DIFFERENTIALLY PRIVATE ALGORITHMS FOR PACKING AND COVERING SEMIDEFINITE PROGRAMS

Anonymous authors
Paper under double-blind review

ABSTRACT

We provide differentially private algorithms for solving packing and covering semidefinite programs (SDPs) under the privacy model where neighboring inputs differ by a single constraint. We give efficient polynomial-time algorithms for both packing and covering SDPs that achieve better utility than the state-of-the-art exponential-time algorithm of Song-Xue-Zhang (NeurIPS 2025), which is designed solely for covering SDPs. Our approach builds upon a variant of the dense Multiplicative Weights Update (MWU) framework, and leverages an oracle which computes approximate largest/smallest eigenvectors. The oracle is implemented via a differentially private power iteration, and may be of independent interest.

1 INTRODUCTION

Semidefinite programming is a fundamental primitive used in algorithm design, statistics, learning, and data analysis. While it has a long history of remarkable developments, modern computational challenges revealed many new aspects of this problem that are yet to be understood. In particular, many of its common applications are naturally data-dependent: covariance estimation Cheng et al. (2019b;a), experimental design Allen-Zhu et al. (2021); van Apeldoorn & Gilyén (2019), metric learning Xing et al. (2002); Lanckriet et al. (2004), clustering relaxations Peng & Wei (2007); Guédon & Vershynin (2016), spectral relaxations of combinatorial problems Lovász (1979); Goemans & Williamson (1995) define constraints that may involve sensitive data. This makes semidefinite programming a natural target for differential privacy. However, developments on privately solving SDPs have been relatively scarce, compared to their linear programming counterpart.

A particularly important class of semidefinite programs is given by packing and covering SDPs. In normalized form, the covering problem is defined as

$$\min_{X \succeq 0} \left\{ \mathrm{Tr}\left(X\right) : \langle A_i, X \rangle \geq 1, \text{ for all } i \in [N] \right\}, \tag{1}$$

while the packing problem is defined as

$$\max_{X \succeq 0} \left\{ \mathrm{Tr}\left(X\right) : \langle A_i, X \rangle \leq 1, \text{ for all } i \in [N] \right\}, \tag{2}$$

where the PSD matrices A_i are private records, $\mathrm{Tr}\left(X\right)$ is the trace of X and $\langle A_i, X \rangle \coloneqq \mathrm{Tr}\left(A_i X\right)$.

We focus on the high-sensitivity privacy model, where neighboring instances differ by the addition or deletion of one constraint matrix. Thus changing one individual can change an entire SDP constraint, rather than merely perturbing one scalar entry. This is the natural semidefinite analog of the constraint-private model for linear programs studied by Hsu-Roth-Roughgarden-Ullman Hsu et al. (2014). This high-sensitivity constraint privacy model imposes an unavoidable tension between feasibility and privacy – in fact, it is impossible to satisfy all constraints while maintaining privacy, even for 1-dimensional linear programs. In the context of private linear programming, the way to overcome this barrier, as proposed by Hsu et al. Hsu et al. (2014), is to satisfy all but a small number of private constraints. The central question becomes how to characterize the number of constraints that need to be dropped in order to guarantee privacy.

For linear programming, Hsu et al. Hsu et al. (2014) provided private algorithms based on the Dense Multiplicative Weights method Herbster & Warmuth (2001). These algorithms essentially follow the standard Plotkin-Shmoys-Tardos Plotkin et al. (1995) framework for approximately solving LPs,

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

with the fundamental difference that here the algorithm maintains a distribution over constraints that is not allowed to concentrate too much on any single private record. This density condition allows one to reduce the sensitivity of the calls to the linear optimization oracle made in each iteration, which ultimately leads to a tradeoff between privacy and the number of violated constraints.

Subsequent work improved the efficiency of the method, as well as the tradeoffs between privacy and the number of violated constraints Kaplan et al. (2025); Ene et al. (2025; 2026). However, all these results are inherent to linear programming and moving from LPs to SDPs introduces new significant challenges. In particular, while the oracle used in each iteration of the dense MWU method for linear programming can be implemented by searching over the finite standard basis, the search space for SDPs becomes infinite. For this reason, the only known result for this problem due to Song-Xue-Zhang Song et al. (2025) is exponential time algorithm, designed only for covering SDPs. It remains a challenging open question to find polynomial time algorithms for both packing and covering SDPs with the best possible utility bound.

In this work, we make progress towards this question by providing new polynomial time algorithms for both problems with improved bounds on the number of violated constraints. Our main contribution is a new polynomial time private algorithm for packing and covering SDPs with its guarantees stated in the following theorem:

Theorem 1.1. *Assume $0 < \epsilon \leq 1$, $0 < \delta \leq e^{-1}$, $0 < \beta < 1/2$, and $0 < \alpha < 1$. Suppose every input matrix satisfies $0 \preceq A_i \preceq \Lambda_{\max} I$. For utility, assume that the covering SDP is feasible or that the packing SDP is bounded, an optimum OPT is attained, and a fixed public target V satisfies*

$$\left(1 - \frac{\alpha}{20}\right) \mathsf{OPT} \leq V \leq \left(1 + \frac{\alpha}{20}\right) \mathsf{OPT}.$$

Algorithm 1 is (ϵ, δ)-differentially private. With probability at least $1-\beta$, its covering output satisfies

$$X \succeq 0, \qquad \mathrm{Tr}(X) \leq (1+\alpha)\mathsf{OPT}, \qquad \langle A_i, X \rangle \geq 1 - \alpha$$

for all but at most s constraints. Its packing output satisfies

$$X \succeq 0, \qquad \mathrm{Tr}(X) \geq (1-\alpha)\mathsf{OPT}, \qquad \langle A_i, X \rangle \leq 1 + \alpha$$

for all but at most s constraints. Moreover, using Algorithm 2 to implement the oracles, the exception budget s can be set to satisfy

$$s \leq \min\left\{ N, \widetilde{O}\left(\frac{\sqrt{d}\ (\Lambda_{\max} V)^{3/2} \sqrt{\log N}}{\alpha^2 \epsilon} \right) \right\},$$

where the suppressed factors in the $\widetilde{O}$-notation are logarithmic in the parameters.

Remark 1.1. Theorem 1.1 assumes access to public values $\Lambda_{\max}$ and V. This is a common assumption in prior work such as Hsu et al. (2014); Ene et al. (2026) (for private LP) and Song et al. (2025) (for private SDP). These values are only required to obtain the guarantee on the number of violated constraints s. For any fixed values of $\Lambda_{\max}$ and V, Algorithm 1 remains private while the utility is guaranteed for all SDPs that satisfy the assumption. One can bypass this assumption by running the algorithm for all guesses that are multiples of $(1 + \alpha/20)$ over the ranges of OPT and $\max_{i \in [N]} \|A_i\|_2$ and using the exponential mechanism to select the best results.

Comparison with prior work Song et al. (2025). Theorem 1.1 improves the result of Song et al. (2025) in two aspects: runtime and utility guarantee. First, our algorithm achieves a polynomial runtime, as opposed to the exponential-time algorithm of Song et al. (2025). While both works rely on the Dense MWU framework, we design a new oracle implementation that bypasses the naive γ-net argument (search by discretizing the space) used in the prior work. It should be highlighted that while it is straightforward to improve the running time of the algorithm—indeed, we also show a simple implementation of the oracle using the Gaussian mechanism in the appendix—improving the utility bound is significantly more challenging. To achieve this, we introduce a more sophisticated private power method to implement the oracle. This method operates in a privacy model where adjacent entries have their difference bounded in spectral norm, as opposed to the more standard notion of adjacency which bounds the Frobenius norm. While this adjacency model for private spectral problems has been considered previously Kapralov & Talwar (2013); Hardt & Price (2014), here we provide a gap-independent method. More precisely, we return a unit vector u

satisfying an additive guarantee $u^\top Au \geq \lambda_{\max}(A) - \widetilde{O}\left(\frac{\sqrt{d}S}{\varepsilon}\right)$ without assuming any separation between the leading eigenvalues. This type of guarantee fits the dense MWU framework, and leads to an improved bound on the number of violated constraints, improving from $\tilde{O}\left(\frac{d(\Lambda_{\max}V)^2}{\alpha^2\epsilon}\right)$ to $\widetilde{O}\left(\frac{\sqrt{d}(\Lambda_{\max}V)^{3/2}\sqrt{\log N}}{\alpha^2\epsilon}\right)$.

1.1 Our technique

We leverage the dense Multiplicative Weights Update framework developed in prior work by Herbster & Warmuth (2001); Hsu et al. (2014); Ene et al. (2026) to reduce solving the problem to the sequence of self-contained subproblems. These subproblems involve privately approximating the top eigenvector (for covering SDPs) and bottom eigenvector (for packing SDPs) of a matrix: given a matrix $\overline{A}$, we need to find vectors $u_{\max}$ and $u_{\min}$ such that $u_{\max}^\top \overline{A} u_{\max} \geq \lambda_{\max}\left(\overline{A}\right) - q$ and $u_{\min}^\top \overline{A} u_{\min} \leq \lambda_{\min}\left(\overline{A}\right) + q$. We obtain an efficient polynomial time algorithm to solve these subproblems (oracle implementation) by introducing a new private power method with gap-independent guarantees and a new baseline approach using the Gaussian mechanism (provided Sec. H).

To compute approximate the top eigenvector of a matrix $\overline{A}$ without knowledge of any structural information about $\overline{A}$, we need an algorithm that does not depend on information about the spectrum such as the gap between the eigenvalues of the input matrix. Among the large literature on privately approximating the top eigenvector, most results depend on additional information about the spectrum, with the rare exception of the Gaussian mechanism Dwork et al. (2014). In the first approach introduced as a baseline for comparison, we use this Gaussian mechanism, which perturbs the input matrix with a symmetric matrix where the elements on the diagonal and above are i.i.d. Gaussians Dwork et al. (2014); Tran & Vishnoi (2025). This Gaussian perturbation provides an error bound that is independent of the gap between eigenvalues of the input matrix and ultimately yields an (ε, δ)-private algorithm where the number of unsatisfied constraints scales linearly with the matrix dimension. This approach is conceptually simple and already gives a polynomial-time algorithm. However, it yields a suboptimal dimension dependence, caused by the fact that while input matrices are bounded in spectral norm, the sensitivity of the Gaussian mechanism scales with the Frobenius norm which can be much larger.

We improve the state of affairs by designing a new algorithm based on a private power iteration that reduces sensitivity by perturbing a d-dimensional unit vector in each step. The private power method performs only noisy matrix-vector multiplications,

$$x_{t+1} = Bx_t + \beta \cdot g_t, \qquad x_{t+1} \leftarrow x_{t+1} / \left\|x_{t+1}\right\|_2,$$

where B has the same eigenvectors as $\overline{A}$ but is rescaled so that the spectrum of B is between $1/2$ and 1, g_t is a standard Gaussian vector with i.i.d. entries, and β is a scalar noise parameter determined by the sensitivity now can be upper bounded in the spectral norm of the sensitivity of the matrix $\overline{A}$. While the private power iteration is a widely studied algorithmic template, its guarantees typically require additional information about the spectrum, which is not apt for guaranteeing privacy when combined with an iterative algorithm such as dense MWU. To the best of our knowledge, only Balcan et al. (2016) obtained gap-independent results. However, their results are for a different privacy model, their solution is a high rank matrix instead of a vector, and they approximately minimize the residual error as opposed to maximizing $u^\top \overline{A} u$, which is what we need.

