcal{A}) \cdot \mathbb{E}\left[x^\top Ax|\mathcal{A}\right] + \exp\{-ck\} \cdot \mathbb{E}\left[x^\top x|\mathcal{A}^c\right] \end{aligned}$$

It suffices to further upper bound $\mathbb{P}(\mathcal{A}) \cdot \mathbb{E}\left[x^\top x|\mathcal{A}^c\right]$. Again by law of total expectation, one has

$$\mathbb{P}(\mathcal{A}) \cdot \mathbb{E}\left[x^\top Ax|\mathcal{A}\right] = \mathbb{P}\left(\mathcal{A} \cap \mathcal{B}\right) \cdot \mathbb{E}\left[x^\top Ax|\mathcal{A} \cap \mathcal{B}\right] + \mathbb{P}\left(\mathcal{A} \cap \mathcal{B}^c\right) \cdot \mathbb{E}\left[x^\top Ax|\mathcal{A} \cap \mathcal{B}^c\right] \tag{7}$$

We upper bound $\left\|x\right\|_2^2$ conditioned on $\mathcal{A} \cap \mathcal{B}$ first. Pick any $x \in \mathcal{A}$, define $R := \mathrm{Supp}(x)$ and $T := \{i \in R : p_i = 1\}$. Notice that $\sqrt{W_{ii}^*} = \left\|u_i\right\|_2$ and $p_i \geq 2/3 \cdot k \left\|u_i\right\|_2 / \sum_{j=1}^d \left\|u_i\right\|_2$, and thus

$$\begin{aligned} \left\|x\right\|_2^2 &= \sum_{i \in T} (u_i^\top g)^2 + \sum_{i \in R \setminus T} \frac{(u_i^\top g)^2}{p_i^2} \leq \sum_{i \in T} (u_i^\top g)^2 + \frac{9}{4k^2} \left(\sum_{i=1}^{d} \left\|u_i\right\|_2\right)^2 \sum_{i \in R \setminus T} \frac{(u_i^\top g)^2}{\left\|u_i\right\|_2^2} \\ &\overset{\mathcal{B}}{\leq} \sum_{i \in S} (C \log d) \cdot \left\|u_i\right\|_2^2 + \frac{9}{4k^2} \left(\sum_{i=1}^{d} \left\|u_i\right\|_2\right)^2 \cdot k \cdot C \log d \leq C \log d \left[1 + \frac{9}{4k} \left(\sum_{i=1}^{d} \left\|u_i\right\|_2\right)^2\right]. \end{aligned}$$

Define $\bar{x} := x/\left\|x\right\|_2$. Hence, for the first term in (7), it is clear that

$$\mathbb{E}[x^\top Ax|\mathcal{A} \cap \mathcal{B}] = \mathbb{E}[\left\|x\right\|_2^2 \bar{x}^\top A\bar{x}|\mathcal{A} \cap \mathcal{B}] \leq C \log d\Big(1 + \frac{9\left(\sum_{i=1}^d \left\|u_i\right\|_2\right)^2}{4k}\Big)\mathbb{E}\left[\bar{x}^\top A\bar{x}|\mathcal{A} \cap \mathcal{B}\right].$$

Next, we upper bound the second term in (7). By Theorem 12, it is clear that $\mathbb{P}(\mathcal{B}^c) \leq 2d^{-3}$, thus

$$\mathbb{P}\left(\mathcal{A} \cap \mathcal{B}^c\right) \mathbb{E}\left[x^\top Ax|\mathcal{A} \cap \mathcal{B}^c\right] \leq \frac{2}{d^3} \mathbb{E}\left[x^\top Ax|\mathcal{A} \cap \mathcal{B}^c\right] \leq \frac{2}{d^3} \left[(x^*)^\top Ax^*\right] \mathbb{E}\left[\left\|x\right\|_2^2 |\mathcal{A} \cap \mathcal{B}^c\right].$$

Then, we give an upper bound for $\mathbb{E}\left[\left\|x\right\|_2^2 |\mathcal{A} \cap \mathcal{B}^c\right]$. By writing down $\left\|x\right\|_2^2$ explicitly, and by the fact that $(u_i^\top g)^2 \leq \left\|u_i\right\|_2^2 \left\|g\right\|_2^2$, we observe that

$$\mathbb{E}\left[\left\|x\right\|_2^2 |\mathcal{A} \cap \mathcal{B}^c\right] \leq \mathbb{E}\left[\sum_{i \in T} (u_i^\top g)^2 + \frac{9}{4k^2} \left(\sum_{i=1}^{d} \left\|u_i\right\|_2\right)^2 \cdot \sum_{i \in R \setminus T} \frac{(u_i^\top g)^2}{\left\|u_i\right\|_2^2} \middle| \mathcal{A} \cap \mathcal{B}^c\right]$$

$$\le \mathbb{E}\left[\sum_{i\in T}\|u_i\|_2^2\|g\|_2^2 + \frac{9}{4k^2}\left(\sum_{i=1}^d \|u_i\|_2\right)^2 \cdot \sum_{i\in R\setminus T}\frac{\|u_i\|_2^2\|g\|_2^2}{\|u_i\|_2^2}\middle|\mathcal{A}\cap\mathcal{B}^c\right]$$
$$\le \mathbb{E}\left[\|g\|_2^2 + \frac{9}{4k^2}\left(\sum_{i=1}^d\|u_i\|_2\right)^2\cdot\sum_{i\in R\setminus T}\|g\|_2^2\middle|\mathcal{A}\cap\mathcal{B}^c\right] \le 3d + \frac{27d}{4k^2}\left(\sum_{i=1}^d\|u_i\|_2\right)^2,$$

where the last inequality follows from $g_i \sim \text{Uniform}(-\sqrt{3},\sqrt{3})$ and hence $\|g\|_2^2 \le 3d$. Finally, combining all bounds above, we obtain that $\mathbb{P}(\mathcal{A})\cdot\mathbb{E}\left[x^\top Ax|\mathcal{A}\right]$ is upper bounded by

$$C\log d\mathbb{P}(\mathcal{A}\cap\mathcal{B})\Big(1+\frac{9(\sum_{i=1}^d\|u_i\|_2)^2}{4k}\Big)\mathbb{E}[\bar{x}^\top A\bar{x}|\mathcal{A}\cap\mathcal{B}] + (x^*)^\top Ax^*\Big(\frac{6}{d^2}+\frac{27(\sum_{i=1}^d\|u_i\|_2)^2}{2k^2d^2}\Big).$$

Hence, we see that $\text{tr}(AW^*)$ is upper bounded by the sum of $\exp\{-ck\}\mathbb{E}\left[x^\top x|\mathcal{A}^c\right]$ and the above quantity. We are now ready to show the two parts in the statement of Theorem 13.

