Differential Privacy for Euclidean Jordan Algebra with Applications to Private Symmetric Cone Programming

Zhao Song
University of California, Berkeley
magic.linuxkde@gmail.com

Jianfei Xue
New York University
jx898@nyu.edu

Lichen Zhang
MIT CSAIL
lichenz@csail.mit.edu

Abstract

In this paper, we study differentially private mechanisms for functions whose outputs lie in a Euclidean Jordan algebra. Euclidean Jordan algebras capture many important mathematical structures and form the foundation of linear programming, second-order cone programming, and semidefinite programming. Our main contribution is a generic Gaussian mechanism for such functions, with sensitivity measured in ℓ_2, ℓ_1, and ℓ_∞ norms. Notably, this framework includes the important case where the function outputs are symmetric matrices, and sensitivity is measured in the Frobenius, nuclear, or spectral norm. We further derive private algorithms for solving symmetric cone programs under various settings, using a combination of the multiplicative weights update method and our generic Gaussian mechanism. As an application, we present differentially private algorithms for semidefinite programming, resolving a major open question posed by [Hsu, Roth, Roughgarden, and Ullman, ICALP 2014].

1 Introduction

As modern machine learning leverages increasingly large models and ever-growing datasets, protecting the privacy of data has become a paramount concern. Differential privacy [DMNS06, DKM+06] has emerged as the gold standard for data privacy, owing to its simplicity, practicality, and strong theoretical guarantees. Designing machine learning algorithms with differential privacy (DP) is particularly valuable, as it enables protection of both user data and model parameters [BIK+17, GKN17, MRTZ18, TLC+20, ATMR21, NBD22, FHL+24]. However, most differentially private algorithms for machine learning are developed in a case-by-case manner — tailored specifically to individual problems to ensure good privacy and utility trade-offs. More generic approaches, such as making stochastic gradient descent (SGD) differentially private [XLW12, BST14, ACG+16], have gained significant traction. Since SGD is a core subroutine in many training algorithms, this line of work provides automatic privacy guarantees across a wide range of models. Following this direction, we develop differentially private algorithms for solving *symmetric cone programs* (SCP), a broad class of convex optimization problems that includes linear programs (LP), second-order cone programs (SOCP), and semidefinite programs (SDP). These convex formulations arise in many machine learning applications, including support vector machines [BGV92, CV95, MMR+01, Joa06, CL11, SSSSC11, GSZ25], matrix completion [CT10, Rec11, CR12, JNS13, ZWL15, GLZ17, KLL+23, GSYZ24, SYYZ25], robust mean and covariance estimation [CDG19, CDGW19], experimental design [vAG19, AZLSW20], and sparse PCA [dEGJL04, ZHT06, dBEG08, ZdEG10, VCLR13].

Symmetric cone programming (SCP) has been extensively studied from an algorithmic perspective, with both first-order methods [CLPV23, ZVTL24] and second-order methods [Fay97a, Fay97b, SA03, Per23] developed, as they provide valuable insights for solving semidefinite

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

programs. However, algorithms for SCP with differential privacy (DP) guarantees have received little attention, largely due to the abstract nature of the problem class. In fact, to the best of our knowledge, the only known DP algorithms in this family are for linear programs [HRRU14, DFVH+20, MMSVV21, BBDH24, BBDH25], and designing DP algorithms for SDPs was posed as a major open question in [HRRU14]. In this paper, we take a significant step forward by designing DP mechanisms not only for SDPs, but for the broader class of SCPs — a much richer and more general family of convex optimization problems. Our main contributions are two-fold: (1) We begin by examining Euclidean Jordan algebras (EJAs), a class of real inner product spaces that includes $\mathbb{R}^k$ and the space of real or complex symmetric matrices of size $r \times r$. We develop a generic Gaussian mechanism over EJAs that provides (ϵ, δ)-differential privacy guarantees when sensitivity is measured in the ℓ_1, ℓ_2, or ℓ_∞ norms. In the case of symmetric matrices, this corresponds to nuclear, Frobenius, and spectral norms, respectively. This stands in contrast to the well-known matrix mechanism [LHR+10, HT09], which aggregates multiple linear queries into a matrix and measures sensitivity column-wise in vector norms. Our framework instead captures the geometric structure of matrices by modeling perturbations to their spectrum. (2) We then design a suite of private first-order algorithms for solving SCPs under various neighboring data settings. Specifically, given two neighboring databases D and D', we consider the following cases: (i) D' has one additional linear constraint compared to D (high sensitivity constraint privacy), (ii) D and D' differ in one entry of the scalar vector b (low sensitivity scalar privacy), (iii) the constraint sets differ in ℓ_∞ norm (low sensitivity constraint privacy), and (iv) the objective elements differ (low sensitivity objective privacy). For the high sensitivity constraint setting, we show that the multiplicative weights update algorithm over constraints from [HRU13, HRRU14] naturally extends to SCP. For the remaining cases, we develop a novel variant of the multiplicative weights algorithm that operates over the primal variable space, guided by an approximate oracle that identifies the most violated constraint. This is combined with the Gaussian mechanism we previously designed for the EJA structure. As a byproduct, we design a multiplicative weights algorithm that could utilize a *noisy oracle* to for solving SCPs, which we believe could lead to applications beyond differential privacy.

Roadmap. In Section 2, we survey related work on both differential privacy and symmetric cone programming. In Section 3, we provide preliminary background on differential privacy and Euclidean Jordan algebras. Section 4 introduces a generic Gaussian mechanism for Euclidean Jordan algebras and establishes privacy guarantees under ℓ_1, ℓ_2, and ℓ_∞ sensitivities. In Section 5, we present algorithms for private symmetric cone programming, covering both high sensitivity and low sensitivity constraint privacy settings. For the high-sensitivity case, we give algorithms for solving covering semidefinite programs and covering symmetric cone programs. Section 6 concludes with a discussion of open problems and limitations of our work. Appendix A provides additional preliminaries, particularly on Euclidean Jordan algebras and regret bounds for symmetric cone multiplicative weights updates. Appendix B presents the full details, analysis, and applications for the high sensitivity constraint privacy setting. In Appendix C, we describe algorithms for the low sensitivity setting, including scalar privacy, constraint privacy, and objective privacy.