Obtaining a gap-independent algorithm poses a significant challenge. Gap-dependent algorithms often rely on a clear separation between the eigenvalues of $\overline{A}$ to argue about the faster growth of the components of x_t along the top eigenvector compared with the remaining. When there is no such clear separation, the noise vector g_t can slow down the growth of the top components while amplifying the growth of the bottom components. This leads to a slow convergence rate or even non-convergence. However, as long as an iterate with small approximation error exists, we can use the Exponential mechanism to find one. Our gap-independent analysis shows the existence of such an iterate and is based on the following idea: the supermartingale property of the iterate sequence maintains that the correlation between the iterates and the top eigenvector remains non-trivial over time, which allows us to bound the cumulative approximation error. Our power method leverages the reduced sensitivity of computing the solution in each iteration by a factor $\sqrt{d}$ to achieve the $\sqrt{d}$ improvement in the utility, compared with the baseline approach and Song et al. (2025)'s algorithm.

As a final remark, our approach builds on the variant of dense MWU developed in the recent work of Ene et al. (2026), which uses the special structure of the input for packing/covering problems. This algorithm requires fewer iterations than that used in Song et al. (2025), improving the dependence on the problem width characterized by $\Lambda_{\max}\mathsf{OPT}$ in prior work.

1.2 Related Work

We review the most relevant literature to our work and provide additional related work in Sec. A.

Private semidefinite and symmetric-cone programming.The recent work of Song et al. (2025) studies differential privacy for optimization over Euclidean Jordan algebras, which lead to a very general framework that includes linear programming, second-order cone programming, and semidefinite programming. Their results yield private algorithms for SDPs in both high-sensitivity and low-sensitivity regimes. The main difference between their work and ours is runtime efficiency and the dependence on the dimensions in the number of violated constraints. For SDPs, their generic oracle must privately optimize a quadratic form over rank-one directions. Their implementation discretizes the relevant set of directions and applies the exponential mechanism over this discretization. This gives a broadly applicable private oracle, but it leads to exponential dependence on the matrix dimension in the runtime.

Private linear programming and high-sensitivity constraint privacy. The study of differentially private algorithms for linear programming was initiated by Hsu et al. (2014). They considered several natural privacy models for LPs, including the high-sensitivity constraint-private model in which neighboring instances differ by the addition or deletion of one arbitrary constraint. In this model, changing one individual's data may change an entire constraint. They showed that, in general, privacy is incompatible with satisfying all constraints, and therefore introduced the guarantee of satisfying all but a small number of constraints. Their results are based on the Dense Multiplicative Weights algorithm Herbster & Warmuth (2001), where the distribution over constraints is restricted so that no single private constraint can receive too much weight, which allows them to reduce constraint sensitivity. Several recent works have improved the algorithmic and quantitative guarantees for private LPs and related linear-algebraic problems. Kaplan et al. (2025) gave efficient private algorithms for linear-algebraic tasks including solving systems of linear inequalities. Ene et al. (2025) gave improved algorithms for privately solving linear programs, and more recently studied positive LPs, including packing, covering, and mixed packing-covering formulations, in the same high-sensitivity constraint-private regime Ene et al. (2026).

2 Preliminaries

Notation. For a symmetric matrix A, let $\lambda_{\max}(A)$ and $\lambda_{\min}(A)$ denote the biggest and smallest eigenvalues of A. Let $\mathrm{Tr}\,(A)$ denote the trace of a square matrix A. For two symmetric matrices A and B of dimension $d \times d$, let $\langle A, B\rangle = \mathrm{Tr}\,(AB)$. Let $\mathcal{A} = \{A_1, \ldots, A_N\}$ be a collection of N matrices of dimension $d \times d$ and define $\langle \mathcal{A}, X\rangle$ be the vector whose i-th coordinate is $\langle A_i, X\rangle$ for a matrix $X \in \mathbb{R}^{d\times d}$. Let OPT denote the optimal objective of the SDP.

Differential Privacy. Two inputs $\mathcal{A}$ and $\mathcal{A}'$ to the problem are neighbors when the larger is obtained from the smaller by adding one matrix. A randomized algorithm $\mathcal{M}$ is said to be (ϵ, δ)-differentially private (DP) if for all neighboring inputs $\mathcal{A}$ and $\mathcal{A}'$ and every measurable set of possible outcomes $\mathcal{O}$,

$$\Pr\left[\mathcal{M}(\mathcal{A}) \in \mathcal{O}\right] \leq e^{\epsilon} \Pr\left[\mathcal{M}(\mathcal{A}') \in \mathcal{O}\right] + \delta.$$

In the case $\delta = 0$, we say the algorithm is ϵ-DP.

We assume that a public number $\Lambda_{\max} > 0$ satisfies $0 \preceq A_i \preceq \Lambda_{\max} I$ for every input matrix. Our algorithm also takes a fixed public target value $V > 0$ that is chosen independently of the private input. The privacy guarantee of the algorithm holds for every fixed choice of V, while for the utility guarantee we assume V as an approximation of OPT.

We will use two standard tools in differential privacy: the Gaussian Dwork et al. (2006) and the exponential mechanisms McSherry & Talwar (2007). Details about these two mechanisms are provided in the appendix.

Algorithm 1 Dense MWU with an Additive Spectral Oracle

1: **Input**: PSD matrices $A_1, \ldots, A_N$, public parameters $N, \Lambda_{\max}, V, \alpha$, exception budget $1 \le s \le N$, mode $b \in \{\mathsf{cover}, \mathsf{pack}\}$, and additive spectral oracles $\mathcal{O}_{\max}$ for $b = \mathsf{cover}$ and $\mathcal{O}_{\min}$ for $b = \mathsf{pack}$
2: Let $T = \lceil 20\Lambda_{\max} V \log N/\alpha^2 \rceil$, $\eta = \alpha/(10\Lambda_{\max} V)$, $U = 1/s$, and $X_0 = 0$.
3: **for** $t = 0, \ldots, T-1$:
4: $\quad p_t = \nabla \mathrm{smin}^U(\eta\langle \mathcal{A}, X_t\rangle)$ if $b = \mathsf{cover}$, and $p_t = \nabla \mathrm{smax}^U(\eta\langle \mathcal{A}, X_t\rangle)$ otherwise.
5: $\quad \overline{A}_t = \sum_{i=1}^N p_{t,i} A_i$.
6: $\quad u_t = \mathcal{O}_{\max}(\overline{A}_t)$ if $b = \mathsf{cover}$, and $u_t = \mathcal{O}_{\min}(\overline{A}_t)$ otherwise.
7: $\quad X_{t+1} = X_t + V u_t u_t^\top$.
8: **Output** $\overline{X} = X_T/T$.

Truncated softmax and softmin functions. Following the work of Ene et al. (2026), we use the truncated softmax and softmin functions, defined as follows. Let $\Delta_n = \left\{ r \in \mathbb{R}^n_{\ge 0} : \sum_{i\in[n]} r_i = 1 \right\}$ be the unit simplex. For $r \in \Delta_n$, let $\omega(r) = \sum_{i\in[n]} r_i \log r_i$ be the negative entropy, with the convention $0\log 0 = 0$. Given a scalar parameter $1/n \le U \le 1$, define $\mathcal{D}^U = \left\{ r \in \Delta_n : \max_{i\in[n]} r_i \le U \right\}$. We define the truncated softmax function smax^U and the truncated softmin function smin^U as:

$$\mathrm{smax}^U(x) = \max_{r\in\mathcal{D}^U} \langle x, r\rangle - \omega(r),$$

$$\text{and } \mathrm{smin}^U(x) = -\mathrm{smax}^U(-x) = \min_{r\in\mathcal{D}^U} \langle x, r\rangle + \omega(r).$$

Our algorithm utilizes the gradient of these functions, $\nabla\mathrm{smax}^U$ and $\nabla\mathrm{smin}^U$. Their properties are provided in the appendix.

3 Dense MWU with an Additive Spectral Oracle

In this section, we provide a general framework to approximately solve packing and covering with differential privacy. Before introducing the algorithm, we give the definition of the additive spectral oracles, which our algorithm makes use of:

Definition 3.1 (Spectral Oracles)**.** Given a symmetric matrix M and an additive error $q \ge 0$, a **top-eigenvalue oracle** returns a unit vector u such that

$$u^\top M u \ge \lambda_{\max}(M) - q,$$

whereas a **bottom-eigenvalue oracle** returns a unit vector satisfying

$$u^\top M u \le \lambda_{\min}(M) + q.$$

We present the general dense MWU framework in Algorithm 1. The algorithm follows a Dense MWU template generalized for SDPs. In each iteration, Algorithm 1 makes call a top-eigenvalue oracle $\mathcal{O}_{\max}$ (when solving a covering SDP) or a bottom-eigenvalue oracle (when solving a packing SDP) and updates the weights for the constraints based on the oracle outputs. The oracles may be randomized and their input may depend adaptively on all previous outputs. For privacy accounting, in iteration t, the oracle call releases u_t based on the private collection of constraints and all previously released vectors $u_1, \ldots, u_{t-1}$; the intermediate quantities p_t and $\overline{A}_t$ are internal and are not released.

The following theorem shows the oracle-to-SDP interface used by all subsequent instantiations. It requires only an additive approximation guarantee of the oracle and does not depend on how the oracle is implemented. Moreover, if the oracle is differentially private, Algorithm 1 is also differentially private due to composition over T iterations. Theorem 3.1 therefore separates the approximation guarantee and the construction of a private oracle, which we study in the next section.

Theorem 3.1. *Assume $0 \preceq A_i \preceq \Lambda_{\max} I$ for every $i \in [N]$, and suppose the relevant covering SDP is feasible or the relevant packing SDP is bounded. In either case, let X^* be an attained optimum*

and write $\mathsf{OPT} = \mathrm{Tr}(X^*)$. *Fix* $\alpha \in (0,1)$, $1 \le s \le N-1$, *and a public target* V *satisfying*

$$\left(1 - \frac{\alpha}{20}\right)\mathsf{OPT} \le V \le \left(1 + \frac{\alpha}{20}\right)\mathsf{OPT}.$$

Run Algorithm 1. Suppose that, conditioned on previous iterations, each oracle call returns a unit vector satisfying its additive guarantee with error at most q *and conditional failure probability at most* β_0. *If* $qV \le \frac{\alpha}{20}$, *then, with probability at least* $1 - T\beta_0$, *the following statements hold:*

For a covering problem, $\overline{X} \succeq 0$, $\mathrm{Tr}(\overline{X}) = V \le (1+\alpha)\mathsf{OPT}$ *and* $\langle A_i, \overline{X}\rangle \ge 1 - \alpha$ *for all but at most* s *constraints. For a packing problem,* $\overline{X} \succeq 0$, $\mathrm{Tr}(\overline{X}) = V \ge (1-\alpha)\mathsf{OPT}$ *and* $\langle A_i, \overline{X}\rangle \le 1 + \alpha$ *for all but at most* s *constraints. Moreover, if conditioned on previous iterations, each oracle call is* (ϵ_0, δ_0)*-differentially private, then Algorithm 1 is* (ϵ, δ)*-differentially private where* $\epsilon = \sqrt{2T\log(1/\delta')}\,\epsilon_0 + T\epsilon_0(e^{\epsilon_0} - 1)$ *and* $\delta = T\delta_0 + \delta'$ *for every* $\delta' \in (0,1)$.