(Proof for part 1) We write $\delta := \exp\{-ck + 2\log 2d/k\}$. Direct calculation shows that

$$\mathbb{E}\left[x^\top x|\mathcal{A}^c\right] = \mathbb{E}\left[\sum_{i:p_i>0}\frac{(u_i^\top g)^2}{p_i^2}\epsilon_i\middle|\mathcal{A}^c\right] \le \mathbb{E}\left[\sum_{i:p_i>0}\frac{(u_i^\top g)^2}{p_i^2}\middle|\mathcal{A}^c\right] = \sum_{i:p_i>0}\frac{\mathbb{E}(u_i^\top g)^2}{p_i^2}$$
$$\le \sum_{i:p_i=1}\frac{\mathbb{E}(u_i^\top g)^2}{p_i^2} + \sum_{i:0<p_i<1}\frac{\mathbb{E}(u_i^\top g)^2}{p_i^2} \le 1 + \frac{9d}{4k^2}\left(\sum_{i=1}^d\|u_i\|_2\right)^2.$$

By the definition of δ, one obtains that $\exp\{-ck\}\mathbb{E}\left[x^\top x|\mathcal{A}^c\right]\le\delta$.

(Proof for part 2) For this part, we assume that $A\succeq 0$. In this special case, we claim that $1 = \|A\|_2 \le \frac{d}{k}\cdot(x^*)^\top Ax^*$. This inequality holds true for the following reasoning:

- first, there exists a $(d-1)\times(d-1)$ principal submatrix $\tilde{A}$ of A such that $\|A\|_2 \le d/(d-1)\cdot\left\|\tilde{A}\right\|_2$ (for a reference, see, e.g., the arguments on page 189 in [25]);
- then, using the above fact repeatedly for the existence of a principal submatrix of size $d-2, d-3\ldots,k$, one can obtain a principal submatrix $\hat{A}$ of size k such that $\|A\|_2\le d/k\cdot\left\|\hat{A}\right\|_2$.

Combining the above inequality and the inequality in the proof for part 1, one obtains that

$$\exp\{-ck\}\mathbb{E}\left[x^\top x|\mathcal{A}^c\right] = \exp\{-ck\}\mathbb{E}\left[x^\top x|\mathcal{A}^c\right]\cdot\|A\|_2$$
$$\le \exp\{-ck\}\cdot\left[1+\frac{9d}{4k^2}\left(\sum_{i=1}^d\|u_i\|_2\right)^2\right]\cdot\frac{d}{k}\cdot(x^*)^\top Ax^* \le \mathcal{O}\left(\frac{1}{\log d}\right)\cdot(x^*)^\top Ax^*,$$

where the last inequality comes from the assumption that $ck\ge 3\log d/k + \log\log d$. □

6.2 Proofs in Section 3.3

In this section, we prove Theorem 6 and Theorem 7 in Section 3.3. To prove them, we need several technical lemmas. We first introduce the core lemma, Theorem 14, which is derived directly from the well-known KKT conditions [29].

Lemma 14. *$W^* = w^*(w^*)^\top$ is an optimal solution to SPCA-SDP if the following conditions hold:*

B1 (Primal feasibility) $\|w^*\|_1\le\sqrt{k}$ and $\|w^*\|_2 = 1$;

B2 (Dual feasibility) $\mu^*\ge 0$, $\lambda^*\in\mathbb{R}$, $Z^*\in[-1,1]^{d\times d}$ such that $Z_{ij}^* = \text{sign}(w_i^*)\cdot\text{sign}(w_j^*)$ if $w_i^*w_j^*\neq 0$, and $\lambda^* I_d\succeq A-\mu^*Z^*$;

B3 (Complementary slackness) $\mu^*(\|w^*\|_1 - \sqrt{k}) = 0$, $(\lambda^* I_d - A + \mu^* Z^*)w^* = 0$.

Moreover, if $\operatorname{rank}(\lambda^* I_d - A + \mu^* U^*) = d - 1$*, then* W^* *is the unique optimal solution to SPCA-SDP.*

Proof. It is clear that the dual problem to SPCA-SDP is given by

$$\min_{\mu,\lambda,S,U} k\mu + \lambda \qquad \text{s.t. } \mu \geq 0,\ S \succeq 0,\ U = A + S - \lambda I_d,\ \|U\|_\infty \leq \mu,$$

and hence conditions B1, B2, and B3 are clear from KKT conditions [29]. Finally, if $\operatorname{rank}(\lambda^* I_d - A + \mu^* U^*) = d - 1$, we know that for any optimal solution W_0 to SPCA-SDP, it must hold that $\operatorname{rank}(W_0) = d - \operatorname{rank}(\lambda^* I_d - A + \mu^* U^*) = 1$ since $\operatorname{tr}(W_0(\lambda^* I_d - A + \mu^* U^*)) = 0$. Due to the fact that $(\lambda^* I_d - A + \mu^* U^*)w^* = 0$, we see that W_0 must be equal to $w^*(w^*)^\top$. □

We are now ready to prove Theorem 6.

Proof of Theorem 6. WLOG, we assume that $\|u\|_2 = 1$, as a scaling of A by a non-negative constant would not change its optimal solution to SPCA-SDP. Furthermore, we assume WLOG that $\lambda = 0$, as adding a scaling of an identity matrix to A would not change its optimal solution to SPCA-SDP either. We complete the proof by utilizing Theorem 14.

(Primal feasibility). For simplicity, we define $t_0 := \min_{i \in T} |u_i|$, with $T = \operatorname{Supp}(u)$. Since $\|u\|_1 > \sqrt{k}$, it is clear that $|T| \geq \|u\|_1^2 > k$. We define a unit vector $w(t) := (u - t \operatorname{sign}(u)) / \|u - t \operatorname{sign}(u)\|_2 \in \mathbb{R}^d$ with $t \in [0, t_0]$. We claim that $\|w(t_0)\|_1 < \sqrt{k}$. Indeed, the claim is equivalent to the quadratic inequality $|T|(|T| - k)t_0^2 - 2\|u\|_1 (|T| - k)t_0 + (\|u\|_1^2 - k) < 0$, which is then implied by (2). Therefore, by the Intermediate Value Theorem, there exists $t^* \in (0, t_0)$ such that $\|w(t^*)\|_1 = \sqrt{k}$.

We thus define $w^* = w(t^*)$, it is clear that $\|w^*\|_2 = 1$, $\|w^*\|_1 = \sqrt{k}$, and $\operatorname{Supp}(w^*) = T$.