2 Related Work

Differential Privacy. Since its introduction in [DMNS06], differential privacy has become the standard notion for providing rigorous privacy guarantees in algorithm design. It has found widespread applications across general machine learning [CM08, WM10, JE19, TF20], deep neural networks [ACG+16, BPS19], computer vision [ZYCW20, LWAL21, TZXL19], natural language processing [YDW+21, WK18], federated learning [BIK+17, SYY+23, SWYZ23] and adaptive data structures [HKM+22, BKM+22, ACSS23, CSW+23, SYYZ23, FFL+25]. The line of work most relevant to ours concerns differentially private algorithms for convex programming, particularly linear programming. The work of [HRU13] was among the first to study private algorithms for zero-sum games, a special case of linear programming. Subsequently, [HRRU14] systematically examined private linear programs under various notions of sensitivity, including constraint, scalar, and objective perturbations. Their algorithms typically produce solutions that may violate constraints slightly due to privacy noise. To address this, [MMSVV21] considered the case where only the scalar vector is private, and showed that constraint satisfaction can be maintained by explicitly perturbing the scalar vector. An alternative approach was proposed in [DFVH+20], which reformulates the private problem as a stochastic chance-constrained program, whose solution satisfies the original constraints

with high probability. Building on these ideas, [BBDH24] extended the framework of [MMSVV21] to handle private constraints, and [BBDH25] further demonstrated that it is possible to privatize all components of a linear program while still satisfying the constraints. Our work is most closely related to [HRRU14], as the other approaches heavily rely on the specific structure of linear programs and extensively use the Laplace mechanism. In contrast, we show that extending these ideas to symmetric cone programming poses significantly greater challenges, both technically and algorithmically.

Euclidean Jordan Algebras and Symmetric Cone Programming. Euclidean Jordan algebras form the algebraic foundation of symmetric cone programming. It is well known that any EJA can be decomposed into a direct sum of simple Jordan algebras, and up to isomorphism, there are exactly five types of simple EJAs: real symmetric matrices, complex Hermitian matrices, quaternionic Hermitian matrices, spin factors, and the Albert algebra [FK94]. Among these, real symmetric matrices underlie semidefinite programming, while spin factors correspond to second-order cone programming. EJAs are commutative but non-associative algebras, and this structure provides key insights into the geometry and algorithmic design of semidefinite programming. As symmetric cone programming generalizes both SOCP and SDP, interior point methods have been extensively developed for this broader setting [Fay97a, Fay97b, SA03, Vie07], with significant effort devoted to abstracting and extending classical SDP techniques. In particular, [Per23] proposes a novel interior point method for symmetric cone programs using geodesic updates on the Riemannian manifold defined by the interior of the symmetric cone. Our work builds heavily on the multiplicative weights update framework introduced in [CLPV23] for online linear optimization over symmetric cones, and extended in [ZVTL24] to general symmetric cone programming. This framework is especially well suited for designing differentially private algorithms, as the oracle component required by multiplicative weights can be implemented via the exponential mechanism.

3 Preliminaries

In this section, we provide some preliminary knowledge on Euclidean Jordan algebras, symmetric cone programming and differential privacy.

3.1 Notations

We use $\widetilde{O}(f)$ to denote $O(f \operatorname{poly}\log f)$. For two real vectors $x, y \in \mathbb{R}^k$, we use $\langle x, y\rangle$ to denote $x^\top y$. We use $\|x\|_1, \|x\|_2, \|x\|_\infty$ to denote the vector ℓ_1, ℓ_2 and ℓ_∞ norms. For two real symmetric matrices $X, Y \in \mathbb{R}^{r\times r}$, we use $\langle X, Y\rangle$ to denote $\operatorname{Tr}(X^\top Y)$, where $\operatorname{Tr}(\cdot)$ is the trace of the matrix. We use $\|X\|_{S_1}, \|X\|_F$ and $\|X\|$ to denote the matrix nuclear, Frobenius and spectral norms. We use $X \succeq Y$ to denote $X - Y$ is a positive semidefinite matrix. We use $\mathcal{N}(\mu, \Sigma)$ to denote the multivariate Gaussian distribution with μ and covariance matrix Σ.

3.2 Euclidean Jordan Algebras and Symmetric Cone Programming

We will exclusively work with Euclidean Jordan algebras, which, as we will see later, generalize many important spaces, including $\mathbb{R}^k$ and the set of all $r \times r$ real symmetric matrices.

Definition 3.1 (Euclidean Jordan algebra (EJA))**.** *An Euclidean Jordan algebra (EJA) is a finite-dimensional vector space $\mathcal{J}$ equipped with:*

- *a bilinear product $\circ : \mathcal{J} \times \mathcal{J} \to \mathcal{J}$ that satisfies, for all $x, y \in \mathcal{J}$, $x \circ y = y \circ x$, and $x^2 \circ (x \circ y) = x \circ (x^2 \circ y)$;*
- *an inner product $\langle \cdot, \cdot\rangle : \mathcal{J} \times \mathcal{J} \to \mathbb{R}$ that satisfies, for all $x, y, z \in \mathcal{J}$, $\langle x \circ y, z\rangle = \langle x, y \circ z\rangle$;*
- *an identity element e that satisfies, for all $x \in \mathcal{J}$, $e \circ x = x \circ e = x$.*

As an example, if $\mathcal{J}$ is the set of all $r \times r$ real symmetric matrices, then the Jordan product is defined by $x \circ y = \frac{1}{2}(xy + yx)$, where $\cdot$ denotes standard matrix multiplication. Euclidean Jordan algebras induce a geometric structure known as a *symmetric cone*, which can be characterized as follows:

Definition 3.2 (Symmetric cone, [Vie07])**.** *A symmetric cone is a closed convex cone $\mathcal{K}$ in a finite-dimensional inner product space $\mathcal{J}$ that satisfies the following properties:*

- *$\mathcal{K}$ is self-dual, i.e., $\mathcal{K}^* := \{y \in \mathcal{J} : \langle y, x\rangle \geq 0, \forall x \in \mathcal{K}\} = \mathcal{K}$.*

- *$\mathcal{K}$ is homogeneous, i.e., for any $u, v \in \operatorname{int}(\mathcal{K})$, there exists an invertible linear transformation $L : \mathcal{J} \to \mathcal{J}$ such that $L(u) = v$ and $L(\mathcal{K}) = \mathcal{K}$.*

$\mathcal{K}$ can be characterized as the cone of squares of an EJA, specifically, there exists an EJA $\mathcal{J}$ with $\mathcal{K} = \{x^2 : x \in \mathcal{J}\}$ where $x^2 = x \circ x$ for $x \in \mathcal{J}$.

There are two key parameters associated with a Euclidean Jordan algebra (EJA) $\mathcal{J}$: its *dimension* and *rank*.