Proof Sketch. The proof of this theorem is a generalization of Ene et al. (2026)'s proof for the case of Linear Programming. We give a proof sketch for the covering case. The complete proof is deferred to the appendix. For covering, we use the properties of the truncated-softmin function and the inequality in Lemma E.1 implies,

$$\begin{aligned}
\mathrm{smin}^U\left(\eta\left\langle \mathcal{A}, X_{t+1}\right\rangle\right) &= \mathrm{smin}^U\left(\eta\left\langle \mathcal{A}, X_t + V u_t u_t^\top\right\rangle\right) \\
&\ge \mathrm{smin}^U\left(\eta\left\langle \mathcal{A}, X_t\right\rangle\right) \\
&\quad + \left(1 - \left\|\eta\left\langle \mathcal{A}, V u_t u_t^\top\right\rangle\right\|_\infty\right)\left\langle \nabla \mathrm{smin}^U\left(\eta\left\langle \mathcal{A}, X_t\right\rangle\right), \eta\left\langle \mathcal{A}, V u_t u_t^\top\right\rangle\right\rangle \\
&\ge \mathrm{smin}^U\left(\eta\left\langle \mathcal{A}, X_t\right\rangle\right) + \left(1 - \frac{\alpha}{10}\right)\left\langle p_t, \eta\langle \mathcal{A}, V u_t u_t^\top\right\rangle\right\rangle \\
&= \mathrm{smin}^U\left(\eta\left\langle \mathcal{A}, X_t\right\rangle\right) + \left(1 - \frac{\alpha}{10}\right)^2. \qquad (3)
\end{aligned}$$

where we have use the property of the output u_t of the $\mathcal{O}_{\max}$ oracle that $\langle \mathcal{A}, V u_t u_t^\top\rangle \ge V\left(\lambda_{\max}\left(\overline{A}_t\right) - q\right) \ge 1 - \frac{\alpha}{10}$ and $\|\eta\langle \mathcal{A}, V u_t u_t^\top\rangle\|_\infty \le \eta\Lambda_{\max} V = \frac{\alpha}{10}$. We can unroll the inequality in (3) and obtain,

$$\begin{aligned}
\mathrm{smin}^U(\eta\langle \mathcal{A}, X_T\rangle) &\ge \mathrm{smin}^U(0) + \eta T\left(1 - \frac{\alpha}{10}\right)^2 \\
&\ge -\log N + \eta T\left(1 - \frac{\alpha}{10}\right)^2 \ge \eta T(1-\alpha),
\end{aligned}$$

where the last inequality uses $\eta T \ge 2\log N/\alpha$. Notice that $\mathrm{smin}^U(\eta\langle \mathcal{A}, X_T\rangle)$ provides a lower bound on the smallest s entries of $\eta\langle \mathcal{A}, X_T\rangle$, from which we can conclude that the number of violated constraints is at most s. □

4 Private Oracles via Private Power Iteration Method

4.1 Problem definition and privacy model

While Section 3 provides a general template to solve packing and covering SDPs with differential privacy, it leaves open the question of how to implement the additive oracles with differential privacy. We address this problem in this section. We start with the following lemma which states the sensitivity of the input to the oracles.

Lemma 4.1. *Let* $\mathcal{A} = \{A_1, \ldots, A_N\}$ *and* $\mathcal{A}' = \{A_1, \ldots, A_{N-1}\}$ *be two neighboring inputs of the SDP. In each iteration* t, *conditioned on all prior iterates* X_i *for all* $i < t$ *to be the same on* $\mathcal{A}$ *and* $\mathcal{A}'$, *the inputs to the oracles* $\overline{A}_t = \sum_{i=1}^N p_{t,i} A_i$ *and* $\overline{A}'_t = \sum_{i=1}^{N-1} p'_{t,i} A_i$ *satisfy: a)* $\left\|\overline{A}_t\right\|_2 \le \Lambda_{\max}$; *b)* $\left\|\overline{A}'_t\right\|_2 \le \Lambda_{\max}$ *and c)* $\left\|\overline{A}_t - \overline{A}'_t\right\|_2 \le \frac{2\Lambda_{\max}}{s}$.

Proof. Since $\|A_i\|_2 \le \Lambda_{\max}$ and $\overline{A}_t$ and $\overline{A}'_t$ are convex combinations of $\{A_1, \ldots, A_N\}$, we have immediately that $\left\|\overline{A}_t\right\|_2 \le \Lambda_{\max}$ and $\left\|\overline{A}'_t\right\|_2 \le \Lambda_{\max}$. The third claim also follows directly from Lemma E.2. □

Lemma 4.1 says that the input to the oracle A satisfies $\|A\|_2 \leq L$ for a public spectral bound $L = \Lambda_{\max} > 0$ and that neighboring inputs A, A' to the oracle satisfy $\|A - A'\|_2 \leq S$ for a sensitivity parameter $S = \frac{2\Lambda_{\max}}{s}$. To implement the oracles, the goal is to output unit vectors $u_{\max}$ and $u_{\min}$ such that

$$u_{\max}^\top A u_{\max} \geq \lambda_{\max}(A) - q, \qquad u_{\min}^\top A u_{\min} \leq \lambda_{\min}(A) + q.$$

Remark 4.1. Prior works such as Dwork et al. (2014) that privately approximate the top eigenvector of a matrix usually consider model the distance between two neighboring matrices A and A' to be bounded in the Frobenius norm, i.e. $\|A' - A\|_F \leq S$. Since $\|A' - A\|_F \leq \sqrt{d}\,\|A - A'\|_2$, one can reduce the spectral-norm model to the Frobenius-norm model and apply previous results (e.g. of Dwork et al. (2014)) to obtain an algorithm to implement the oracle. We provide such an algorithm in the appendix. This simple algorithm has polynomial runtime, which is a direct improvement to the exponential time algorithm by Song et al. (2025). However, due to the $\sqrt{d}$ factor that appears when changing the norm, the utility bound for this approach incurs a non-ideal extra $\sqrt{d}$ factor. We introduce a new algorithm to improve this factor next.

4.2 Algorithm

Algorithm 2 Two-Sided Additive Private Power Iteration

Input: symmetric matrix A, public spectral bound $L \geq \|A\|_2$, sensitivity S, and privacy and failure parameters ϵ, δ, β
Let $R = \lceil 8\log(8/\beta) \rceil$, $\ell = \log\left(e + \frac{16dRL\epsilon}{S\beta\delta}\right)$, and $q_0 = C_0 \frac{S}{\epsilon}\sqrt{dR\log\frac{8}{\delta}}\,\ell^{3/2}$,where C_0 is a sufficiently large universal constant
If $q_0 \geq 2L$, output two fixed, input-independent unit vectors and halt
Let $T = \max\{1, \lfloor c_0 L\ell/q_0 \rfloor\}$, $\Delta = S/(4L)$, and $G_{2RT} = 8\sqrt{2RT\log(1/\delta)}/\epsilon$, where $c_0 > 0$ is a sufficiently small universal constant
for $s \in \{+1, -1\}$:
    $B_s = (3LI + sA)/(4L)$
    Independently for each $r \in [R]$, let $x_{s,r,0} = z_{s,r}/\|z_{s,r}\|_2$ where $z_{s,r} \sim \mathcal{N}(0, I_d)$
    for $(r, t) \in [R] \times \{0, \ldots, T-1\}$:
        $y_{s,r,t+1} = B_s x_{s,r,t} + \Delta G_{2RT} g_{s,r,t}$, where $g_{s,r,t} \sim \mathcal{N}(0, I_d)$
        $x_{s,r,t+1} = y_{s,r,t+1}/\|y_{s,r,t+1}\|_2$
    From $\mathcal{C}_s = \{x_{s,r,t} : r \in [R], 0 \leq t \leq T\}$ sample x_s with probability proportional to $\exp\left(\frac{\epsilon x^\top B_s x}{8\Delta}\right)$.
Output $u_{\max} = x_{+1}$ and $u_{\min} = x_{-1}$

We now give a private power iteration algorithm to implement the additive spectral oracles. For the input matrix A with a public spectral bound $L \geq \|A\|_2$ and sensitivity S, the algorithm proceeds as follows. For simplicity, we omit the subscript r used in the algorithm. We define for $s \in \{+1, -1\}$,

$$B_s = \frac{3}{4} \cdot I + \frac{s}{4L} \cdot A \tag{4}$$

and initialize a solution $x_{s,0} = z_s/\|z_s\|_2$ where z_s is a $\mathbb{R}^d$ vector sampled from $\mathcal{N}(0, I_d)$. Here $s = +1$ is for approximating the top eigenvector while $s = -1$ is for the bottom singular vector. We then run the power method for T iterations, each of which uses the matrix B_s to update the iterate and calibrate the Gaussian noise added, before normalizing it.

$$y_{s,t+1} = B_s x_{s,t} + \text{Gaussian noise}; \qquad x_{s,t+1} = y_{s,t+1}/\|y_{s,t+1}\|_2.$$

After repeating this procedure R times to boost the success probability, the whole pool of output solutions is passed through the exponential mechanism to select the one with the lowest error. The complete algorithm is given in Figure 2.

Remark 4.2. Matrices B_s satisfy $\frac{1}{2}I \preceq B_s \preceq I$ and have spectral-norm sensitivity $\Delta = \frac{S}{4L}$ while preserving the eigenvector of A: the leading eigenvector of B_{+1} is a largest-eigenvalue vector of A, whereas the leading eigenvector of B_{-1} is a smallest-eigenvalue vector of A. The shift to make $B_s \succeq I/2$ is so that the leading eigenvalue of B_s is always within a constant factor of the maximum of the range, well above the noise from the Gaussian perturbation.

4.3 Analysis overview

We give an overview of the analysis of this algorithm. Letting B be any of B_{+1} and B_{-1}, we focus on one run of T iteration of the algorithm and show the following variance-sensitive form of the gap-independent power-iteration argument, which is the main ingredient in the additive analysis.

Lemma 4.2. *Let B satisfy $I/2 \preceq B \preceq I$, and consider the iterative updates*

$$x_{t+1} = \frac{Bx_t + \sigma g_t}{\|Bx_t + \sigma g_t\|_2}, \qquad x_0 = \frac{z}{\|z\|_2},$$

where $g_0, \dots, g_{T-1}, z$ are independent standard Gaussian vectors. There are universal constants $c > 0$ and $C > 0$ such that, if

$$\sigma^2 dT \le c, \tag{5}$$

then with probability at least $1/4$, one of the first T iterates satisfies

$$\min_{0 \le t < T} \left\{ \lambda_{\max}(B) - x_t^\top B x_t \right\} \le C \left(\frac{\log(ed)}{T} + \sigma^2 d \right) = \tilde{O}\left(\frac{1}{T} \right). \tag{6}$$

Consequently, for R independent executions, the probability that none of the R solutions satisfy (6) is at most $e^{-R/4}$.