(Dual feasibility). By the definition of w^*, we see that $\operatorname{sign}(w^*) = \operatorname{sign}(u)$, and we obtain that

$$u = \|u - t^* \operatorname{sign}(u)\|_2 w^* + t^* \operatorname{sign}(w^*) := \alpha w^* + \beta \operatorname{sign}(w^*),$$

where $\alpha, \beta > 0$ by definition. We then define

$$\lambda^* := \|u - t^* \operatorname{sign}(u)\|_2 (u^\top w^*) = \alpha(u^\top w^*) = \alpha(\alpha + \beta\sqrt{k}) > 0$$

$$\mu^* := \frac{t^*(u^\top w^*)}{\sqrt{k}} = \frac{\beta(u^\top w^*)}{\sqrt{k}} = \frac{\beta(\alpha + \beta\sqrt{k})}{\sqrt{k}} > 0.$$

We further define $Z^* = \operatorname{sign}(u) \operatorname{sign}(u)^\top$, and we show that

$$\lambda^* I_d \succeq uu^\top - \mu^* Z^* \iff \lambda^* I_d \succeq uu^\top - \mu^* \operatorname{sign}(u) \operatorname{sign}(u)^\top.$$

For simplicity, define $M := \lambda^* I_d - uu^\top + \mu^* \operatorname{sign}(u) \operatorname{sign}(u)^\top$. Since $u \in \operatorname{Span}(w^*, \operatorname{sign}(w^*))$, it is clear that for any non-zero vector $v \in \operatorname{Span}(w^*, \operatorname{sign}(w^*))^\perp$, $v^\top M v = \lambda^* \|v\|_2^2 > 0$. Next, take any non-zero vector $v \in \operatorname{Span}(w^*, \operatorname{sign}(w^*))$, we also show that $v^\top M v \geq 0$. First, we observe that

$$Mw^* = (\lambda^* I_d - uu^\top + \mu^* \operatorname{sign}(u) \operatorname{sign}(u)^\top)w^* = \lambda^* w^* - (u^\top w^*)u + \mu^* \sqrt{k} \operatorname{sign}(u) = 0,$$

where the last equality follows from the fact that $u = \alpha w^* + \beta \operatorname{sign}(w^*) = \alpha w^* + \beta \operatorname{sign}(u)$ and how λ^* and μ^* are defined. For simplicity, define $s := \operatorname{sign}(w^*) = \operatorname{sign}(u)$, and we notice that

$$s^\top M s = s^\top \left(\lambda^* I_d - uu^\top + \mu^* \operatorname{sign}(u) \operatorname{sign}(u)^\top\right) s = (\lambda^* + \mu^* |T|)|T| - \|u\|_1^2$$

$$= \alpha^2 + \alpha\beta\sqrt{k} + \frac{\alpha\beta|T|}{\sqrt{k}} + \beta^2 |T|^2 - (\alpha\sqrt{k} + \beta|T|)^2 = \alpha^2(|T| - k) + \frac{\alpha\beta|T|}{\sqrt{k}}(|T| - k) > 0.$$

Thus, for any $v = aw^* + b \operatorname{sign}(w^*)$, we see that $v^\top M v = b^2 \operatorname{sign}(w^*)^\top M \operatorname{sign}(w^*) \geq 0$.

Hence, we have shown that $M \succeq 0$, with its smallest eigenvalue being zero, and a positive second smallest eigenvalue. This implies that $\operatorname{rank}(M) = d - 1$.

(Complementary slackness). The complementary slackness conditions follow from the fact that $\|w^*\|_1 = \sqrt{k}$ and $Mw^* = 0$. □

Before we jump in the proof of Theorem 7, let us explain the high-level ideas. To utilize Theorem 14, the key is to construct a feasible primal solution w^* with great property. Since we are not able to present a closed-form solution of w^*, we instead show that, such w^* has a special parameterized form $w^* = w^*(\lambda)$, where the only parameter is $\lambda \in \mathbb{R}$. We show that there must exists an interval I of λ such that for some $\lambda^* \in I$, $w^*(\lambda^*)$ satisfies primal feasibility. The interval is given in Theorem 15. Since we do not know what the exact value of w^*, or equivalently, λ^*, we additionally need A2 in Assumption 1 helps keep the sign of $w^*(\lambda)$ unchanged in that interval.

In the next lemma, we discuss the property of a special function:

Lemma 15. *Under Assumption 1, consider the vector* $w(\lambda) := (A_{S,S} - \lambda I_{|S|})^{-1} \operatorname{sign}(v_1)$, *and define the function* $R(\lambda) := \|w(\lambda)\|_1 / \|w(\lambda)\|_2$ *for* $\lambda \in (\lambda_2, \lambda_1)$. *Then, there exists a* $\lambda^* \in [\lambda_1/(D+1), \lambda_1) \subseteq (\lambda_2, \lambda_1)$ *such that* $R(\lambda^*) = \sqrt{k}$, *where* $D = \sqrt{\frac{4\|v_1\|_1^4 - k\|v_1\|_1^2}{k\sum_{i=2}^{|S|}\alpha_i^2}}$ *as in Assumption 1. Moreover, the constant* $\delta > 0$ *indeed exists for A3 in Assumption 1, and we have that*

$$\frac{2\|v_1\|_1^2}{\delta} \geq \left\|(A_{S,S} - \lambda^* I_{|S|})^{-1} \operatorname{sign}(v_1)\right\|_2 \geq \frac{(1-\gamma)\|v_1\|_1^2}{\sqrt{|S|}\lambda_1} \cdot \frac{D+1}{D}.$$

Proof. We first show that the function $R(\lambda)$ is continuous on the interval (λ_2, λ_1). Indeed, we see that the matrix $(A_{S,S} - \lambda I_{|S|})^{-1} = \sum_{i=1}^{|S|} \frac{1}{\lambda_i - \lambda} v_i v_i^\top$ is continuous on the interval, and thus $R(\lambda)$.

To show the existence of λ^*, we use the intermediate value theorem for continuous function. First, we observe that $(A_{S,S} - \lambda I_{|S|})^{-1} \operatorname{sign}(v_1) = \sum_{i=1}^{|S|} \frac{v_i^\top \operatorname{sign}(v_1)}{\lambda_i - \lambda} v_i$. Since $v_1^\top \operatorname{sign}(v_1) = \|v_1\|_1 > \sqrt{k} > 0$, we see that when $\lambda \to \lambda_1-$, the vector is dominated by $\|v_1\|_1/(\lambda_1 - \lambda) \cdot v_1$, and hence $R(\lambda) \to \|v_1\|_1 / \|v_1\|_2 = \|v_1\|_1 > \sqrt{k}$.

Thus, to use intermediate value theorem, it suffices to find a $\lambda_* \in (\lambda_2, \lambda_1)$ such that $R(\lambda_*) \leq \sqrt{k}$. We take $\lambda_* = \lambda_1/(D+1)$, where D is defined in Assumption 1, i.e., $D = \sqrt{\frac{4\|v_1\|_1^4 - k\|v_1\|_1^2}{k\sum_{i=2}^{|S|}\alpha_i^2}}$. From A1, it is clear that $\lambda_* \in (\lambda_2, \lambda_1)$. Along with A2 in Assumption 1, we see that