Definition 3.3 (Dimension and isomorphism)**.** *Let $\mathcal{J}$ be an EJA. Then there exists a positive integer k such that $\mathcal{J}$ is isomorphic to $\mathbb{R}^k$. This integer k is called the* dimension *of $\mathcal{J}$, denoted by $\dim(\mathcal{J}) = k$. Moreover, there exists a linear isomorphism $\phi : \mathcal{J} \to \mathbb{R}^k$ such that for all $x, y \in \mathcal{J}$, we have $\langle x, y\rangle = \langle \phi(x), \phi(y)\rangle$, i.e., ϕ is an isometry with respect to the inner product.*

As an example, consider the set of $r \times r$ real symmetric matrices. The dimension of this space is $\frac{r(r+1)}{2}$, since one can map these matrices to vectors of the same dimension. To define the rank, we first introduce a suitable spectral decomposition for any element $x \in \mathcal{J}$.

Definition 3.4 (Jordan frame)**.** *Let $q_1, \ldots, q_r \in \mathcal{J}$ be elements satisfying idempotency ($q_i^2 = q_i$) and primitiveness ($q_i \neq 0$ and cannot be written as a sum of two nonzero idempotents). We say $\{q_1, \ldots, q_r\}$ is a Jordan frame if (1) $q_i \circ q_j = 0$ for $i \neq j$; (2) $\sum_{i=1}^r q_i = e$.*

The rank of an EJA is defined as the size of a Jordan frame used to form a *spectral decomposition*.

Definition 3.5 (Spectral decomposition)**.** *For any $x \in \mathcal{J}$, there exists a set of unique real numbers $\lambda_1, \ldots, \lambda_r$ and a Jordan frame $\{q_1, \ldots, q_r\}$ such that $x = \sum_{i=1}^r \lambda_i q_i$. The minimal such r is called the rank of $\mathcal{J}$ (denoted by $\operatorname{rank}(\mathcal{J}) = r$). We use $\operatorname{Tr}(x) = \sum_{i=1}^r \lambda_i$ to denote the trace of the element x. We define the exponentiation function $\exp : \mathcal{J} \to \mathcal{J}$ as $\exp\left(\sum_{i=1}^r \lambda_i q_i\right) := \sum_{i=1}^r \exp(\lambda_i) q_i$.*

Returning to the example of the set of $r \times r$ real symmetric matrices, we observe that the spectral decomposition coincides with the standard matrix spectral decomposition. Therefore, the rank of this space is r. Note the quadratic gap between the rank and the dimension for this example. The final concept we introduce here is the trace-based inner product and the associated family of norms.

Definition 3.6 (Inner product and norm)**.** *Given $x, y \in \mathcal{J}$, we define their inner product as $\langle x, y\rangle = \operatorname{Tr}(x \circ y)$. For any $p \in [1, \infty)$, the ℓ_p norm of x is defined as $\|x\|_p = \left(\sum_{i=1}^r |\lambda_i(x)|^p\right)^{1/p}$, and for $p = \infty$, the ℓ_∞ norm is defined as $\|x\|_\infty = \max_{i \in [r]} |\lambda_i(x)|$.*

Standard inequalities related to the inner product, such as Cauchy–Schwarz and Hölder's inequality, continue to hold in this setting; we defer their proofs to Appendix A. We are now ready to define symmetric cone programming.

Definition 3.7 (Symmetric cone programming)**.** *Given a collection of elements $a_1, \ldots, a_m, c \in \mathcal{J}$ and $b \in \mathbb{R}^m$, the symmetric cone program (SCP) is defined as*

$$\begin{aligned} &\max \langle c, x\rangle \\ &\text{s.t. } \langle a_i, x\rangle \leq b_i, \quad \forall i \in [m], \\ &\quad x \succeq_{\mathcal{K}} 0, \end{aligned}$$

where $\succeq_{\mathcal{K}}$ denotes the generalized cone inequality, i.e., $x \succeq_{\mathcal{K}} y$ means $x - y \in \mathcal{K}$.

3.3 Differential Privacy

Differential privacy is a strong and rigorous notion of privacy, first introduced in [DMNS06]. We state its formal definition below.

Definition 3.8 (Differential privacy, [DMNS06])**.** *A randomized mechanism $\mathcal{M} : \mathcal{D} \to \mathcal{R}$ is said to provide (ϵ, δ)-differential privacy if, for every pair of neighboring databases $D, D' \in \mathcal{D}$ (differing in exactly one record) and every measurable subset $S \subseteq \mathcal{R}$,*

$$\Pr[\mathcal{M}(D) \in S] \leq e^\epsilon \Pr[\mathcal{M}(D') \in S] + \delta.$$

Here $\epsilon > 0$ and $0 \leq \delta < 1$ are privacy parameters. The special case $\delta = 0$ is referred to as ϵ-differential privacy.

We will make use of the standard Gaussian mechanism.

Definition 3.9 (Gaussian mechanism [DKM+06]). *Let $\epsilon, \delta > 0$. A mapping $f : \mathcal{D} \to \mathbb{R}^k$ has ℓ_2-sensitivity Δ_2 if for all neighboring databases D, D' differing in one record, $\|f(D) - f(D')\|_2 \leq \Delta_2$. The Gaussian mechanism releases $\mathcal{M}(D) = f(D) + \nu$, where $\nu \sim \mathcal{N}(0, \sigma^2 I_k)$ and $\sigma = \frac{\Delta_2\sqrt{2\log(1.25/\delta)}}{\epsilon}$. This mechanism satisfies (ϵ, δ)-differential privacy.*

We note that [BW18] has improved the parameters of Gaussian mechanism, in particular the dependence on ϵ, δ for σ. They further extend the analysis to work for all ranges of ϵ. In this work, we focus on providing a preliminary privacy analysis of EJA, thus we adopt the more traditional and simple bound for Gaussian mechanisms.

To analyze the utility of the Gaussian mechanism, it is convenient to recall a standard tail bound for Gaussian random vectors.

Lemma 3.10 (Laurent and Massart [LM00]). *Let $Z \sim \chi^2_k$ be a chi-squared random variable with k degrees of freedom. Suppose each component has zero mean and variance σ^2. Then:*

$$\Pr[Z - k\sigma^2 \geq (2\sqrt{kt} + 2t)\sigma^2] \leq \exp(-t),$$
$$\Pr[k\sigma^2 - Z \geq 2\sqrt{kt}\sigma^2] \leq \exp(-t).$$

We also make use of the exponential mechanism [MT07], a tool for achieving privacy when the output lies in a discrete or non-numeric domain. This mechanism relies on a *quality score*, a real-valued function that evaluates the utility of pairing a database with a candidate output. Given a database, the exponential mechanism then selects, in a privacy-preserving way, an element whose quality score is close to the maximum achievable value.