Lemma 4.2 says that for each run of T iterations, with an appropriately set noise level σ, with a constant probability, there exists an iterate that approximates the top eigenvector of B with error $\tilde{O}\left(\frac{1}{T}\right)$, and we can boost the success probability and independent repetitions. Using this lemma and by setting the parameters as in Algorithm 2, we then show the privacy and utility of Algorithm 2 in the following theorem.

Theorem 4.1. *Assume $0 < \epsilon \le 1$ and $0 < \delta, \beta < 1/4$. Algorithm 2 is (ϵ, δ)-differentially private. With probability at least $1 - \beta$, its outputs satisfy*

$$u_{\max}^\top A u_{\max} \ge \lambda_{\max}(A) - q, \qquad u_{\min}^\top A u_{\min} \le \lambda_{\min}(A) + q,$$

where

$$q = \min \left\{ 2L,\ C \frac{S}{\epsilon} \sqrt{dR \log \frac{8}{\delta}}\, \ell^{3/2} \right\} = \widetilde{O}\left(\frac{\sqrt{d} S}{\epsilon} \right).$$

Here R and ℓ are the parameters in Algorithm 2; thus all suppressed factors are logarithmic in the problem and failure parameters. In particular, the bound is independent of the eigengaps of A.

Proof overview of Lemma 4.2. We define $\mu = \lambda_{\max}(B)$, set $D = B/\mu$ and $\eta = \sigma/\mu$, and consider the unnormalized process for convenience

$$w_0 = x_0, \qquad w_{t+1} = D w_t + \eta \|w_t\|_2 g_t, \qquad Q_t = \|w_t\|_2^2. \tag{7}$$

This unnormalized process has the same behavior as normalized one, i.e. $x_t = w_t / \|w_t\|_2$. It is by the standard argument that the initial random solution w_0 has a nontrivial correlation with the top eigenvector of B. The proof has two steps.

Step 1 (Claim G.1). We will first show that the Gaussian noise from the power iterations cannot degrade this correlation too much. Thus, the final solution still has a nontrivial correlation with the top eigenvector and thus $\|w_T\|_2$ is still nontrivial. To do this, let $a_t = v^\top w_t$ where v is a unit top eigenvector v of D. We can write

$$a_{t+1} = a_t + \eta \sqrt{Q_t} \cdot \xi_t, \qquad \text{where} \qquad \xi_t = v^\top g_t \sim \mathcal{N}(0, 1),$$

which means $(a_t)_{t \ge 0}$ is a martingale. Moreover, we can show that $Q_t/(1+\eta^2 d)^t$ is a nonnegative supermartingale. Using Doob's maximal inequality on (Q_t), we can bound the predictable quadratic variation through time of (a_t), and show with constant probability that $|a_T| = \Omega\left(1/\sqrt{d}\right)$. Thus $\|w_T\|_2^2 \ge a_T^2 = \Omega\left(1/d\right)$.

Step 2 (Claim G.2). We will show that the added noise to the power method is too small to significantly change the iterates' trajectory from the standard case without noise so eventually, there must

exist an iterate with the guaranteed error bound. To do this, we relates the above lower bound of $\|w_T\|_2$ with the error bound, by expressing

$$1 - x_t^\top D x_t \le m_t + \text{ noise } - \log \frac{Q_{t+1}}{Q_t},$$

where m_t is a martingale difference sequence. Both m_t and the noise term can be bounded by standard inequalities whereas the terms $-\log \frac{Q_{t+1}}{Q_t}$ form a telescoping sum and can be bounded using the bound from step 1 of the proof.

Proof overview of Theorem 4.1. To show the privacy guarantee, we show the sensitivity of the iterates, by bounding on neighboring inputs A and A', and correspondingly B_s and B_s'. Conditioned on the history of updates, the current normalized vector $x_{s,r,t}$ is fixed, and we can bound the sensitivity:

$$\|x_{s,r,t+1} - x'_{s,r,t+1}\|_2 = \|(B_s - B'_s)x_{s,r,t}\|_2 \le \|B_s - B'_s\|_2 \le \Delta.$$

Thus we can derive privacy bounds by using the Gaussian mechanism together with the composition theorem. To show the utility guarantee, we show that the noise level $\sigma = \Delta G_{2RT}$ and the number of iterations T satisfy the condition of Lemma 4.2. Finally, we conclude the proof by using the standard guarantee of the exponential mechanism, to select an approximately best iterate.

5 Proof of Theorem 1.1

Finally, we present the proof of Theorem 1.1.

Proof. The branch $s = N$ is input independent, hence private, and its feasibility assertion permits all N constraints as exceptions. Its trace is V, so the claimed objective inequality follows from the assumed relation between V and OPT. We therefore consider $s \le N - 1$.

Fix an outer iteration and condition on its preceding transcript. Lemma 4.1 gives $\|\overline{A}_t - \overline{A}'_t\|_2 \le \frac{2\Lambda_{\max}}{s} := S$ for a neighboring input $\mathcal{A}'$ or $\mathcal{A}$ and $0 \preceq \overline{A}_t \preceq \Lambda_{\max} I$. Set $T = \left\lceil \frac{20 W \log N}{\alpha^2} \right\rceil$ in Algorithm 1, $\epsilon_0 = \frac{\epsilon}{2\sqrt{2T\log(2/\delta)}}$, $\delta_0 = \frac{\delta}{2T}$, and $\beta_0 = \frac{\beta}{2T}$. Let $L = \Lambda_{\max}$, $R_0 = \left\lceil 8 \log \frac{8}{\beta_0} \right\rceil$, $H_0 = \log \frac{8}{\delta_0}$, $\overline{\ell} = \log\left(e + \frac{32 d R_0 N T^2}{\beta\delta}\right)$, $s_0 = \left\lceil C_1 \frac{W}{\alpha\epsilon_0} \sqrt{d R_0 H_0}\, \overline{\ell}^{3/2} \right\rceil$, $s = \min\{N, s_0\}$. If $s = N$, output the fixed matrix VI/d. Otherwise run Algorithm 1 with exception budget $s = s_0$. At every outer iteration, evoke Algorithm 2 as the oracle with parameter $(\epsilon_0, \delta_0, \beta_0, L, S)$. Apply Theorem B.2 with $\delta' = \delta/2$. The composed δ parameter is $T\delta_0 + \delta/2 = \delta$. Since $\epsilon_0 \le 1$ and $e^{\epsilon_0} - 1 \le 2\epsilon_0$, the composed ϵ parameter is at most

$$\sqrt{2T\log(2/\delta)}\,\epsilon_0 + T\epsilon_0(e^{\epsilon_0} - 1) \le \frac{\epsilon}{2} + 2T\epsilon_0^2 = \frac{\epsilon}{2} + \frac{\epsilon^2}{4\log(2/\delta)} \le \epsilon.$$

This proves (ϵ, δ)-DP. It remains to verify oracle utility.

For one call, the logarithmic parameter in Algorithm 2 is at most $\overline{\ell}$ because

$$\log\left(e + \frac{16 d R_0 \Lambda_{\max} \epsilon_0}{(2\Lambda_{\max}/s)\beta_0\delta_0}\right) = \log\left(e + \frac{32 d R_0 s \epsilon_0 T^2}{\beta\delta}\right) \le \log\left(e + \frac{32 d R_0 N T^2}{\beta\delta}\right) = \overline{\ell}.$$

Theorem 4.1 therefore bounds the additive error by $q \le C \frac{2\Lambda_{\max}}{s\epsilon_0} \sqrt{d R_0 H_0}\, \overline{\ell}^{3/2}$. Taking $C_1 \ge 40C$ in the definition of s_0 ensures $qV \le \alpha/20$. Each call fails with conditional probability at most β_0, so a union bound gives total failure probability at most $T\beta_0 = \beta/2$. The utility conclusions now follow from Theorem 3.1.

Finally, in covering mode any feasible constraint gives $1 \le \Lambda_{\max}\mathsf{OPT}$; in packing mode at least one constraint is tight at an optimum (otherwise the optimum could be scaled up), giving the same inequality. Thus$\Lambda_{\max}\mathsf{OPT} \ge 1$. Hence $W \ge 1 - \alpha/20$, so substituting the definition of ϵ_0 and $T = O(W \log N/\alpha^2)$ into s_0 gives

$$s_0 = \widetilde{O}\left(\frac{\sqrt{d}\, W \sqrt{T}}{\alpha\epsilon}\right) = \widetilde{O}\left(\frac{\sqrt{d}\, W^{3/2}\sqrt{\log N}}{\alpha^2\epsilon}\right) \qquad \text{for Algorithm 2,}$$

which proves the stated exception bound. □

AI USE STATEMENT

We used generative AI tools for proof ideas of Algorithm 2, for proofreading and verifying the correctness of the paper. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

Zeyuan Allen-Zhu, Yuanzhi Li, Aarti Singh, and Yining Wang. Near-optimal discrete optimization for experimental design: A regret minimization approach. *Mathematical Programming*, 186(1): 439–478, 2021.

Maria-Florina Balcan, Simon Shaolei Du, Yining Wang, and Adams Wei Yu. An improved gap-dependency analysis of the noisy power method. In *Conference on Learning Theory*, pp. 284–309. PMLR, 2016.

Afonso S Bandeira and Ramon van Handel. Sharp nonasymptotic bounds on the norm of random matrices with independent entries. *The Annals of Probability*, 44(4):2479–2506, 2016.

Yu Cheng, Ilias Diakonikolas, and Rong Ge. High-dimensional robust mean estimation in nearly-linear time. In *Proceedings of the thirtieth annual ACM-SIAM symposium on discrete algorithms*, pp. 2755–2771. SIAM, 2019a.

Yu Cheng, Ilias Diakonikolas, Rong Ge, and David P Woodruff. Faster algorithms for high-dimensional robust covariance estimation. In *Conference on Learning Theory*, pp. 727–757. PMLR, 2019b.

Cynthia Dwork, Krishnaram Kenthapadi, Frank McSherry, Ilya Mironov, and Moni Naor. Our data, ourselves: Privacy via distributed noise generation. In *EUROCRYPT*, volume 4004 of *Lecture Notes in Computer Science*, pp. 486–503. Springer, 2006.

Cynthia Dwork, Guy N. Rothblum, and Salil P. Vadhan. Boosting and differential privacy. In *Proceedings of the 51st Annual IEEE Symposium on Foundations of Computer Science*, pp. 51–60. IEEE, 2010.

Cynthia Dwork, Kunal Talwar, Abhradeep Thakurta, and Li Zhang. Analyze gauss: optimal bounds for privacy-preserving principal component analysis. In *Proceedings of the forty-sixth annual ACM symposium on Theory of computing*, pp. 11–20, 2014.

Alina Ene, Huy Le Nguyen, Ta Duy Nguyen, and Adrian Vladu. Solving linear programs with differential privacy. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2025)*, pp. 65–1. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2025.

Alina Ene, Huy Nguyen, Ta Duy Nguyen, and Adrian Vladu. Solving positive linear programs with differential privacy. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=zlSioMUQ2Y.

Michel X Goemans and David P Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *Journal of the ACM (JACM)*, 42(6): 1115–1145, 1995.

Olivier Guédon and Roman Vershynin. Community detection in sparse networks via grothendieck's inequality. *Probability Theory and Related Fields*, 165(3):1025–1049, 2016.

Moritz Hardt and Eric Price. The noisy power method: A meta algorithm with applications. *Advances in neural information processing systems*, 27, 2014.