$$\begin{aligned}R(\lambda_*) &= \frac{\sum_{j=1}^{|S|}\left|\frac{\|v_1\|_1}{\lambda_1-\lambda_*} v_{1,j} + \sum_{i=2}^{|S|}\frac{\alpha_i}{\lambda_i-\lambda_*} v_{i,j}\right|}{\sqrt{\sum_{i=1}^{|S|}\left(\frac{\alpha_i}{\lambda_i-\lambda_*}\right)^2}} = \frac{\sum_{j=1}^{|S|}\left|\|v_1\|_1 v_{1,j} + \sum_{i=2}^{|S|}\frac{\lambda_1-\lambda_*}{\lambda_i-\lambda_*} v_{i,j}\right|}{\sqrt{\sum_{i=1}^{|S|}\alpha_i^2\left(\frac{\lambda_1-\lambda_*}{\lambda_i-\lambda_*}\right)^2}}\\ &\leq \frac{2\|v_1\|_1^2}{\sqrt{\sum_{i=1}^{|S|}\alpha_i^2\left(\frac{\lambda_1-\lambda_*}{\lambda_i-\lambda_*}\right)^2}} \leq \frac{2\|v_1\|_1^2}{\sqrt{\|v_1\|_1^2 + \sum_{i=2}^{|S|}\alpha_i^2\left(\frac{\lambda_1-\lambda_*}{\lambda_*}\right)^2}} = \frac{2\|v_1\|_1^2}{\sqrt{\frac{4\|v_1\|_1^4}{k}}} = \sqrt{k}.\end{aligned}$$

To see the existence of $\delta > 0$ in Assumption 1, we use the previous observation that $R(\lambda)$ is a continuous function on the interval (λ_2, λ_1), and that $R(\lambda) \to \|v_1\|_1 > \sqrt{k}$ as $\lambda \to \lambda_1-$. Then, we give a lower bound for $\left\|(A_{S,S} - \lambda I_{|S|})^{-1} \operatorname{sign}(v_1)\right\|_2$. By Cauchy-Schwarz inequality and by A2,

$$\left\|(A_{S,S} - \lambda I_{|S|})^{-1} \operatorname{sign}(v_1)\right\|_2 \leq \left\|(A_{S,S} - \lambda I_{|S|})^{-1} \operatorname{sign}(v_1)\right\|_1 \leq \frac{2}{\lambda - \lambda^*}\|v_1\|_1^2 \leq \frac{2}{\delta}\|v_1\|_1^2$$

Finally, by Cauchy-Schwarz inequality and again by A2, we have that

$$\begin{aligned}\left\|(A_{S,S} - \lambda I_{|S|})^{-1} \operatorname{sign}(v_1)\right\|_2 &\geq \frac{1}{\sqrt{|S|}} \cdot \left\|(A_{S,S} - \lambda I_{|S|})^{-1} \operatorname{sign}(v_1)\right\|_1 \geq \frac{(1-\gamma)\|v_1\|_1^2}{\sqrt{|S|}\cdot(\lambda_1 - \lambda^*)}\\ &\geq \frac{(1-\gamma)\|v_1\|_1^2}{\sqrt{|S|}\cdot(\lambda_1 - \lambda_*)} = \frac{(1-\gamma)\|v_1\|_1^2}{\sqrt{|S|}\lambda_1} \cdot \frac{D+1}{D}.\end{aligned}$$

□

Finally, we are ready to prove Theorem 7.

Proof of Theorem 7. We complete the proof by utilizing Theorem 14.

(Primal feasibility). We construct the vector w^* as follows: first, let S be the set in Assumption 1, and let $S^c := [d]\backslash S$. We set the entries of $w^*_{S^c}$ to be all zero, and choose $w^*_S := (A_{S,S} - \lambda^* I_{|S|})^{-1}\operatorname{sign}(v_1)/\left\|(A_{S,S} - \lambda^* I_{|S|})^{-1}\operatorname{sign}(v_1)\right\|_2$, where λ^* is the one in Theorem 15. By Theorem 15, it is clear that $\|w^*\|_2 = \|w^*_S\|_2 = 1$, and $\|w^*\|_1 = \|w^*_S\|_1 = \sqrt{k}$.

(Dual feasibility). We first set $\mu^* := 1/\sqrt{k}\cdot\left\|(A_{S,S} - \lambda^* I_{|S|})^{-1}\operatorname{sign}(v_1)\right\|_2 \geq 0$. Then, with a bit abuse of notation, we set λ^* to be exactly the one in Theorem 15.

For Z^*, we first make the following observation: A2 implies that the sign of w^*_S is exactly the same as v_1, i.e., $\operatorname{sign}(w^*_S) = \operatorname{sign}(v_1)$. Therefore, we set $Z^*_{S,S} := \operatorname{sign}(w^*_S)\operatorname{sign}(w^*_S)^\top = \operatorname{sign}(v_1)\operatorname{sign}(v_1)^\top$. For the remaining $(i,j) \in [d]\times[d]$, with $i,j \notin S$, we set $Z^*_{ij} = A_{ij}/\mu^*$. This implies $Z^*_{ij} \in [-1,1]$, as we can see in A3 in Assumption 1, one has

$$\max\left\{\max_{\substack{i\in S,\\ j\in S^c}}|A_{ij}|, \max_{i,j\in S^c}|A_{ij}|\right\} \leq \frac{\delta}{2\sqrt{k}\,\|v_1\|_1^2} \leq \frac{1}{\sqrt{k}\left\|(A_{S,S} - \lambda^* I_{|S|})^{-1}\operatorname{sign}(v_1)\right\|_2} = \mu^*,$$

where the last inequality is from Theorem 15. We leave the proof of $\lambda^* I_d \succeq A - \mu^* Z^* \succeq 0$ to the last part in this proof.

(Complementary Slackness). It is clear that $\mu^*(\|w^*\|_1 - \sqrt{k}) = 0$ holds as $\|w^*\|_1 = \sqrt{k}$, and hence we only need to show that $(\lambda^* I_d - A + \mu^* Z^*)w^* = 0$. An observation is that the matrix $-A + \mu^* Z^*$ have zero entries outside of the support $S\times S$, and $\operatorname{Supp}(w^*) = S$, and thus we only need to show that $(\lambda^* I_{|S|} - A_{S,S} + \mu^*\operatorname{sign}(w^*_S)\operatorname{sign}(w^*_S)^\top)w^*_S = 0$. This comes from a direct calculation that $\operatorname{sign}(w^*_S)^\top w^*_S = \|w^*_S\|_1 = \sqrt{k}$, and from the definition of w^*_S we see that

$$(A_{S,S} - \lambda^* I_{|S|})w^*_S = \frac{\operatorname{sign}(w^*_S)}{\left\|(A_{S,S} - \lambda^* I_{|S|})^{-1}\operatorname{sign}(v_1)\right\|_2} = \mu^*\cdot\sqrt{k}\cdot\operatorname{sign}(w^*_S).$$

(Proof of $\lambda^* I_d \succeq A - \mu^* Z^*$). In this part, we show that $\lambda^* I_d \succeq A - \mu^* Z^*$. Since $\lambda^* \geq \lambda_2 \geq 0$, and the matrix $-A + \mu^* Z^*$ have zero entries outside of the support $S\times S$, it suffices to show that $\lambda^* I_{|S|} \succeq A_{S,S} - \mu^*\operatorname{sign}(v_1)\operatorname{sign}(v_1)^\top$.