Definition 3.11 (Exponential mechanism, [MT07]). *For a database D and range $\mathcal{R}$, the exponential mechanism chooses $r \in \mathcal{R}$ with probability proportional to $\exp\left(\frac{\epsilon}{2\Delta}Q(r, D)\right)$. Here $Q : \mathcal{R} \times \mathcal{D} \to \mathbb{R}$ is a quality score function, and Δ denotes its sensitivity with respect to neighboring databases. This mechanism is ϵ-differential privacy.*

The exponential mechanism satisfies the following utility guarantee.

Lemma 3.12 (Accuracy guarantee for exponential mechanism, [KPRU14]). *Given $\beta \in (0, 1)$ and database D, let OPT denote the maximum value of the quality score Q can attain on database D. Then, with an ϵ-private exponential mechanism with quality score on database D and outputs satisfies that, with probability at least $1 - \beta$, $Q(r, D) \geq \mathrm{OPT} - \frac{2\Delta}{\epsilon}\log\left(\frac{|\mathcal{R}|}{\beta}\right)$.*

Finally, we recall a standard composition tool to combine private mechanisms.

Lemma 3.13 ([DRV10]). *Let $\delta \in (0, 1)$, and $\mathcal{M}_1, \ldots, \mathcal{M}_k$ be ϵ'-private, adaptively chosen mechanisms, then the composition $\mathcal{M}_1 \circ \ldots \circ \mathcal{M}_k$ is (ϵ, δ)-private, provided that $\epsilon' = \frac{\epsilon}{\sqrt{8k\log(1/\delta)}}$.*

We remark that while adaptive composition has been improved via moment accountant [ACG+16] and Rényi DP [Mir17, WBK19], we use the simpler adaptive composition for illustrating the main idea.

4 A DP Framework for Euclidean Jordan Algebras

In this section, we introduce a differential privacy (DP) framework for Euclidean Jordan algebras (EJA), based on the Gaussian mechanism. We begin with a motivating example. Let $\mathcal{S}^r$ denote the set of all $r \times r$ real symmetric matrices. Suppose we are designing a private recommendation system, where each client record is represented by an r-dimensional feature vector u. The goal is to privately release the covariance matrix of the dataset. In this setting, two neighboring databases D, D' differ in exactly one feature vector, and the function $f : \mathcal{D} \to \mathcal{S}^r$ outputs the covariance matrix of the data. It is then natural to define sensitivity using the *matrix Schatten-p norms* rather than entrywise norms. For instance, the ℓ_∞ sensitivity corresponds to the spectral norm $\|f(D) - f(D')\|$, while the ℓ_1 sensitivity corresponds to the nuclear norm $\|f(D) - f(D')\|_{S_1}$. These spectrum-aware notions of sensitivity highlight the inadequacy of mechanisms like the Laplace mechanism, which adds noise entrywise, and motivate the need for a more principled approach. This leads us to consider a general formulation of differential privacy in the context of EJAs, which we refer to as DP-EJA.

Definition 4.1. *Let $\mathcal{D}$ be the universe of databases, and let $f : \mathcal{D} \to \mathcal{J}$. For neighboring databases D, D', we define:*

- *f has ℓ_1 sensitivity Δ_1 if $\|f(D) - f(D')\|_1 \leq \Delta_1$;*
- *f has ℓ_2 sensitivity Δ_2 if $\|f(D) - f(D')\|_2 \leq \Delta_2$;*
- *f has ℓ_∞ sensitivity Δ_∞ if $\|f(D) - f(D')\|_\infty \leq \Delta_\infty$.*

Perhaps the first question one might ask for DP-EJA is: how do we define an additive noise mechanism? It is not immediately clear what constitutes a "Gaussian element" in an EJA $\mathcal{J}$. Thus our first order of business is to develop a Gaussian mechanism for EJAs.

Lemma 4.2 (Generic Gaussian mechanism)**.** *Let $\mathcal{J}$ be an EJA with $\dim(\mathcal{J}) = k$, equipped with an isometry $\phi : \mathcal{J} \to \mathbb{R}^k$. Let $\epsilon, \delta > 0$ and let $f : \mathcal{D} \to \mathcal{J}$ be a function with ℓ_2 sensitivity Δ_2. Consider the following mechanism:*

- *Set $\sigma = \frac{\Delta_2 \sqrt{2 \log(1.25/\delta)}}{\epsilon}$;*
- *Generate a Gaussian noise vector $\nu \sim \mathcal{N}(0, \sigma^2 I_k)$;*
- *Set $z = \phi^{-1}(\nu)$.*

The mechanism releases $f(D) + z$ and is (ϵ, δ)-differentially private.

Proof. The proof is straightforward. Since ϕ is an isometry, we have that for any $x, y \in \mathcal{J}$, $\langle \phi(x), \phi(y) \rangle = \langle x, y \rangle$, which implies that $\|x\|_2 = \|\phi(x)\|_2$. Therefore, the ℓ_2 sensitivity of f satisfies $\|f(D) - f(D')\|_2 = \|\phi(f(D)) - \phi(f(D'))\|_2$. By Definition 3.9, the Gaussian mechanism $(\phi \circ f)(D) + \nu$ is (ϵ, δ)-differentially private. Since ϕ^{-1} is also an isometry, we have $\phi^{-1}((\phi \circ f)(D) + \nu) = f(D) + \phi^{-1}(\nu) = f(D) + z$, which implies that the release $f(D) + z$ is (ϵ, δ)-private. □

Lemma 4.2 provides a simple method for constructing a Gaussian element in $\mathcal{J}$: choose an isometry ϕ, generate a Gaussian vector in $\mathbb{R}^k$, and then apply ϕ^{-1}. Such an isometry always exists, as one can construct it using an orthonormal basis for $\mathcal{J}$. As an example, consider $\mathcal{J} = \mathcal{S}^r$, the space of $r \times r$ real symmetric matrices. Then $\dim(\mathcal{J}) = k = \frac{r(r+1)}{2}$, and the isometry ϕ maps a symmetric matrix to a vector with the entries from its upper triangular part. Lemma 4.2 in this case corresponds to generating a Gaussian vector in $\mathbb{R}^k$, and then applying ϕ^{-1} to obtain a symmetric matrix with the upper triangular part filled by the Gaussian vector. The ℓ_2 norm on $\mathcal{S}^r$ corresponds to the Frobenius norm, so it is easy to see that Lemma 4.2 recovers the standard Gaussian mechanism for matrix-valued functions under Frobenius norm sensitivity. As a consequence, we obtain private mechanisms for ℓ_1 and ℓ_∞ sensitivity via norm inequalities:

Corollary 4.3. *Let $\mathcal{J}$ be an EJA with $\dim(\mathcal{J}) = k$, equipped with an isometry $\phi : \mathcal{J} \to \mathbb{R}^k$. Let $\epsilon, \delta > 0$, and let $f : \mathcal{D} \to \mathcal{J}$ have ℓ_1 sensitivity Δ_1. Then, the generic Gaussian mechanism (Lemma 4.2) with sensitivity parameter Δ_1 is (ϵ, δ)-differentially private.*

Corollary 4.4. *Let $\mathcal{J}$ be an EJA with $\dim(\mathcal{J}) = k$, equipped with an isometry $\phi : \mathcal{J} \to \mathbb{R}^k$. Let $\epsilon, \delta > 0$, and let $f : \mathcal{D} \to \mathcal{J}$ have ℓ_∞ sensitivity Δ_∞. Then, the generic Gaussian mechanism (Lemma 4.2) with sensitivity parameter $\sqrt{r}\Delta_\infty$ is (ϵ, δ)-differentially private.*

One might wonder whether it is possible to perturb only the eigenvalues instead of all k dimensions. This idea is particularly tempting for $\mathcal{S}^r$, since $k = O(r^2)$ while the rank is only r. A naïve approach would be to first compute a spectral decomposition $f(D) = \sum_{i=1}^r \lambda_i q_i$, then inject scalar Gaussian noise to each eigenvalue, yielding $\sum_{i=1}^r (\lambda_i + \nu_i) q_i$. This approach also appears attractive for handling ℓ_1 sensitivity by injecting Laplace noise to each eigenvalue. Unfortunately, this method fails to ensure differential privacy. It is not sufficient to perturb only the eigenvalues — the Jordan frame must also be randomized. This can sometimes be achieved by sampling a random Jordan frame $\{p_1, \ldots, p_r\}$ and outputting $\sum_{i=1}^r (\lambda_i + \nu_i) p_i$. However, this raises challenges in bounding the ℓ_2 norm of the result, as one must account for differences between Jordan frames. In the special case $\mathcal{J} = \mathbb{R}^k$, this issue disappears since there is a unique Jordan frame $\{e_1, \ldots, e_k\}$ and rank coincides with dimension. For ℓ_1 sensitivity, one might wish to obtain a pure ϵ-private mechanism using the Laplace mechanism: namely, sample a Laplace noise vector in $\mathbb{R}^k$ with suitable parameters and map it into $\mathcal{J}$ via ϕ^{-1}. Unfortunately, this also fails: the ℓ_1 norm in $\mathbb{R}^k$ does not, in general, correspond to the ℓ_1 norm in $\mathcal{J}$ (except when $\mathcal{J} = \mathbb{R}^k$). In essence, the isometry ϕ only preserves the inner product — and hence

the ℓ_2 norm — but not other norms. For this reason, our mechanisms for ℓ_1 and ℓ_∞ sensitivity are derived from the generic Gaussian mechanism.

We also note that the utility guarantee for the generic Gaussian mechanism follows from the isometry property of ϕ: the ℓ_2 norm of the noise element $z \in \mathcal{J}$ equals the ℓ_2 norm of the Gaussian vector $\nu \in \mathbb{R}^k$, which scales as $\sqrt{k}$ due to Lemma 3.10. This implies a weaker bound when translating to ℓ_∞ norm. For $\mathcal{S}^r$, standard results in random matrix theory state that $\|z\|_\infty = O(\sqrt{r})$ [Wig58]. For general EJAs, it is well-known that they are direct sums of five simple EJAs [FK94], and if it does not has spin factor as its component, then $\|z\|_\infty = O(\sqrt{r_{\max}})$ where $r_{\max}$ is the largest rank among its components, and is $O(\sqrt{k_{\max}} + \sqrt{r_{\max}})$ if it has spin factor as its component and $k_{\max}$ is the largest dimension. To unify the discussion, we conservatively adopt the weaker $O(\sqrt{k})$ bound in this work.

5 Private Symmetric Cone Programming

In this section, we develop differentially private algorithms for solving symmetric cone programs. As an application, we obtain private algorithms for semidefinite programming, resolving a major open question posed in [HRRU14]. Following the framework of [HRRU14] for linear programming, we study private algorithms under several settings: (1) *High sensitivity constraint privacy*, where neighboring databases may differ by one entire constraint; (2) *Low sensitivity constraint privacy*, where all databases have the same number of constraints, and neighboring databases differ in the ℓ_∞ norm of those constraints; (3) *Scalar privacy*, where neighboring databases differ in the right-hand side vector b, again measured in ℓ_∞ norm; (4) *Objective privacy*, where neighboring instances differ in the objective element c under ℓ_∞ norm. Our algorithm for the high sensitivity constraint setting is a generalization of the dense multiplicative weights update (MWU) method used in [HRRU14] for linear programs. For the other three settings, we adopt the MWU framework for symmetric cone programming introduced in [CLPV23, ZVTL24], but propose a novel scheme in which the update direction is determined by identifying the most violated constraint.

5.1 High Sensitivity Constraint Privacy

In this setting, the SCP instances for two neighboring databases share the same objective element $c \in \mathcal{J}$ but differ by one constraint and its corresponding scalar value: they have the same first m constraints, while D' contains an additional constraint and scalar value b'_{m+1}. Note that this additional constraint can be arbitrary. As first observed in [HRRU14], it is generally impossible to design a private algorithm that satisfies *all* constraints, as a new constraint can significantly alter the optimal solution of the original program. The key idea in [HRRU14] is to run an MWU procedure over a restricted set of constraints, so that the output solution satisfies most of the constraints while preserving privacy. This is achieved by applying Bregman projection [HW01] and performing MWU over a projected dense distribution over the constraints. While [HRRU14] analyzes this method in the context of linear programming, we show that it extends naturally to symmetric cone programming, since the algorithm operates over constraints. As a consequence, we obtain private algorithms for covering semidefinite programs that are especially useful in regimes where the number of constraints m is much larger than the matrix dimension r. Specifically, consider the following covering SDP:

$$\begin{aligned} \min_{X \succeq 0} \ & \operatorname{Tr}(X) \\ \text{s.t. } & \langle A_i, X \rangle \geq 1, \quad \forall i \in [m], \end{aligned}$$

where $A_1, \ldots, A_m \succeq 0$ and $\max_{i \in [m]} \|A_i\| \leq 1$. All matrices are of size $r \times r$. Covering SDPs have a wide range of applications in machine learning, including robust mean estimation [CDG19], robust covariance estimation [CDGW19], and E-optimal experimental design [vAG19].