Moritz Hardt and Aaron Roth. Beyond worst-case analysis in private singular vector computation. In *Proceedings of the forty-fifth annual ACM symposium on Theory of computing*, pp. 331–340, 2013.

Mark Herbster and Manfred K. Warmuth. Tracking the best linear predictor. *J. Mach. Learn. Res.*, 1:281–309, 2001.

Justin Hsu, Aaron Roth, Tim Roughgarden, and Jonathan R. Ullman. Privately solving linear programs. In *ICALP (1)*, volume 8572 of *Lecture Notes in Computer Science*, pp. 612–624. Springer, 2014.

Haim Kaplan, Yishay Mansour, Shay Moran, Uri Stemmer, and Nitzan Tur. On differentially private linear algebra. In *Proceedings of the 57th Annual ACM Symposium on Theory of Computing*, pp. 2362–2373, 2025.

Michael Kapralov and Kunal Talwar. On differentially private low rank approximation. In *Proceedings of the twenty-fourth annual ACM-SIAM symposium on Discrete algorithms*, pp. 1395–1414. SIAM, 2013.

Gert RG Lanckriet, Nello Cristianini, Peter Bartlett, Laurent El Ghaoui, and Michael I Jordan. Learning the kernel matrix with semidefinite programming. *Journal of Machine learning research*, 5(Jan):27–72, 2004.

László Lovász. On the shannon capacity of a graph. *IEEE Transactions on Information theory*, 25 (1):1–7, 1979.

Frank McSherry and Kunal Talwar. Mechanism design via differential privacy. In *FOCS*, pp. 94–103. IEEE Computer Society, 2007.

Andrés Muñoz Medina, Umar Syed, Sergei Vassilvitskii, and Ellen Vitercik. Private optimization without constraint violations. In *AISTATS*, volume 130 of *Proceedings of Machine Learning Research*, pp. 2557–2565. PMLR, 2021.

Dung Nguyen and Anil Kumar Vullikanti. Differentially private exact recovery for stochastic block models. In *International Conference on Machine Learning*, pp. 37798–37839. PMLR, 2024.

Julien Nicolas, César Sabater, Mohamed Maouche, Sonia Ben Mokhtar, and Mark Coates. Differentially private and decentralized randomized power method. *arXiv preprint arXiv:2411.01931*, 2024.

Jiming Peng and Yu Wei. Approximating k-means-type clustering via semidefinite programming. *SIAM journal on optimization*, 18(1):186–205, 2007.

Serge A. Plotkin, David B. Shmoys, and Éva Tardos. Fast approximation algorithms for fractional packing and covering problems. *Math. Oper. Res.*, 20(2):257–301, 1995.

Mohamed Seif, Yanxi Chen, Andrea J Goldsmith, and H Vincent Poor. Differentially private sketch-and-solve for community detection via semidefinite programming. *IEEE Journal on Selected Areas in Information Theory*, 5:331–346, 2024.

Zhao Song, Jianfei Xue, and Lichen Zhang. Differential privacy for euclidean jordan algebra with applications to private symmetric cone programming. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

Phuc Tran and Nisheeth K Vishnoi. Perturbation bounds for low-rank inverse approximations under noise. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

Joran van Apeldoorn and András Gilyén. Improvements in quantum sdp-solving with applications. In *46th International Colloquium on Automata, Languages, and Programming (ICALP 2019)*, pp. 99–1. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2019.

Eric Xing, Michael Jordan, Stuart J Russell, and Andrew Ng. Distance metric learning with application to clustering with side-information. *Advances in neural information processing systems*, 15, 2002.

A Additional Related Work

Private spectral algorithms and PCA. Dwork et al. (2014) study privacy-preserving PCA via Gaussian perturbation, and provide nearly optimal additive guarantees for releasing a low-dimensional subspace of a data matrix. Hardt & Roth (2013) study private singular vector computation, and show that the worst-case dimension dependence can sometimes be replaced by a coherence parameter. Hardt & Price (2014) introduce and analyze the noisy power method, with applications to private PCA and streaming PCA. This is further refined by Balcan et al. (2016), who show that the gap dependence can be substantially improved: for recovering a k dimensional subspace, rather than depending on the consecutive gap $\sigma_k - \sigma_{k+1}$, their bounds depend on a larger gap $\sigma_k - \sigma_{q+1}$, where q is an intermediate parameter.

All these methods are stated as recovery guarantees for a leading eigenvector or leading invariant subspace, which is the reason why their statements naturally involve an eigengap. Indeed, if the leading eigenspace is not separated from the rest of the spectrum, then the target subspace is unstable or not uniquely defined. By contrast, in this paper, we cannot assume that the matrices we use have isolated top eigenvalues. However, we do not need to identify a particular top eigenvector or recover a particular top eigenspace. It suffices to return a unit vector with a sufficiently large Rayleigh quotient for the input matrix. This objective remains meaningful even when many top eigenvalues are equal or nearly equal. Thus our private power iteration is tailored to a gap-independent approximation task, rather than to private PCA or subspace recovery.

More recent works have studied private and decentralized variants of power iteration and PCA Nicolas et al. (2024). These focus on private spectral estimation, typically under entry- and row-level adjacency for a data matrix or related sensitivity models. Their guarantees are designed for recovering a principal component or a leading subspace of a fixed data-derived matrix.

Private optimization under lower-sensitivity models. There is also a broad literature on private convex optimization in settings with lower sensitivity. A useful comparison is the work of Muñoz Medina et al. (2021), who study private optimization with linear constraints when the right-hand side is private and the algorithm is required to output a feasible solution without violating constraints. This model is substantially different from ours. There, the constraint matrix is public and only the right-hand side depends on private data. In our setting, the private record is an entire SDP constraint matrix. This high-sensitivity constraint-private model is the reason that exact satisfaction of all constraints is impossible in general and why our guarantees allow a controlled number of violated constraints.

Application-specific private SDP methods. Several works combine differential privacy with SDP relaxations for specific statistical problems, especially community detection under stochastic block models. Seif et al. (2024) study differentially private community detection over binary symmetric stochastic block models using an approach based on semidefinite programming. Nguyen & Vullikanti (2024) study exact recovery for several stochastic block models variants under edge differential privacy and use SDP estimators together with stability-based privacy mechanisms. These results are complementary to ours, since they exploit the structure of particular graph models and recovery guarantees, whereas our goal is to give general-purpose private algorithms for packing and covering SDPs under constraint adjacency.

B Differential Privacy

Gaussian mechanism. Let f be a function that takes as input a dataset $\mathcal{A}$ and outputs a vector in $\mathbb{R}^d$. We say f has ℓ_2-sensitivity Δ if on any neighboring inputs $\mathcal{A}$ and $\mathcal{A}'$, $\|f(\mathcal{A}) - f(\mathcal{A}')\|_2 \leq \Delta$. We have the following result by Dwork et al. (2006).

Theorem B.1. *Assume $0 < \epsilon \leq 1$ and $0 < \delta < 1$. Given a dataset $\mathcal{A}$ and a function f with ℓ_2-sensitivity Δ, the algorithm $\mathcal{M}$, which outputs $f(\mathcal{A}) + E$ where E is sampled from the Gaussian distribution $\mathcal{N}(0, \sigma^2 I)$ with $\sigma \geq \frac{\Delta}{\epsilon}\sqrt{2\log\frac{1.25}{\delta}}$, is (ϵ, δ)-DP.*

Adaptive composition. We use the standard advanced-composition theorem Dwork et al. (2010), stated here in the form needed below.

Theorem B.2. *Let $\delta' > 0$. An adaptive composition of K mechanisms, each conditionally (ϵ_0, δ_0)-DP, is $\left(\sqrt{2K\log(1/\delta')}\,\epsilon_0 + K\epsilon_0(e^{\epsilon_0} - 1),\ K\delta_0 + \delta'\right)$-DP.*

Theorem B.3. *Suppose $0 < \epsilon \le 1$ and $0 < \delta \le e^{-1}$. Consider K adaptive vector queries whose conditional ℓ_2 sensitivities are at most Δ_j. If query j is perturbed by an independent Gaussian vector with coordinate standard deviation*

$$\sigma_j \ge \frac{4\Delta_j\sqrt{K\log(1/\delta)}}{\epsilon},$$

then the complete adaptive transcript is (ϵ, δ)-DP.

Proof. Fix an ordered pair of neighboring inputs and condition at round j on a common preceding transcript. Let h_j be the difference between the two conditional query answers, so $\|h_j\|_2 \le \Delta_j$. Under the first input, the conditional privacy-loss increment of the Gaussian release is

$$L_j = \frac{\|h_j\|_2^2}{2\sigma_j^2} + \frac{\langle g_j, h_j\rangle}{\sigma_j}, \qquad g_j \sim \mathcal{N}(0, I).$$

Thus, conditionally on the past, $L_j - \mathbb{E}[L_j \mid \mathcal{F}_{j-1}]$ is sub-Gaussian with variance proxy $\|h_j\|_2^2/\sigma_j^2$. Define the predictable variance and drift bounds

$$V_{\max} = \frac{\epsilon^2}{16\log(1/\delta)}, \qquad A_{\max} = \frac{V_{\max}}{2} = \frac{\epsilon^2}{32\log(1/\delta)}.$$

The assumed noise level implies, for every transcript,

$$\sum_{j=1}^{K}\frac{\|h_j\|_2^2}{\sigma_j^2} \le V_{\max}, \qquad \sum_{j=1}^{K}\mathbb{E}[L_j \mid \mathcal{F}_{j-1}] = \sum_{j=1}^{K}\frac{\|h_j\|_2^2}{2\sigma_j^2} \le A_{\max}.$$

Applying the exponential supermartingale bound with the deterministic variance proxy $V_{\max}$ therefore gives, except with probability δ,

$$\sum_{j=1}^{K} L_j \le A_{\max} + \sqrt{2V_{\max}\log(1/\delta)} = \frac{\epsilon^2}{32\log(1/\delta)} + \frac{\epsilon}{2\sqrt{2}} \le \epsilon.$$

The same argument applies to every ordered neighboring pair. The privacy-loss characterization of approximate differential privacy now proves the theorem. □

Exponential mechanism. Let $Q(i, \mathcal{A})$ be a score function which takes as input a candidate i from a finite set S and a dataset $\mathcal{A}$ and outputs a real value. We say Q has sensitivity Δ if for neighboring inputs $\mathcal{A}$ and $\mathcal{A}'$ and for all $i \in S$, $|Q(i, \mathcal{A}) - Q(i, \mathcal{A}')| \le \Delta$. We have the following result.

Theorem B.4. *Given a dataset $\mathcal{A}$ and a score function Q with sensitivity Δ, the exponential mechanism $\mathsf{EM}_{\epsilon,Q}$, which outputs $i \in S$ with probability proportional to $\exp\left(\frac{\epsilon Q(i,\mathcal{A})}{2\Delta}\right)$, is ϵ-differentially private. Furthermore, with probability at least $1 - \beta$,*

$$Q(\mathsf{EM}_{\epsilon,Q}(\mathcal{A}), \mathcal{A}) \ge \max_{j\in S} Q(j, \mathcal{A}) - \frac{2\Delta\left(\log|S| + \log\frac{1}{\beta}\right)}{\epsilon}.$$

C Eigenvalue Inequalities

For reference, we record two standard eigenvalue and singular-value inequalities.