Define the i-th largest eigenvalue of $\lambda^* I_{|S|} - A_{S,S} + \mu^*\operatorname{sign}(v_1)\operatorname{sign}(v_1)^\top$ to be ν_i, and define the i-th largest eigenvalue of $\lambda^* I_{|S|} - A_{S,S}$ to be γ_i. By the fact that $(\lambda^* I_{|S|} - A_{S,S} + \mu^*\operatorname{sign}(w^*_S)\operatorname{sign}(w^*_S)^\top)w^*_S = 0$, it is clear that $\nu_{|S|} = 0$. Therefore, it suffices to show that $\nu_{|S|-1} \geq 0$. By Weyl's inequality, it is clear that $\left|\nu_{|S|-1} - \gamma_{|S|-1}\right| \leq \left\|\mu^*\operatorname{sign}(w^*_S)\operatorname{sign}(w^*_S)^\top\right\|_2 = \mu^*\cdot k$. Since $\gamma_{|S|-1} = \lambda^* - \lambda_2 \geq 0$, by Theorem 15, and then by A1, we see that

$$\begin{aligned}\nu_{|S|-1} &\geq \gamma_{|S|-1} - \mu^*\cdot k = \lambda^* - \lambda_2 - \mu^*\cdot k \geq \frac{\lambda_1}{D+1} - \lambda_2 - \frac{\sqrt{k}}{\left\|(A_{S,S} - \lambda^* I_{|S|})^{-1}\operatorname{sign}(v_1)\right\|_2}\\ &\geq \frac{\lambda_1}{D+1} - \lambda_2 - \frac{\sqrt{k}}{\frac{(1-\gamma)\|v_1\|_1^2}{\sqrt{|S|}\lambda_1}\cdot\frac{D+1}{D}} = \frac{\lambda_1}{D+1}\left(1 - \frac{D\sqrt{k\cdot|S|}}{(1-\gamma)\,\|v_1\|_1^2}\right) - \lambda_2 \geq 0\end{aligned}$$

Finally, if the inequality in A1 in Assumption 1 is strict, then we see that $\nu_{|S|-1} > 0$, and hence the matrix $\lambda^* I_d - A + \mu^* U^*$ has rank $d-1$. Therefore, $W^* = w^*(w^*)^\top$ is the unique optimal solution to SPCA-SDP. □

7 Proofs in Section 4

In this section, we present the proofs of Theorems 8 and 9 in Section 4.

We first provide the proof of Theorem 8. In order to do so, we need several propositions. The first proposition characterizes how close W^* is to the "sparse spike" in a deterministic model:

Proposition 16. *Suppose the input matrix in SPCA can be written as $A = B^\top B + E$, where $B^\top B$ admits a k-sparse eigenvector v corresponding to its largest eigenvalue $\lambda_1(B^\top B)$, and $E \in \mathbb{R}^{d\times d}$ is a matrix such that $\|E\|_\infty \le a$. Assume that $\lambda_1(B^\top B) - \lambda_2(B^\top B) > 0$, and denote W^* an optimal solution to SPCA-SDP. Then, we have*

$$\|W^* - vv^\top\|_F \le \frac{2ak}{\lambda_1 - \lambda_2} + \sqrt{\frac{2ak}{\lambda_1 - \lambda_2}}$$

To prove Theorem 16, we utilize the curvature lemma presented in [41]. This lemma helps transoform the problem of bounding the Frobenius distance of W^* and $vv^\top$ into bounding the difference of their objective values of SPCA-SDP, which can be done at more ease.

Lemma 17 (Lemma 3.1 in [41])**.** *Let B be a symmetric $d \times d$ matrix and P be the projection onto the subspace spanned by the eigenvectors of B corresponding to its l largest eigenvalues $\lambda_1 \ge \lambda_2 \ge \cdots \ge \lambda_l$. If $\delta := \lambda_l - \lambda_{l+1} > 0$, then for any symmetric matrix F satisfying $0 \preceq F \preceq I_d$ and* $\mathrm{tr}(F) = l$, *we have*

$$\|P - F\|_F^2 \le \frac{2}{\delta}\,\mathrm{tr}(B(P - F)).$$

Remark. *In Theorem 17, suppose B admits a singular value decomposition $B = \sum_{i=1}^d \lambda_i u_i u_i^\top$, then the matrix $P = \sum_{i=1}^l u_i u_i^\top$.*

Proof of Theorem 16. In the proof, we will denote $\lambda_1 := \lambda_1(B^\top B)$ and $\lambda_2 := \lambda_2(B^\top B)$, i.e., the largest and the second largest eigenvalue of $B^\top B$, respectively. By Theorem 17, we obtain that

$$\left\|vv^\top - W^*\right\|_F^2 \le \frac{2}{\lambda_1 - \lambda_2}\,\mathrm{tr}\left(B^\top B(vv^\top - W^*)\right) = \frac{2}{\lambda_1 - \lambda_2}\,\mathrm{tr}\left((A - E)(vv^\top - W^*)\right).$$

Since $vv^\top$ is feasible to problem SPCA-SDP, we have $\mathrm{tr}(AW^*) \ge \mathrm{tr}(Avv^\top)$. Combining this with the above inequality, we see that

$$\left\|vv^\top - W^*\right\|_F^2 \le \frac{2}{\lambda_1 - \lambda_2}\left[-\mathrm{tr}\left(E(vv^\top - W^*)\right)\right] \le \frac{2\left\|E\right\|_\infty}{\lambda_1 - \lambda_2}\left\|vv^\top - W^*\right\|_1 \le \frac{2a\left\|vv^\top - W^*\right\|_1}{\lambda_1 - \lambda_2}.$$

Let $\hat{W}^*_{S^c}$ be a matrix such that it is zero out in (S,S) block, and it coincides with the remaining entries in W^*. By Cauchy-Schwartz inequality and the fact that $\left\|\hat{W}^*_{S^c}\right\|_1 \le k$, we obtain that

$$\left\|vv^\top - W^*\right\|_1 = \left\|v_S v_S^\top - W^*_{S,S}\right\|_1 + \left\|\hat{W}^*_{S^c}\right\|_1 \le k\left\|v_S v_S^\top - W^*_{S,S}\right\|_F + k$$

Therefore, we see that

$$\left\|vv^\top - W^*\right\|_F^2 \le \frac{2ak}{\lambda_1 - \lambda_2}\left\|v_S v_S^\top - W^*_{S,S}\right\|_F + \frac{2ak}{\lambda_1 - \lambda_2}.$$

We conclude the proof by noticing that, for some non-negative number $c \ge 0$, the quadratic inequality $x^2 \le cx + c$ implies that $x \le (c + \sqrt{c^2 + 4c})/2 \le c + \sqrt{c}$. □

To apply Theorem 16 to Model 1, we need the following high probability bound:

Lemma 18. *In Model 1, denote λ_1, λ_2 to be the largest and second largest eigenvalue of Σ, respectively, and assume $\lambda_1 - \lambda_2 > 0$. Write*

$$A = n\Sigma + \left(B^\top B - n\Sigma\right) + \left(M^\top M + M^\top B + B^\top M\right) =: n\Sigma + E. \tag{8}$$

Then,

$$\frac{1}{n}\left\|E\right\|_\infty \le C\sigma^2\left(\sqrt{\frac{\log d}{n}} + \frac{\log d}{n}\right) + \frac{b^2}{n} + \frac{2b}{\sqrt{n}} \cdot \sqrt{C\sigma^2\left(\sqrt{\frac{\log d}{n}} + \frac{\log d}{n}\right) + \max_{i\in[d]}\Sigma_{ii}} \tag{9}$$

with probability at least $1 - \mathcal{O}(d^{-10})$ for some absolute constant $C > 0$.