Theorem 5.1 (Informal version of Theorem B.10)**.** *Let $\epsilon > 0$, $\delta \in (0,1)$ be the DP parameters, and let $\beta \in (0,1)$ be the failure probability. Given a covering SDP with m constraints over $r \times r$ matrices, there exists an algorithm (Algorithm 3) that finds $X^* \succeq 0$ such that $\langle A_i, X^* \rangle \geq 1 - \alpha$ for all but s constraints, with probability at least $1 - \beta$. s and α satisfy*

$$s = \Omega\left(\frac{r}{\epsilon} \log^{1/2}(1/\delta) \log(1/\beta) \log m\right), \quad \alpha = O(\mathrm{OPT}).$$

Moreover, the algorithm is ϵ-differentially private with respect to high sensitivity constraint privacy.

The core idea behind Theorem 5.1 is to use the dense MWU framework described above, paired with a private oracle that performs a simple linear minimization. To ensure privacy, the oracle is implemented using the exponential mechanism. A major technical challenge is that, even when the optimal value of the program is fixed to OPT, the number of feasible solutions is infinite, as they correspond to extreme rays of the positive semidefinite cone intersected with a hyperplane. A naïve application of Lemma 3.12 would yield a vacuous bound due to $|\mathcal{R}| = \infty$. To overcome this, we use a γ-net argument to discretize the space of feasible solutions. By carefully choosing γ, we ensure the size of the net is $\exp(r)$, which allows us to apply Lemma 3.12 with a penalty factor of r. This contrasts with the private covering LP algorithm in [HRRU14], where the feasible solutions are simply scaled standard basis vectors in $\mathbb{R}^r$, and hence $|\mathcal{R}| = r$, resulting in no such dependence on r.

Inspired by our private algorithm for covering SDP, we further extend the framework to any covering *symmetric cone program* over a simple Euclidean Jordan algebra:

$$\begin{aligned} &\min_{x \in \mathcal{K}} \ \mathrm{Tr}(x) \\ &\text{s.t. } \langle a_i, x \rangle \geq 1, \quad \forall i \in [m]. \end{aligned}$$

It is well-known that, up to isomorphism, there are five types of simple Jordan algebras: $r \times r$ real symmetric matrices, $r \times r$ complex Hermitian matrices, $r \times r$ quaternionic Hermitian matrices, r-dimensional spin factors, and the exceptional Albert algebra [FK94]. In particular, the cone of squares for r-dimensional spin factors corresponds to the r-dimensional second-order cone. To apply the machinery developed for private covering SDP to general SCP, we observe that the oracle's optimal solutions are (scaled) primitive idempotents in $\mathcal{J}$. Therefore, a γ-net argument requires bounding the dimension of the set of primitive idempotents. For all simple Jordan algebras, this dimension is $O(r)$ [FK94], which mirrors the dimension in the SDP case where $\mathcal{J} = \mathcal{S}^r$. To the best of our knowledge, this is the first attempt to quantize the rays of primitive idempotents via a γ-net, and we hope this approach enables further applications and tighter bounds in future work.

Theorem 5.2 (Informal version of Theorem B.13)**.** *Let $\mathcal{J}$ be a simple EJA of rank r. Let $\epsilon > 0$, $\delta \in (0, 1)$ be the DP parameters, and let $\beta \in (0, 1)$ be the failure probability. Given a covering SCP with m linear constraints, there exists an algorithm (Algorithm 3) that outputs a point $x^* \in \mathcal{K}$ such that $\langle a_i, x^* \rangle \geq 1 - \alpha$ for all but s constraints, with probability at least $1 - \beta$. The parameters s and α satisfy*

$$s = \Omega\left(\frac{r}{\epsilon} \log^{1/2}(1/\delta) \log(1/\beta) \log m\right), \quad \alpha = O(\mathrm{OPT}).$$

Moreover, the algorithm is ϵ-differentially private with respect to high sensitivity constraint privacy.

5.2 Low Sensitivity Constraint Privacy

In the low sensitivity setting, two neighboring databases have the same number of constraints and differ in the ℓ_∞ norm. Specifically, let $A(D), A(D')$ be the constraint sets corresponding to neighboring databases D and D', respectively. We assume that $\max_{i \in [m]} \|a_i(D) - a_i(D')\|_\infty \leq \Delta_\infty$. As observed in [HRRU14] for the LP setting, it is possible to approximately satisfy all constraints by applying MWU over the *variables*. In particular, the oracle in this setting is given the constraint set $A \in \mathcal{J}^m$, scalar vector $b \in \mathbb{R}^m$, and a point $x \in \mathcal{K}$, and it must return an approximately most violated constraint a_i such that $\langle a_i, x \rangle - b_i \geq \max_{j \in [m]} \langle a_j, x \rangle - b_j - \alpha$ with high probability. We refer to such an oracle as an (α, γ)-*dual oracle*, which achieves additive approximation α and fails with probability at most γ. This oracle is particularly well-suited to privatization via the exponential mechanism: the score function $Q(i, b) = \langle a_i, x \rangle - b_i$ leads to both privacy and accuracy guarantees. It remains to show that one can indeed solve the SCP in a first-order fashion using such an oracle. In the case of linear programming, this is already established by the classical work of [PST95]. Our first result shows that an approximate most violated constraint oracle can likewise be used to solve SCPs over symmetric cones in a first-order manner. Before stating the result, we define the *width* of the constraint set as $\rho = \max_{i \in [m]} \|a_i\|_\infty$.

Theorem 5.3 (Informal version of Theorem C.2)**.** *Given an SCP with m linear constraints, suppose there exists a feasible point $x \in \mathcal{K}$ with $\mathrm{Tr}(x) = 1$, and access to an $(\alpha/2, \gamma)$-dual oracle. Then, there exists an algorithm (Algorithm 5) that finds a distribution element $x^* \in \mathcal{K}$ such that $\langle a_i, x^* \rangle \leq b_i + \alpha$ for all $i \in [m]$, with probability at least $1 - T\gamma$, where $T = O\left(\frac{\rho^2 \log r}{\alpha^2}\right)$.*

The assumption on the existence of x with $\mathrm{Tr}(x) = 1$ is without loss of generality, as it can always be satisfied by scaling down any optimal solution by its ℓ_1 norm. Previous MWU algorithms for symmetric cone programming have relied on a *primal oracle*, where the input is a point $x \in \mathcal{K}$ with $\mathrm{Tr}(x) = 1$, and the output is a vector $y \in \mathbb{R}^m$ satisfying $\langle \sum_i y_i a_i - c, x \rangle \geq 0$ and $b^\top y \leq \alpha$ [ZVTL24]. Our proof instead builds on the regret bound for MWU over EJAs as established in [CLPV23].