Theorem C.1 (Weyl's inequality)**.** *Let A and B be two symmetric matrices and for a matrix X, let $\lambda_i(X)$ denote the i-th biggest eigenvalue of X. We have*

$$\lambda_i(A + B) \le \lambda_i(A) + \lambda_{\max}(B).$$

Theorem C.2 (Corollary 3.9 (Bandeira & van Handel (2016)))**.** *Let X be a $d \times d$ symmetric matrix where X_{ij} are i.i.d from $\mathcal{N}(0, \sigma^2)$ for $i \ge j$. Then for any $0 < \epsilon \le \frac{1}{2}$ and every $t \ge 0$*

$$\Pr\left[\|X\|_2 \ge 2(1+\epsilon)\,\sigma\left(\sqrt{d} + \frac{3\sqrt{\log d}}{\sqrt{\log(1+\epsilon)}}\right) + t\right] \le e^{-t^2/4}.$$

D Martingale Inequalities

Theorem D.1 (Doob's maximal inequality). *Let $(X_t)_{t\geq 0}$ be a nonnegative supermartingale. Then for any positive number C,*

$$\Pr\left[\max_{t\geq 0} X_t \geq C\right] \leq \frac{\mathbb{E}\left[X_0\right]}{C}.$$

Theorem D.2 (Doob's L^2 inequality). *Let $(X_t)_{t\geq 0}$ be a square-integrable martingale. Then for any positive number C,*

$$\Pr\left[\max_{0\leq t\leq T} |X_t| \geq C\right] \leq \frac{\mathbb{E}\left[X_T^2\right]}{C}.$$

E Truncated softmax and softmin Properties

We restate the following property of these functions from Ene et al. (2026).

Lemma E.1. *(Ene et al., 2026, Proposition 5) For any x, we have $\nabla smax^U(x) \in \mathcal{D}^U$ and $\nabla smin^U(x) \in \mathcal{D}^U$. Furthermore, for $u \in \mathbb{R}^n_{\geq 0}$ such that $D = \max_i\{u_i\} \leq 1$,*

$$smax^U(x+u) \leq smax^U(x) + (1+D)\left\langle \nabla smax^U(x), u\right\rangle, \tag{8}$$

$$\text{and } smin^U(x+u) \geq smin^U(x) + (1-D)\left\langle \nabla smin^U(x), u\right\rangle. \tag{9}$$

Lemma E.2. *Let $m \geq s$, $U = 1/s$, $x \in \mathbb{R}^m$, and $a \in \mathbb{R}$. Let $x' = (x, a)$ be a $(m+1)$-dimensional vector constructed by concatenating a to x. Write $p = \nabla smax^U(x)$ and $p' = \nabla smax^U((x,a))$. Then $p'_i \leq p_i$ for every $i \leq m$ and*

$$\sum_{i=1}^{m} |p_i - p'_i| = p'_{m+1} \leq \frac{1}{s}.$$

The same statements hold with smax replaced by smin. Consequently, if $\|A_i\|_2 \leq \Lambda$ for all $i \leq m+1$, then

$$\left\|\sum_{i=1}^{m} p_i A_i - \sum_{i=1}^{m+1} p'_i A_i\right\|_2 \leq \frac{2\Lambda}{s}.$$

Proof. The strong convexity of the negative entropy $\omega(r)$ makes each optimizer unique. The Karush-Kuhn-Tucker conditions give the following explicit formula for $p = \nabla \text{smax}^U(x)$ (see also (Ene et al., 2026, Section 2.4)):

$$p_i = \min\{U, ce^{x_i}\},$$

where the normalization factor $c \geq 0$ is chosen so that the coordinates p_i sum to one. After adding a coordinate, the normalization factor cannot increase; hence $p'_i \leq p_i$ for every old coordinate. Since the first m coordinates of p' sum to $1 - p'_{m+1}$, their total decrease is exactly $p'_{m+1} \leq U$. The softmin statement follows by applying the same argument to $-x$. Finally, the triangle inequality gives

$$\left\|\sum_{i=1}^{m}(p_i - p'_i)A_i - p'_{m+1}A_{m+1}\right\|_2 \leq \Lambda\left(\sum_{i=1}^{m}|p_i - p'_i| + p'_{m+1}\right) \leq 2\Lambda U.$$

□

F Missing Proofs from Section 3

Proof of Theorem 3.1. Fix an iteration t and condition on the preceding iterations. Let p_t and $\overline{A}_t$ be as in Algorithm 1. For covering, if X^* is an optimal solution, then

$$1 \leq \langle \overline{A}_t, X^*\rangle \leq \lambda_{\max}(\overline{A}_t)\text{Tr}(X^*) = \lambda_{\max}(\overline{A}_t)\mathsf{OPT}.$$

Therefore, if the oracle is successful then the returned solution u_t satisfies

$$\left\langle p_t, \langle \mathcal{A}, V u_t u_t^\top \rangle \right\rangle = V u_t^\top \overline{A}_t u_t \geq V \lambda_{\max}(\overline{A}_t) - qV \geq \frac{V}{\mathsf{OPT}} - qV \geq 1 - \frac{\alpha}{10}.$$

For packing, feasibility of X^* gives

$$\lambda_{\min}(\overline{A}_t)\mathsf{OPT} \leq \langle \overline{A}_t, X^* \rangle \leq 1,$$

and therefore

$$\left\langle p_t, \langle \mathcal{A}, V u_t u_t^\top \rangle \right\rangle \leq V \lambda_{\min}(\overline{A}_t) + qV \leq \frac{V}{\mathsf{OPT}} + qV \leq 1 + \frac{\alpha}{10}.$$

For both cases, we have

$$\|\eta \langle \mathcal{A}, V u_t u_t^\top \rangle\|_\infty \leq \eta \Lambda_{\max} V = \frac{\alpha}{10}.$$

For covering, the truncated-softmin inequality in Lemma E.1implies

$$\begin{aligned}
\operatorname{smin}^U \left(\eta \left\langle \mathcal{A}, X_{t+1} \right\rangle\right) &= \operatorname{smin}^U \left(\eta \left\langle \mathcal{A}, X_t + V u_t u_t^\top \right\rangle\right) \\
&\geq \operatorname{smin}^U \left(\eta \left\langle \mathcal{A}, X_t \right\rangle\right) \\
&\quad + \left(1 - \left\|\eta \left\langle \mathcal{A}, V u_t u_t^\top \right\rangle\right\|_\infty\right) \left\langle \nabla \operatorname{smin}^U \left(\eta \left\langle \mathcal{A}, X_t \right\rangle\right), \eta \left\langle \mathcal{A}, V u_t u_t^\top \right\rangle \right\rangle \\
&\geq \operatorname{smin}^U \left(\eta \left\langle \mathcal{A}, X_t \right\rangle\right) + \left(1 - \frac{\alpha}{10}\right) \left\langle p_t, \eta \langle \mathcal{A}, V u_t u_t^\top \rangle \right\rangle \\
&= \operatorname{smin}^U \left(\eta \left\langle \mathcal{A}, X_t \right\rangle\right) + \left(1 - \frac{\alpha}{10}\right)^2.
\end{aligned}$$

Unrolling this inequality from iteration $T-1$ back to 0, we have

$$\begin{aligned}
\operatorname{smin}^U(\eta \langle \mathcal{A}, X_T \rangle) &\geq \operatorname{smin}^U(0) + \eta T \left(1 - \frac{\alpha}{10}\right)^2 \\
&\geq -\log N + \eta T \left(1 - \frac{\alpha}{10}\right)^2 \geq \eta T (1 - \alpha),
\end{aligned}$$

where the last inequality uses $\eta T \geq 2 \log N / \alpha$. Let $v_i = \langle A_i, X_T \rangle$ and order the coordinates as $v_{\pi(1)} \leq \cdots \leq v_{\pi(N)}$ where π is a permutation over $[N]$. Using the uniform distribution on the s smallest coordinates of $\langle \mathcal{A}, X_T \rangle$ gives

$$\eta T (1 - \alpha) \leq \operatorname{smin}^U(\eta v) \leq \frac{\eta}{s} \sum_{j=1}^{s} v_{\pi(j)} - \log s.$$

Hence $v_{\pi(s)} \geq s^{-1} \sum_{j=1}^{s} v_{\pi(j)} \geq T(1-\alpha)$, so all but at most s coordinates of $\langle \mathcal{A}, \overline{X} \rangle$ are at least $1 - \alpha$.

Similarly, for packing problems, the truncated-softmax inequality in Lemma E.1gives

$$\operatorname{smax}^U(\eta \langle \mathcal{A}, X_T \rangle) \leq \operatorname{smax}^U(0) + \eta T \left(1 + \frac{\alpha}{10}\right)^2 \leq \log N + \eta T \left(1 + \frac{\alpha}{10}\right)^2 \leq \eta T (1 + \alpha).$$

Ordering the coordinates as $v_{\pi(1)} \geq \cdots \geq v_{\pi(N)}$ and using the uniform distribution on the s largest coordinates gives

$$\frac{\eta}{s} \sum_{j=1}^{s} v_{(j)} + \log s \leq \operatorname{smax}^U(\eta v) \leq \eta T (1 + \alpha).$$

Thus $v_{\pi(s)} \leq s^{-1} \sum_{j=1}^{s} v_{\pi(j)} \leq T(1+\alpha)$, and all but at most s coordinates of $\langle \mathcal{A}, \overline{X} \rangle$ are at most $1 + \alpha$. Since every update has trace V, $\operatorname{Tr}(\overline{X}) = V$ in both cases. A union bound over T oracle calls gives failure probability at most $T\beta_0$. Finally, the privacy guarantee follows from Theorem B.2 for adaptive composition of (ϵ, δ)-DP mechanisms. □

G Missing Proofs from Section (2)

Proof of Lemma 4.2. Let $\mu = \lambda_{\max}(B)$, set $D = B/\mu$ and $\eta = \sigma/\mu$. Thus $I/2 \preceq D \preceq I$ and $\eta \le 2\sigma$. For convenience, consider the unnormalized process

$$w_0 = x_0, \qquad w_{t+1} = Dw_t + \eta\|w_t\|_2 g_t, \qquad Q_t = \|w_t\|_2^2. \tag{10}$$

An induction shows that $x_t = w_t/\|w_t\|_2$ almost surely: after multiplying the update defining x_{t+1} by the positive scalar $\|w_t\|_2/\mu$, its numerator becomes exactly the right-hand side of (10). In other words, normalizing after every iteration has the same effect as normalizing only once at the end.

Claim G.1. For constant c sufficiently small, with probability at least $7/20$, we have $Q_T \ge \frac{1}{64}$.