Proof. We first recall that each row of B has zero-mean and admits a covariance matrix Σ in Model 1. As every single entry of $B^\top B$ is a summation of sub-Gaussian random variables with parameter σ^2, then by Bernstein's inequality (see, e.g., Theorem 2.8.1 in [40]) and an argument of union bound, we obtain that

$$\left\|\frac{1}{n}B^\top B - \Sigma\right\|_\infty \le C\sigma^2\left(\sqrt{\frac{\log d}{n}} + \frac{\log d}{n}\right) \tag{10}$$

with probability at least $1-d^{-10}$ for some absolute constant $C > 0$. By Cauchy Schwartz inequality, we also see that

$$\left\|B^\top M\right\|_\infty \le \left\|B\right\|_{1\to 2}\left\|M\right\|_{1\to 2} \le b\left\|B\right\|_{1\to 2}, \qquad \left\|M^\top M\right\|_\infty \le \left\|M\right\|_{1\to 2}^2 \le b^2 \tag{11}$$

Then, notice that

$$\frac{1}{n}\left\|B\right\|_{1\to 2}^2 = \max_{i\in[d]}\left(\frac{1}{n}B^\top B\right)_{ii} \overset{(10)}{\le} C\sigma^2\left(\sqrt{\frac{\log d}{n}} + \frac{\log d}{n}\right) + \max_{i\in[d]}\Sigma_{ii} \tag{12}$$

with probability at least $1 - d^{-10}$. Recall that we write

$$A = n\Sigma + \left(B^\top B - n\Sigma\right) + \left(M^\top M + M^\top B + B^\top M\right) = n\Sigma + E.$$

Combining (10), (11), (12), we see that

$$\frac{1}{n}\left\|E\right\|_\infty \le C\sigma^2\left(\sqrt{\frac{\log d}{n}} + \frac{\log d}{n}\right) + \frac{b^2}{n} + \frac{2b}{\sqrt{n}} \cdot \sqrt{C\sigma^2\left(\sqrt{\frac{\log d}{n}} + \frac{\log d}{n}\right) + \max_{i\in[d]}\Sigma_{ii}}.$$

□

Next, we establish the following lemma, which transforms the upper bound of $\left\|E\right\|_\infty/n$ studied in Theorem 18 into a user-friendly bound:

Lemma 19. *Define $f := f(C, \Sigma, \sigma, d, n)$ to be the RHS of (9). Assume that $\lambda_1 - \lambda_2 > 0$. There exists an absolute constant $C^* > 0$ such that when*

$$n \ge n^* := \max\left\{C^* \cdot \left[\frac{k^2\sigma^4\log d + b^2k^2\left(\sigma^2 + \max\Sigma_{ii}\right)}{(\lambda_1 - \lambda_2)^2a^4} + \frac{kb^2}{(\lambda_1 - \lambda_2)a^2}\right], \log d\right\},$$

one has $f \le (\lambda_1 - \lambda_2)\cdot a^2/(8k)$ with probability at least $1 - d^{-10}$. Moreover, if for $l \ge 1$, $n \ge l \cdot n^*$, then $f \le (\lambda_1 - \lambda_2)\cdot a^2/(8\sqrt{l}k)$ with probability at least $1 - d^{-10}$.

Proof. Since $n \ge \log d$, one has $1 \ge \sqrt{\frac{\log d}{n}} \ge \frac{\log d}{n}$. By the fact that $\sqrt{x+y} \le \sqrt{x} + \sqrt{y}$ for non-negative scalars x, y, it suffices to show that

$$2C\sigma^2\sqrt{\frac{\log d}{n}} + \frac{b^2}{n} + \frac{2b}{\sqrt{n}} \cdot \left(\sqrt{2C\sigma^2} + \sqrt{\max_{i\in[d]}\Sigma_{ii}}\right) \le \frac{\lambda_1 - \lambda_2}{8k} \cdot a^2. \tag{13}$$

It is then clear that for some large $C^* > 0$, (13) holds true.

Suppose, in addition, $n \geq l \cdot n^*$, then by definition of f, it is also clear that

$$2C\sigma^2\sqrt{\frac{\log d}{n}} + \frac{b^2}{n} + \frac{2b}{\sqrt{n}} \cdot \left(\sqrt{2C\sigma^2} + \sqrt{\max_{i\in[d]} \Sigma_{ii}}\right) \leq \frac{\lambda_1 - \lambda_2}{8\sqrt{lk}} \cdot a^2.$$

□

Combining everything we have developed so far, we are ready to prove Theorem 8:

Proof of Theorem 8. Define $f(C, \Sigma, \sigma, d, n)$ to be the RHS of (9). By Theorem 16 and Theorem 18, we obtain that

$$\begin{aligned}\left\|W^* - vv^\top\right\|_\infty \leq \|W^* - vv^\top\|_F &\leq \frac{2k}{\lambda_1 - \lambda_2} \cdot \frac{1}{n} \|E\|_\infty + \frac{1}{n}\sqrt{\frac{2k}{\lambda_1 - \lambda_2} \cdot \|E\|_\infty} \\ &\leq \frac{2kf(C, \Sigma, \sigma, d, n)}{\lambda_1 - \lambda_2} + \frac{1}{\sqrt{n}}\sqrt{\frac{2kf(C, \Sigma, \sigma, d, n)}{\lambda_1 - \lambda_2}}\end{aligned}$$

holds with probability at least $1 - d^{-10}$.