Algorithm 1 Constraint private SCP solver.

procedure CONSTRAINTPRIVATESCP($A \in \mathcal{J}^m, b \in \mathbb{R}^m$)
  $x^1 \leftarrow e/r$
  Let α, γ be the parameters for the oracle, ϵ, δ be the parameters for DP
  Let ORACLE be an (α, γ)-dual oracle
  $T \leftarrow \frac{144 \log r}{\alpha^2}, \epsilon' \leftarrow \frac{\epsilon}{4\sqrt{T \log(1/\delta)}}, \eta \leftarrow \frac{\alpha}{12\rho}$
  for $t = 1 \to T$ **do**
    $\sigma \leftarrow \frac{\Delta_\infty \sqrt{2r \log(T/\delta)}}{\epsilon'}$
    $p^t \leftarrow \textsc{Oracle}(A, b, x^t)$
    $\nu^t \sim \mathcal{N}(0, \sigma^2 I_k)$
    $z^t \leftarrow \phi^{-1}(\nu^t)$
    $\widehat{\ell^t} \leftarrow \frac{a_{p^t} + z^t}{2}$
    $x^{t+1} \leftarrow \frac{\exp(-\sum_{i=1}^t \eta \widehat{\ell^i})}{\mathrm{Tr}(\exp(-\sum_{i=1}^t \eta \widehat{\ell^i}))}$
  end for
  return $\overline{x} \leftarrow \frac{1}{T} \sum_{t=1}^T x^t$
end procedure

We observe that the private data A is accessed both during the oracle step and the loss computation. To ensure privacy in the oracle step, we use the exponential mechanism; for the loss computation, we apply the generic Gaussian mechanism developed in Section 4, injecting a Gaussian noise into the constraint returned by the oracle.

Theorem 5.4 (Informal version of Theorem C.7 and C.9)**.** *Let $A \in \mathcal{J}^m$ satisfy $\lambda(a_i) \subseteq [-1, 1]$ for all $i \in [m]$, and let $b \in \mathbb{R}^m$. Let $\beta, \epsilon > 0$ and $\delta \in (0, 1)$. Then Algorithm 1 using the exponential mechanism to implement a dual oracle returns a distributional element x^* such that with probability at least $1 - \beta$, $\langle a_i, x^* \rangle \leq b_i + \alpha$ for all $i \in [m]$, where*

$$\alpha = \widetilde{O}\left(\frac{\Delta_\infty^{1/2} r^{1/4} k^{1/4}}{\epsilon^{1/2}} \cdot \mathrm{polylog}(r, 1/\beta, 1/\delta) \right).$$

Moreover, the algorithm is (ϵ, δ)-private with respect to low sensitivity constraint privacy.

Compared to the low sensitivity constraint privacy LP result of [HRRU14], our bound incurs an additional $k^{1/4}$ factor in α. This arises because the sensitivity is measured in the ℓ_∞ norm, yet the noise added is a k-dimensional Gaussian vector. By standard concentration bounds on the norm of Gaussian vectors (see Lemma 3.10), the ℓ_2 norm of the Gaussian vector — and consequently, the corresponding Gaussian element z — scales with $\sqrt{k}$. In contrast, [HRRU14] adds independent entrywise Laplace noise, and the magnitude of any single perturbation is at most Δ_∞/ϵ with high probability. As discussed in Section 4, our framework does not permit injecting noise directly into the eigenvalues of the EJA elements, since this would require sampling a random Jordan frame as well — posing additional complexity and potential distortion. We also develop algorithms for the setting where the scalar vector $b \in \mathbb{R}^m$ or the objective element $c \in \mathcal{J}$ is private under low sensitivity. These follow as variants of the MWU algorithm developed for low sensitivity constraint privacy SCP. We defer the details to Appendix C.

Remark 5.5. *We interpret Algorithm 1 as an approximate multiplicative weights update (MWU) scheme with noisy oracles. Instead of applying the MWU rule directly to the primal variables, the algorithm first perturbs the oracle output with Gaussian noise and then uses the perturbed value for the update. Theorem 5.4 can thus be viewed as quantifying how the injected noise influences constraint violations. More broadly, by altering the noise distribution, one can adapt the noisy MWU framework to other settings that require different types of error guarantees.*

6 Conclusion

We study differentially private algorithms for convex programming, with a particular focus on symmetric cone programming. To this end, we develop a generic Gaussian mechanism for Euclidean Jordan algebras that provides differential privacy guarantees under ℓ_1, ℓ_2, and ℓ_∞ norms. We incorporate this mechanism into private solvers for symmetric cone programs under both high and low sensitivity settings. For the high sensitivity regime, we generalize the analysis of [HRRU14] beyond linear programming and apply it to covering semidefinite programs and covering symmetric cone programs. In the low sensitivity setting, we design a private solver based on an approximately most violated constraint oracle, in conjunction with our generic Gaussian mechanism. As a direct consequence, we obtain a family of differentially private algorithms for semidefinite programming — a longstanding open problem originally posed by [HRRU14] — that also encompass a wide range of applications in machine learning.

There are several limitations of our work, which we leave as directions for future research. (1) The differential privacy mechanisms we employ are relatively basic; recent advances in moment accounting and Rényi DP [ACG+16, Mir17, WBK19] could yield stronger trade-offs between privacy and utility. (2) Although our private solvers provide meaningful guarantees, they only approximately satisfy the constraints. In high sensitivity settings, this manifests as a small number of constraint violations. A central open question is whether it is possible to design private algorithms that satisfy all constraints exactly. For linear programming, [MMSVV21, BBDH24, BBDH25] develop techniques to preserve feasibility under privacy. Extending these methods to the broader setting of symmetric cone programming (and even SDPs) is considerably more challenging, since one must contend with privacy-preserving perturbations to the *spectrum* of matrices rather than to entries of a vector.

Acknowledgment

We thank anonymous NeurIPS reviewers for their constructive comments. Lichen Zhang is supported by a Mathworks Fellowship and a Simons Dissertation Fellowship in Mathematics.

References

[ACG+16] Martín Abadi, Andy Chu, Ian Goodfellow, H. Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security (CCS)*, pages 308–318. ACM, 2016.