Proof. Let $\mathcal{F}_t$ contain the initialization and the noise through time $t-1$. From $D \preceq I$ and the fact that conditioned on prior iterations, g_t is a standard Gaussian vector, we have

$$\mathbb{E}[Q_{t+1} \mid \mathcal{F}_t] = \|Dw_t\|_2^2 + \eta^2 d Q_t \le (1+\eta^2 d) Q_t.$$

Hence $Q_t/(1+\eta^2 d)^t$ is a nonnegative supermartingale. By setting the constant c in (5) sufficiently small, we have $(1+\eta^2 d)^T \le (1+\frac{\sigma^2 d}{4})^T \le (1+\frac{c}{4T})^T \le 2$. Therefore, Doob's maximal inequality gives

$$\Pr\left(\max_{0\le t\le T} Q_t > 128\right) \le \frac{2\mathbb{E}\left[Q_0\right]}{128} \le \frac{1}{64}. \tag{11}$$

Fix a unit leading eigenvector v of D and write $a_t = v^\top w_t$. Since $Dv = v$,

$$a_{t+1} = v^\top\left(Dw_t + \eta\|w_t\|_2 g_t\right) = a_t + \eta\sqrt{Q_t}\cdot\xi_t, \qquad \text{where } \xi_t = v^\top g_t \sim \mathcal{N}(0,1).$$

Let $\tau = \min\left(\{t \in \{0,\dots,T\} : Q_t > 128\} \cup \{T+1\}\right)$ and stop this martingale before the first increment with $t \ge \tau$. Its predictable quadratic variation through time T is at most $128\eta^2 T$. Moreover, by the rotational invariance of the Gaussian distribution, $a_0 = v^\top w_0$ is identically distributed as $z_1/\|z\|_2$. Since

$$\Pr(|z_1| \ge 1/2,\ \|z\|_2 \le 2\sqrt{d}) \ge 1 - \frac{1}{\sqrt{2\pi}} - \frac{1}{4} > \frac{7}{20},$$

we have $|a_0| \ge 1/(4\sqrt{d})$ with probability greater than $7/20$. Conditional on any such initialization, Doob's L^2 inequality yields

$$\Pr\left(\max_{k\le T}|a_{k\wedge\tau} - a_0| > \frac{|a_0|}{2} \;\middle|\; a_0\right) \le \frac{128\eta^2 T}{a_0^2/4} \le 8192\eta^2 dT \le \frac{1}{64},$$

where the last inequality follows by choosing c small enough. On the complement of this event and the event in (11), we have

$$|a_T| \ge \frac{|a_0|}{2} \ge \frac{1}{8\sqrt{d}},$$

which implies $Q_T = \|w_T\|_2^2 \ge a_T^2 \ge 1/(64d)$. □

Next we relate the change between Q_{t+1} and Q_t and the quality of the iterate $\lambda_{\max}(D) - x_t^\top D x_t$. Let $r_t = Dx_t$, $s_t = \|r_t\|_2^2 = x_t^\top D^2 x_t$, and $L_t = 1 - x_t^\top D x_t$.

Claim G.2. $\sum_{t<T} L_t \le \sum_{t<T} m_t + 4\eta^2\|g_t\|_2^2 - \log\frac{Q_{t+1}}{Q_t}$, where $m_t = \frac{2\eta\langle r_t, g_t\rangle}{s_t}$.

Proof. Since $I/2 \preceq D \preceq I$, we have $s_t \ge \lambda_{\min}\left(D\right)^2 \ge 1/4$ and $D^2 \preceq D$. Consequently, using $\log(1+y) \le y$ for $y > -1$,

$$\log\frac{Q_{t+1}}{Q_t} = \log\|r_t + \eta g_t\|_2^2 = \log\left(1 + \frac{2\eta\langle r_t, g_t\rangle + \eta^2\|g_t\|_2^2}{s_t}\right)$$
$$\le \log s_t + \frac{2\eta\langle r_t, g_t\rangle + \eta^2\|g_t\|_2^2}{s_t}$$

$$\leq -L_t + m_t + 4\eta^2 \|g_t\|_2^2, \tag{12}$$

Indeed, $s_t = x_t^\top D^2 x_t \leq x_t^\top D x_t = 1 - L_t$, so $\log s_t \leq \log(1 - L_t) \leq -L_t$. We obtain

$$L_t \leq m_t + 4\eta^2 \|g_t\|_2^2 - \log \frac{Q_{t+1}}{Q_t}.$$

□

Finally, we show the lemma's guarantee. We have, the variables m_t are martingale differences and

$$\mathbb{E}[m_t^2 \mid \mathcal{F}_t] = \frac{4\eta^2}{s_t^2} \|r_t\|_2^2 = \frac{4\eta^2}{s_t} \leq 16\eta^2.$$

Thus Chebyshev's inequality for martingales, together with $\eta^2 dT \leq 4c$, gives

$$\Pr\left(\sum_{t<T} m_t > 1\right) \leq 16\eta^2 T \leq \frac{1}{64},$$

for a universal constant c sufficiently small. Also, Markov's inequality gives

$$\Pr\left(\sum_{t<T} \|g_t\|_2^2 > 64dT\right) \leq \frac{1}{64}.$$

Intersect these two events with the initialization, norm, and leading-coordinate events above: their joint probability is at least $\frac{7}{20} - \frac{4}{64} > \frac{1}{4}$. On this intersection, summing (12) and using $Q_0 = 1$ and $Q_T \geq 1/(64d)$ yields

$$\sum_{t<T} L_t \leq \sum_{t<T} m_t + 4\eta^2 \|g_t\|_2^2 - \log \frac{Q_{t+1}}{Q_t} \leq \log(64d) + 1 + 256\eta^2 dT.$$

At least one $t < T$ therefore satisfies

$$\lambda_{\max}(B) - x_t^\top B x_t = \mu L_t \leq C\left(\frac{\log(ed)}{T} + \sigma^2 d\right),$$

because $\mu \leq 1$ and $\eta \leq 2\sigma$. Finally, independent trajectories fail simultaneously with probability at most $(3/4)^R \leq e^{-R/4}$, which proves the final assertion. □

Proof of Theorem 4.1. **Privacy**. The branch condition $q_0 \geq 2L$ depends only on public parameters; in that branch the output is fixed and hence private. Consider the remaining branch and write $m = 2RT$ and $K = \log(1/\delta)$. Conditioned on the transcript before an update, $x_{s,r,t}$ is fixed and has unit norm. Consequently, for neighboring inputs,

$$\|(B_s - B'_s)x_{s,r,t}\|_2 \leq \|B_s - B'_s\|_2 \leq \Delta.$$

Thus every one of the m adaptive vector queries has ℓ_2 sensitivity at most Δ. Adding Gaussian noise with standard deviation ΔG_{2RT} and the identity $G_{2RT} = 8\sqrt{m \log(1/\delta)}/\epsilon$ meet the hypothesis of Theorem B.3 with privacy parameter $\epsilon/2$. It follows directly that the full Gaussian phase is jointly $(\epsilon/2, \delta)$-DP.

Condition on the complete noised-query transcript and on all data-independent initialization vectors $x_{s,r,0}$. The two candidate sets are then fixed. For every candidate x, the score $x^\top B_s x$ has sensitivity at most Δ. The displayed sampling probability in Algorithm 2 is exactly the exponential mechanism with privacy parameter $\epsilon/4$, since its exponent is $(\epsilon/4)x^\top B_s x/(2\Delta)$. Each selector is therefore $\epsilon/4$-DP, even after conditioning on all preceding outputs. Adaptive sequential composition of the Gaussian phase and the two selectors proves (ϵ, δ)-differential privacy.

Utility. First suppose $q_0 \geq 2L$. Every unit vector has both largest- and smallest-eigenvalue Rayleigh loss at most $2L$, so the fixed output in the early branch proves the claimed bound. We henceforth assume $q_0 < 2L$. If $T = 1$, then $\lfloor c_0 L\ell/q_0 \rfloor \leq 1$ and hence $q_0 > c_0 L\ell/2 \geq c_0 L/2$. The same trivial $2L = O(q_0)$ bound proves this case after increasing the universal constant in the theorem. It remains to consider $T \geq 2$.

Set $K = \log(1/\delta)$ and $\sigma = \Delta G_{2RT}$. As a reminder, $\Delta = \frac{S}{4L}$ and $G_{2RT} = 8\sqrt{2RT\log(1/\delta)}/\epsilon$, direct substitution gives

$$\sigma^2 = \frac{8S^2RTK}{L^2\epsilon^2}. \tag{13}$$

Since $T = \lfloor c_0 L\ell/q_0 \rfloor \geq 2$,

$$\frac{c_0 L\ell}{2q_0} \leq T \leq \frac{c_0 L\ell}{q_0}. \tag{14}$$

Using the upper bound in (14) twice and the definition of q_0, we obtain

$$\begin{aligned}\sigma^2 dT &\leq \frac{8S^2RK}{L^2\epsilon^2} \cdot d \cdot \left(\frac{c_0 L\ell}{q_0}\right)^2 = \frac{8c_0^2 S^2 dRK\ell^2}{\epsilon^2 q_0^2} \\ &= \frac{8c_0^2 S^2 dRK\ell^2}{\left(C_0 S\sqrt{dR\log\frac{8}{\delta}}\,\ell^{3/2}\right)^2} = \frac{8c_0^2}{C_0^2}\frac{K}{\log(8/\delta)}\frac{1}{\ell} \leq c.\end{aligned}$$

Thus the stability hypothesis of Lemma 4.2 holds when c_0/C_0 is a sufficiently small universal constant.

We next bound the loss supplied by that lemma. The inequality $q_0 < 2L$, together with the definition of $\ell = \log\left(e + \frac{16dRL\epsilon}{S\beta\delta}\right)$, implies $\ell \geq \log(ed)$ after taking $C_0 \geq 1$. Indeed, rearranging $q_0 < 2L$ gives

$$\frac{L\epsilon}{S} > \frac{C_0}{2}\sqrt{dR\log(8/\delta)}\,\ell^{3/2}.$$

Since $R \geq 1$, $\beta\delta < 1/16$, and $\ell \geq 1$, this implies $\frac{16dRL\epsilon}{S\beta\delta} \geq e(d-1)$,which is equivalent to the claimed lower bound on ℓ. The lower bound in (14) therefore gives

$$\frac{\log(ed)}{T} \leq \frac{2q_0\log(ed)}{c_0 L\ell} \leq \frac{2q_0}{c_0 L}.$$

Likewise, using (13) and the upper bound on T,

$$\begin{aligned}\sigma^2 d &\leq \frac{8S^2RK}{L^2\epsilon^2} \cdot d \cdot \frac{c_0 L\ell}{q_0} = \frac{8c_0 S^2 dRK\ell}{L\epsilon^2 q_0} \\ &= \frac{q_0}{L} \cdot \frac{8c_0 S^2 dRK\ell}{\epsilon^2 q_0^2} = \frac{q_0}{L} \cdot \frac{8c_0 S^2 dRK\ell}{\left(C_0 S\sqrt{dR\log\frac{8}{\delta}}\,\ell^{3/2}\right)^2} \\ &= \frac{q_0}{L} \cdot \frac{8c_0 K}{C_0^2\log\frac{8}{\delta}\,\ell^2} \leq \frac{q_0}{L}\cdot\frac{8c_0}{C_0^2}\end{aligned}$$

Lemma 4.2 now shows that any one trajectory contains a candidate with B_s-Rayleigh loss at most $\frac{Cq_0}{L}$ with probability at least $1/4$ when we raise the constant C to absorb the factor $\left(\frac{2}{c_0} + \frac{8c_0}{C_0^2}\right)$. The R trajectories for a fixed sign are independent. Hence the probability that none contains such a candidate is at most

$$e^{-R/4} \leq e^{-2\log(8/\beta)} \leq \frac{\beta}{4}.$$

There are $R(T+1)$ candidates for each sign. The standard exponential-mechanism tail bound says that, except with probability $\beta/4$ for that sign,

$$x_s^\top B_s x_s \geq \max_{x\in\mathcal{C}_s} x^\top B_s x - \frac{8\Delta}{\epsilon}\log\frac{4R(T+1)}{\beta}. \tag{15}$$

To bound the last term, observe from the definitions that

$$T \leq \frac{c_0}{C_0}\frac{L\epsilon/S}{\sqrt{dR\log(8/\delta)\ell}} \leq \frac{L\epsilon}{S}.$$

Because $T \geq 2$, we have $T+1 \leq 3T/2$. Together with the preceding bound on T and $\delta < 1/4$,

$$\frac{4R(T+1)}{\beta} \leq \frac{6RL\epsilon}{S\beta} \leq e + \frac{16dRL\epsilon}{S\beta\delta} = e^{\ell}.$$

Thus $\log(4R(T+1)/\beta) \leq \ell$. Therefore

$$\frac{8\Delta}{\epsilon} \log \frac{4R(T+1)}{\beta} \leq \frac{2S\ell}{L\epsilon} \leq C\frac{q_0}{L}.$$

The two candidate-existence events and the two selection events fail with total probability at most β.