Next, we show that there exists an absolute constant $C^* > 0$ such that when

$$n \geq n^* := \max\left\{C^* \cdot \left[\frac{k^2\sigma^4 \log d + b^2k^2\left(\sigma^2 + \max \Sigma_{ii}\right)}{(\lambda_1 - \lambda_2)^2 a^4} + \frac{kb^2}{(\lambda_1 - \lambda_2)a^2}\right], \frac{4}{a^2}, \log d\right\},$$

we have that

$$\frac{2kf(C, \Sigma, \sigma, d, n)}{\lambda_1 - \lambda_2} + \frac{1}{\sqrt{n}}\sqrt{\frac{2kf(C, \Sigma, \sigma, d, n)}{\lambda_1 - \lambda_2}} \leq \frac{a^2}{2}.$$

Note that the above inequality is implied by the fact that

$$\sqrt{\frac{2kf(C, \Sigma, \sigma, d, n)}{\lambda_1 - \lambda_2}} \leq \frac{-\frac{1}{\sqrt{n}} + \sqrt{\frac{1}{n} + 2a^2}}{2} = \frac{a^2}{\sqrt{\frac{1}{n} + 2a^2} + \frac{1}{\sqrt{n}}}.$$

Then it is clear that when $n \geq 4/(a^2)$, one has that $\frac{a^2}{\sqrt{\frac{1}{n}+2a^2}+\frac{1}{\sqrt{n}}} \geq \frac{a^2}{2a} = \frac{a}{2}$. Therefore, it suffices to show that for some n large enough

$$\sqrt{\frac{2kf(C, \Sigma, \sigma, d, n)}{\lambda_1 - \lambda_2}} \leq \frac{a}{2} \quad \Longleftrightarrow \quad \frac{2kf(C, \Sigma, \sigma, d, n)}{\lambda_1 - \lambda_2} \leq \frac{a^2}{4} \Longleftrightarrow \quad f \leq \frac{\lambda_1 - \lambda_2}{8k} \cdot a^2,$$

which is then implied by Theorem 19. □

Finally, we present the proof of Theorem 9. The high level idea in the proof is to make use of the fact that W^* is close to $vv^\top$ as developed in Theorem 8, and hence there is a simple way to control the quality of the solution we obtained in our algorithm.

Proof of Theorem 9. In the proof, we will constantly use the following fact:

Fact 2. *Suppose $x \in \mathbb{R}^d$ is a unit k-sparse vector, then for a $d \times d$ matrix E, one has $\left|x^\top Ex\right| \leq k\|E\|_\infty$.*

Fact 2 can be shown via Holder's inequality and the fact that $\left\|xx^\top\right\|_1 = \|x\|_1^2 \leq k\|x\|_2^2 = 1$.

We write $A = n\Sigma + \left(B^\top B - n\Sigma\right) + \left(M^\top M + M^\top B + B^\top M\right) =: n\Sigma + E$. It is clear that

$$\frac{1}{n}(x^*)^\top Ax^* \geq \max_{\substack{\|x\|_2=1 \\ \|x\|_0\leq k}} x^\top\Sigma x - \max_{\substack{\|x\|_2=1 \\ \|x\|_0\leq k}} x^\top\left(\frac{1}{n}E\right)x = \lambda_1 - \max_{\substack{\|x\|_2=1 \\ \|x\|_0\leq k}} x^\top\left(\frac{1}{n}E\right)x \geq \lambda_1 - \frac{k}{n}\|E\|_\infty,$$

where the first inequality uses the basic proposition of max function, and the second inequality follows from Fact 2. Similarly, one can obtain that

$$\frac{1}{n}(x^*)^\top Ax^* \leq \lambda_1 + \frac{k}{n}\left\|E\right\|_\infty. \tag{14}$$

We denote $\bar{x}$ the heuristic solution obtained via lines 1 - 2 in Algorithm 2. By the definition of $\bar{x}$, and the fact that $\bar{x}$ shares the same support with v by Theorem 8 with high probability by Theorem 8, one can see that $\frac{1}{n}\bar{x}^\top A\bar{x} \geq \frac{1}{n}v^\top Av = v^\top \Sigma v + \frac{1}{n}v^\top Ev \geq \lambda_1 - \frac{k}{n}\left\|E\right\|_\infty$ where we use Fact 2 in the last inequality. Summing everything up, we see that

$$\frac{(x^*)^\top Ax^* - \bar{x}^\top A\bar{x}}{n} \leq \frac{1}{n}(x^*)^\top Ax^* - \lambda_1 + \frac{k}{n}\left\|E\right\|_\infty \leq \frac{1}{n}\operatorname{tr}(AW^*) - \lambda_1 + \frac{k}{n}\left\|E\right\|_\infty \leq \frac{2k}{n}\left\|E\right\|_\infty. \tag{15}$$

Define $\epsilon := \frac{1}{8\sqrt{l}-1}$. Next, we show that

$$\frac{\lambda_1 - \lambda_2}{8\sqrt{l}k}\cdot a^2 \leq \frac{\lambda_1}{k}\cdot\frac{\epsilon}{2+\epsilon}. \tag{16}$$

In fact, (16) is equivalent to $2(\lambda_1 - \lambda_2)a^2 \leq \left[8\sqrt{l}\lambda_1 - (\lambda_1 - \lambda_2)a^2\right]\cdot\epsilon$. Since $0 < \lambda_1 - \lambda_2 \leq \lambda_1$, $a \leq 1$, we see that $\frac{2(\lambda_1-\lambda_2)a^2}{8\sqrt{l}\lambda_1 - (\lambda_1-\lambda_2)a^2} \leq \frac{2\lambda_1}{8\sqrt{l}\lambda_1 - \lambda_1} = \epsilon$ as desired.

Finally, combining (14), one obtains that with probability at least $1 - d^{-10}$,

$$\frac{2k}{n}\left\|E\right\|_\infty \leq \epsilon\left(\lambda_1 - \frac{k}{n}\left\|E\right\|_\infty\right) \leq \epsilon\cdot\frac{1}{n}\cdot(x^*)^\top Ax^*, \tag{17}$$

where the first inequality is implied by Theorem 18, Theorem 19, and (16): $\frac{1}{n}\left\|E\right\|_\infty \leq \frac{\lambda_1-\lambda_2}{8\sqrt{l}k}\cdot a^2 \leq \frac{\lambda_1}{k}\cdot\frac{\epsilon}{2+\epsilon}$. The desired approximation ratio is then given by (15) and (17). □

8 Acknowledgements

A. Del Pia and D. Zhou are partially funded by AFOSR grant FA9550-23-1-0433. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the Air Force Office of Scientific Research. The authors would love to thank Agniva Chowdhury and Lijun Ding for the discussions on algorithms to solve SPCA-SDP.

References

[1] Arash A Amini and Martin J Wainwright. High-dimensional analysis of semidefinite relaxations for sparse principal components. In *IEEE International Symposium on Information Theory*, pages 2454–2458, 2008.

[2] MOSEK ApS. *The MOSEK optimization toolbox for MATLAB manual. Version 9.2.*, 2020.

[3] Megasthenis Asteris, Dimitris Papailiopoulos, Anastasios Kyrillidis, and Alexandros G Dimakis. Sparse PCA via bipartite matchings. *Advances in Neural Information Processing Systems*, 28, 2015.

[4] Lauren Berk and Dimitris Bertsimas. Certifiably optimal sparse principal component analysis. *Mathematical Programming Computation*, 11:381–420, 2019.

[5] Quentin Berthet and Philippe Rigollet. Computational lower bounds for sparse PCA. *arXiv preprint arXiv:1304.0828*, 2013.

[6] Quentin Berthet and Philippe Rigollet. Optimal detection of sparse principal components in high dimension. 2013.