Finally, Equation (4) gives the exact identities

$$\lambda_{\max}(A) - x^\top Ax = 4L\left(\lambda_{\max}(B_+) - x^\top B_+ x\right),$$
$$x^\top Ax - \lambda_{\min}(A) = 4L\left(\lambda_{\max}(B_-) - x^\top B_- x\right).$$

Thus both losses are at most Cq_0 on the joint success event. Since $q_0 = C_0(S/\epsilon)\sqrt{dR\log(8/\delta)}\ell^{3/2}$, increasing the universal constant C in the theorem covers this regime as well as the two trivial regimes above. This proves the result. □

An implication of Theorem 4.1 is that we can also privately estimate the top and bottom eigenvalue of a matrix with error $\widetilde{O}\left(\frac{\sqrt{d}S}{\epsilon}\right)$.

Corollary G.1. *There is an (ϵ, δ)-differentially private algorithm that releases numerical estimates $\widehat{\lambda}_{\max}$ and $\widehat{\lambda}_{\min}$ such that, with probability at least $1-\beta$,*

$$|\widehat{\lambda}_{\max} - \lambda_{\max}(A)| \leq \widetilde{O}\left(\frac{\sqrt{d}S}{\epsilon}\right), \qquad |\widehat{\lambda}_{\min} - \lambda_{\min}(A)| \leq \widetilde{O}\left(\frac{\sqrt{d}S}{\epsilon}\right).$$

Proof. Run Algorithm 2 with privacy parameter $\epsilon/2$ and failure parameter $\beta/2$, and then release

$$\widehat{\lambda}_{\max} = u_{\max}^\top A u_{\max} + \xi_+, \qquad \widehat{\lambda}_{\min} = u_{\min}^\top A u_{\min} + \xi_-,$$

where ξ_+, ξ_- are independent Laplace random variables of scale $4S/\epsilon$. Conditional on all preceding outputs, each vector is fixed and

$$|u^\top Au - u^\top A'u| \leq \|A - A'\|_2 \leq S.$$

Thus each scalar release is $\epsilon/4$-DP. Sequential composition with Theorem 4.1 proves (ϵ, δ)-differential privacy.

For a Laplace variable of scale $4S/\epsilon$,

$$\Pr\left(|\xi| > \frac{4S}{\epsilon}\log\frac{4}{\beta}\right) = \frac{\beta}{4}.$$

A union bound controls both noises except with probability $\beta/2$. On that event and the theorem's success event, the variational inequalities $u_{\max}^\top Au_{\max} \leq \lambda_{\max}(A)$ and $u_{\min}^\top Au_{\min} \geq \lambda_{\min}(A)$ give

$$|\widehat{\lambda}_{\max} - \lambda_{\max}(A)| \leq q + \frac{4S}{\epsilon}\log\frac{4}{\beta},$$
$$|\widehat{\lambda}_{\min} - \lambda_{\min}(A)| \leq q + \frac{4S}{\epsilon}\log\frac{4}{\beta}.$$

Replacing ϵ by $\epsilon/2$ and β by $\beta/2$ changes only universal and logarithmic factors in q. This proves the claim. □

H PRIVATE ORACLES VIA THE GAUSSIAN MECHANISM

Algorithm 3 Additive Spectral Oracles via the Gaussian Mechanism

1: **Input**: symmetric matrix A, spectral-norm sensitivity S, and privacy parameters ϵ, δ
2: Let $G = \frac{\sqrt{d}S}{\epsilon}\sqrt{2\log\frac{2}{\delta}}$
3: Let E be a symmetric matrix where the upper triangle (including the diagonal) is i.i.d. samples from $\mathcal{N}(0, G)$
4: Let $u_{\max}$ and $u_{\min}$ be the top and bottom eigenvectors of $A + E$
5: Output $u_{\max}$ and $u_{\min}$

In this section, we provide a simple polynomial time implementation of the additive spectral oracles using the Gaussian mechanism. This

Theorem H.1. *Given $\epsilon, \delta, \beta > 0$, and parameter $S > 0$, Algorithm 3 is (ϵ, δ)-differentially private and with probability at least $1 - \beta$ its outputs satisfy*

$$u_{\max}^\top A u_{\max} \geq \lambda_{\max}(A) - q, \qquad u_{\min}^\top A u_{\min} \leq \lambda_{\min}(A) + q,$$

where

$$q = 2\left(3\sqrt{d} + 15\sqrt{\log d} + 2\sqrt{\log\frac{1}{\beta}}\right)\frac{\sqrt{d}S}{\epsilon}\sqrt{2\log\frac{2}{\delta}} = \tilde{O}\left(\frac{dS}{\epsilon}\right).$$

Proof. **Privacy**. We compute the sensitivity of the input. Our privacy model defines the notion of neighboring input in terms of the spectral norm of the difference, yet we need to provide a bound on its ℓ_2 norm; in other words, for two PSD matrices A and A' such that $\|A' - A\|_2 \leq S$, we need to bound $\|A' - A\|_F$. We have

$$\|A' - A\|_F \leq \sqrt{d}\,\|A' - A\|_2 \leq \sqrt{d}S.$$

Therefore adding noise $E \sim \mathcal{N}(0, G^2)$ for $G \sim \frac{\sqrt{d}S}{\epsilon}\sqrt{2\log\frac{2}{\delta}}$ and releasing $A + E$ is (ϵ, δ)-differentially private, by the standard guarantee of the Gaussian mechanism. Finding top and bottom eigenvectors of $A + E$ does not impact privacy due to post processing.

Utility. By Theorem C.2, we have with probability at least $1 - \beta$,

$$\|E\|_2 \leq \left(3\sqrt{d} + 15\sqrt{\log d} + 2\sqrt{\log\frac{1}{\beta}}\right)G.$$

We have

$$\begin{aligned}
u_{\max}^\top A u_{\max} &= u_{\max}^\top (A + E)\, u_{\max} - u_{\max}^\top E u_{\max} \\
&\geq \lambda_{\max}(A + E) - \|E\|_2 \\
&\overset{(*)}{\geq} \lambda_{\max}(A) - \lambda_{\max}(E) - \|E\|_2 \\
&\geq \lambda_{\max}(A) - 2\left(3\sqrt{d} + 15\sqrt{\log d} + 2\sqrt{\log\frac{1}{\beta}}\right)G,
\end{aligned}$$

where in $(*)$, we use Weyl's inequality C.1. Similarly,

$$\begin{aligned}
u_{\min}^\top A u_{\min} &= u_{\min}^\top (A + E)\, u_{\min} - u_{\min}^\top E u_{\min} \\
&\leq \lambda_{\min}(A + E) + \|E\|_2 \\
&\leq \lambda_{\min}(A) + \lambda_{\max}(E) + \|E\|_2 \\
&\leq \lambda_{\min}(A) + 2\left(3\sqrt{d} + 15\sqrt{\log d} + 2\sqrt{\log\frac{1}{\beta}}\right)G.
\end{aligned}$$

□

Theorem H.2. *Assume $0 < \epsilon \le 1$, $0 < \delta \le e^{-1}$, $0 < \beta < 1/2$, and $0 < \alpha < 1$. Using Algorithm 3 to implement the additive spectral oracles in Algorithm 1, the exception budget can be set to*

$$s \le \min\left\{N, \widetilde{O}\left(\frac{\sqrt{d}\, W^{3/2}\sqrt{\log N}}{\alpha^2 \epsilon}\right)\right\},$$

where $W = \Lambda_{\max} V$ and the suppressed factors in the $\widetilde{O}$-notation are logarithmic in the displayed parameters.

Proof. Let $s_0 = \left\lceil 4\left(3\sqrt{d} + 15\sqrt{\log d} + 2\sqrt{\log \frac{1}{\beta_0}}\right)\frac{\sqrt{d}}{\epsilon_0}\sqrt{2\log\frac{2}{\delta}}\frac{WC_1}{\alpha}\right\rceil$, $s = \min\{N, s_0\}$. If $s = N$, output the fixed matrix VI/d. Otherwise run Algorithm 1 with exception budget $s = s_0$. At every outer iteration, evoke Algorithm 3 as the oracle with parameter (ϵ_0, δ_0). Apply Theorem B.2 with $\delta' = \delta/2$. The composed δ parameter is $T\delta_0 + \delta/2 = \delta$. Since $\epsilon_0 \le 1$ and $e^{\epsilon_0} - 1 \le 2\epsilon_0$, the composed ϵ parameter is at most

$$\sqrt{2T\log(2/\delta)}\,\epsilon_0 + T\epsilon_0(e^{\epsilon_0} - 1) \le \frac{\epsilon}{2} + 2T\epsilon_0^2 = \frac{\epsilon}{2} + \frac{\epsilon^2}{4\log(2/\delta)} \le \epsilon.$$

This proves (ϵ, δ)-DP. Theorem H.1 bounds the error of the oracle

$$q \le 4\left(3\sqrt{d} + 15\sqrt{\log d} + 2\sqrt{\log\frac{1}{\beta_0}}\right)\frac{\sqrt{d}}{\epsilon_0}\sqrt{2\log\frac{2}{\delta}}\frac{\Lambda_{\max}}{s}.$$

For C_1 sufficiently large, we guarantee that $qV \le \alpha/20$. Each call fails with conditional probability at most β_0, so a union bound gives total failure probability at most $T\beta_0 = \beta/2$. Therefore utility conclusions now follow from Theorem 3.1.

In covering mode any feasible constraint gives $1 \le \Lambda_{\max}\mathsf{OPT}$; in packing mode at least one constraint is tight at an optimum (otherwise the optimum could be scaled up), giving the same inequality. Thus$\Lambda_{\max}\mathsf{OPT} \ge 1$. Hence $W \ge 1 - \alpha/20$, so the ceiling in the definition of T changes it by only a universal factor. Therefore, substituting the definition of ϵ_0 and $T = O(W\log N/\alpha^2)$ into s_0 gives

$$s_0 = \widetilde{O}\left(\frac{d\,W\sqrt{T}}{\alpha\epsilon}\right) = \widetilde{O}\left(\frac{d\,W^{3/2}\sqrt{\log N}}{\alpha^2\epsilon}\right) \qquad \text{for Algorithm 3}$$

which proves the stated exception bound. □