[7] Dimitris Bertsimas, Ryan Cory-Wright, and Jean Pauphilet. Solving large-scale sparse PCA to certifiable (near) optimality. *J. Mach. Learn. Res.*, 23(13):1–35, 2022.

[8] Dimitris Bertsimas and Driss Lahlou Kitane. Sparse PCA: A geometric approach. *Journal of Machine Learning Research*, 23:1–33, 2022.

[9] Samuel Burer and Renato DC Monteiro. A nonlinear programming algorithm for solving semidefinite programs via low-rank factorization. *Mathematical Programming*, 95(2):329–357, 2003.

[10] Samuel Burer and Renato DC Monteiro. Local minima and convergence in low-rank semidefinite programming. *Mathematical programming*, 103(3):427–444, 2005.

[11] Siu On Chan, Dimitris Papailiopoulos, and Aviad Rubinstein. On the worst-case approximability of sparse PCA. *arXiv preprint arXiv:1507.05950*, 2015.

[12] Siu On Chan, Dimitris Papailliopoulos, and Aviad Rubinstein. On the approximability of sparse PCA. In *Conference on Learning Theory*, pages 623–646. PMLR, 2016.

[13] Agniva Chowdhury, Petros Drineas, David P Woodruff, and Samson Zhou. Approximation algorithms for sparse principal component analysis. *arXiv preprint arXiv:2006.12748*, 2020.

[14] Diego Cifuentes. On the burer–monteiro method for general semidefinite programs. *Optimization Letters*, 15(6):2299–2309, 2021.

[15] Alexandre d'Aspremont, Laurent Ghaoui, Michael Jordan, and Gert Lanckriet. A direct formulation for sparse PCA using semidefinite programming. *Advances in neural information processing systems*, 17, 2004.

[16] Alberto Del Pia. Sparse PCA on fixed-rank matrices. *Mathematical Programming*, pages 1–19, 2022.

[17] Alberto Del Pia, Dekun Zhou, and Yinglun Zhu. Efficient sparse PCA via block-diagonalization. In *The Thirteenth International Conference on Learning Representations*, 2025.

[18] Yash Deshpande and Andrea Montanari. Sparse PCA via covariance thresholding. *Advances in Neural Information Processing Systems*, 27, 2014.

[19] Santanu S Dey, Rahul Mazumder, Marco Molinaro, and Guanyi Wang. Sparse principal component analysis and its l_1-relaxation. *arXiv preprint arXiv:1712.00800*, 2017.

[20] Santanu S Dey, Rahul Mazumder, and Guanyi Wang. Using ℓ_1-relaxation and integer programming to obtain dual bounds for sparse PCA. *Operations Research*, 70(3):1914–1932, 2022.

[21] Santanu S Dey, Marco Molinaro, and Guanyi Wang. Solving sparse principal component analysis with global support. *Mathematical Programming*, pages 1–39, 2022.

[22] 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.

[23] 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.

[24] Uffe Haagerup. The best constants in the khintchine inequality. *Studia Mathematica*, 70(3):231–283, 1981.

[25] Roger A Horn and Charles R Johnson. Matrix analysis, 1985.

[26] Ian T Jolliffe. *Principal component analysis for special types of data.* Springer, 2002.

[27] Jinhak Kim, Mohit Tawarmalani, and Jean-Philippe P Richard. Convexification of permutation-invariant sets and an application to sparse PCA. *arXiv preprint arXiv:1910.02573*, 2019.

[28] Robert Krauthgamer, Boaz Nadler, and Dan Vilenchik. Do semidefinite relaxations solve sparse PCA up to the information limit? *The Annals of Statistics*, 43(3):1300–1322, 2015.

[29] Harold W Kuhn and Albert W Tucker. Nonlinear programming. In *Traces and emergence of nonlinear programming*, pages 247–258. Springer, 2014.

[30] Monique Laurent and Franz Rendl. Semidefinite programming and integer programming. In K. Aardal, G. Nemhauser, and R. Weismantel, editors, *Handbook on Discrete Optimization*, pages 393–514. Elsevier B.V., December 2005.

[31] Seokho Lee, Michael P Epstein, Richard Duncan, and Xihong Lin. Sparse principal component analysis for identifying ancestry-informative markers in genome-wide association studies. *Genetic epidemiology*, 36(4):293–302, 2012.

[32] Yongchun Li and Weijun Xie. Exact and approximation algorithms for sparse PCA. *arXiv preprint arXiv:2008.12438*, 2020.

[33] Malik Magdon-Ismail. Np-hardness and inapproximability of sparse PCA. *Information Processing Letters*, 126:35–38, 2017.

[34] Michael Mitzenmacher and Eli Upfal. *Probability and computing: Randomization and probabilistic techniques in algorithms and data analysis.* Cambridge university press, 2017.

[35] Nikhil Naikal, Allen Y Yang, and S Shankar Sastry. Informative feature selection for object recognition via sparse PCA. In *2011 International Conference on Computer Vision*, pages 818–825. IEEE, 2011.

[36] Brendan O'Donoghue, Eric Chu, Neal Parikh, and Stephen Boyd. SCS: Splitting conic solver, version 3.2.4. `https://github.com/cvxgrp/scs`, November 2023.

[37] Dimitris Papailiopoulos, Alexandros Dimakis, and Stavros Korokythakis. Sparse PCA through low-rank approximations. In *International Conference on Machine Learning*, pages 747–755. PMLR, 2013.

[38] Clément W Royer, Michael O'Neill, and Stephen J Wright. A newton-cg algorithm with complexity guarantees for smooth unconstrained optimization. *Mathematical Programming*, 180:451–488, 2020.

[39] Lieven Vandenberghe and Stephen Boyd. Semidefinite programming. *SIAM review*, 38(1):49–95, 1996.

[40] Roman Vershynin. *High-dimensional probability: An introduction with applications in data science*, volume 47. Cambridge university press, 2018.

[41] Vincent Q Vu, Juhee Cho, Jing Lei, and Karl Rohe. Fantope projection and selection: A near-optimal convex relaxation of sparse PCA. *Advances in neural information processing systems*, 26, 2013.

[42] Alex L Wang and Fatma Kılınç-Karzan. On the tightness of sdp relaxations of qcqps. *Mathematical Programming*, 193(1):33–73, 2022.

[43] Alex L Wang and Fatma Kılınç-Karzan. Accelerated first-order methods for a class of semidefinite programs. *Mathematical Programming*, 209(1):503–556, 2025.

[44] Tengyao Wang, Quentin Berthet, and Richard J Samworth. Statistical and computational trade-offs in estimation of sparse principal components. *The Annals of Statistics*, 44(5):1896–1930, 2016.

[45] Alp Yurtsever, Olivier Fercoq, and Volkan Cevher. A conditional-gradient-based augmented lagrangian framework. In *International Conference on Machine Learning*, pages 7272–7281. PMLR, 2019.

[46] Tong Zhang. Adaptive forward-backward greedy algorithm for learning sparse representations. *IEEE transactions on information theory*, 57(7):4689–4708, 2011.