[ACSS23] Idan Attias, Edith Cohen, Moshe Shechner, and Uri Stemmer. A framework for adversarial streaming via differential privacy and difference estimators. In Yael Tauman Kalai, editor, *14th Innovations in Theoretical Computer Science (ITCS 2023)*, volume 251 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 8:1–8:19. Schloss Dagstuhl — Leibniz-Zentrum für Informatik, 2023.

[ATMR21] Galen Andrew, Om Thakkar, H. Brendan McMahan, and Swaroop Ramaswamy. Differentially private learning with adaptive clipping. In *Advances in Neural Information Processing Systems*, volume 34, pages 17455–17466, 2021.

[AZLSW20] Zeyuan Allen-Zhu, Yuanzhi Li, Aarti Singh, and Yining Wang. Near-optimal discrete optimization for experimental design: A regret minimization approach. *Mathematical Programming*, pages 1–40, 2020.

[BBDH24] Alexander Benvenuti, Brendan Bialy, Miriam Dennis, and Matthew Hale. Guaranteed feasibility in differentially private linearly constrained convex optimization. *IEEE Control Systems Letters*, 8:2745–2750, 2024.

[BBDH25] Alexander Benvenuti, Brendan Bialy, Miriam Dennis, and Matthew Hale. Differentially private linear programming: Reduced sub-optimality and guaranteed constraint satisfaction. *arXiv preprint arXiv:2501.19315*, 2025.

[BGV92] Bernhard E. Boser, Isabelle M. Guyon, and Vladimir N. Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, COLT '92, 1992.

[BIK+17] Keith Bonawitz, Vladimir Ivanov, Ben Kreuter, Antonio Marcedone, H. Brendan McMahan, Sarvar Patel, Daniel Ramage, Aaron Segal, and Karn Seth. Practical secure aggregation for privacy-preserving machine learning. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security (CCS)*, pages 1175–1191. ACM, 2017.

[Bir46] George D. Birkhoff. Tres observaciones sobre el álgebra lineal. *Revista A de la Universidad Nacional de Tucumán, Serie A*, pages 147–151, 1946.

[BKM+22] Amos Beimel, Haim Kaplan, Yishay Mansour, Kobbi Nissim, Thatchaphol Saranurak, and Uri Stemmer. Dynamic algorithms against an adaptive adversary: Generic constructions and lower bounds. In *Proceedings of the 54th Annual ACM SIGACT Symposium on Theory of Computing*, pages 1671–1684, 2022.

[BPS19] Eugene Bagdasaryan, Omid Poursaeed, and Vitaly Shmatikov. Differential privacy has disparate impact on model accuracy. *Advances in Neural Information Processing Systems (NeurIPS)*, 32:15479–15488, 2019.

[BST14] Raef Bassily, Adam Smith, and Abhradeep Thakurta. Private empirical risk minimization: Efficient algorithms and tight error bounds. In *Proceedings of the 55th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 464–473. IEEE, 2014.

[BW18] Borja Balle and Yu-Xiang Wang. Improving the gaussian mechanism for differential privacy: Analytical calibration and optimal denoising. In *Proceedings of the 35th International Conference on Machine Learning (ICML 2018)*, volume 80 of *Proceedings of Machine Learning Research*, pages 403–412. PMLR, 2018.

[CDG19] 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 (SODA)*, pages 2755–2771. SIAM, 2019.

[CDGW19] Yu Cheng, Ilias Diakonikolas, Rong Ge, and David P. Woodruff. Faster algorithms for high-dimensional robust covariance estimation. In *Proceedings of the 32nd Conference on Learning Theory (COLT)*, volume 99 of *Proceedings of Machine Learning Research*, pages 1–31. PMLR, 2019.

[CL11] Chih-Chung Chang and Chih-Jen Lin. Libsvm: A library for support vector machines. *ACM Trans. Intell. Syst. Technol.*, may 2011.

[CLPV23] Ilayda Canyakmaz, Wayne Lin, Georgios Piliouras, and Antonios Varvitsiotis. Multiplicative updates for online convex optimization over symmetric cones, 2023.

[CM08] Kamalika Chaudhuri and Claire Monteleoni. Privacy-preserving logistic regression. In *NIPS*, volume 8, pages 289–296. Citeseer, 2008.

[CR12] Emmanuel Candès and Benjamin Recht. Exact matrix completion via convex optimization. *Commun. ACM*, 2012.

[CSW+23] Yeshwanth Cherapanamjeri, Sandeep Silwal, David Woodruff, Fred Zhang, Qiuyi Zhang, and Samson Zhou. Robust algorithms on adaptive inputs from bounded adversaries. In *The Eleventh International Conference on Learning Representations*, 2023.

[CT10] Emmanuel J. Candès and Terence Tao. The power of convex relaxation: Near-optimal matrix completion. *IEEE Trans. Inf. Theor.*, 2010.

[CV95] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Mach. Learn.*, 1995.

[dBEG08] Alexandre d'Aspremont, Francis Bach, and Laurent El Ghaoui. Optimal solutions for sparse principal component analysis. *Journal of Machine Learning Research*, 9:1269–1294, 2008.

[dEGJL04] Alexandre d'Aspremont, Laurent El Ghaoui, Michael I. Jordan, and Gert R. G. Lanckriet. A direct formulation for sparse pca using semidefinite programming. In *Advances in Neural Information Processing Systems*, volume 17, pages 41–48. MIT Press, 2004.

[DFVH+20] Vladimir Dvorkin, Ferdinando Fioretto, Pascal Van Hentenryck, Jalal Kazempour, and Pierre Pinson. Differentially private convex optimization with feasibility guarantees. *arXiv preprint arXiv:2006.12338*, 2020.

[DKM+06] Cynthia Dwork, Krishnaram Kenthapadi, Frank McSherry, Ilya Mironov, and Moni Naor. Our data, ourselves: Privacy via distributed noise generation. In *Advances in Cryptology - EUROCRYPT 2006, 25th Annual International Conference on the Theory and Applications of Cryptographic Techniques*, volume 4004 of *Lecture Notes in Computer Science*, pages 486–503. Springer, 2006.

[DMNS06] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of Cryptography: Third Theory of Cryptography Conference, TCC 2006, New York, NY, USA, March 4-7, 2006. Proceedings 3*, pages 265–284. Springer, 2006.

[DRV10] Cynthia Dwork, Guy N Rothblum, and Salil Vadhan. Boosting and differential privacy. In *2010 IEEE 51st annual symposium on foundations of computer science*, pages 51–60. IEEE, 2010.

[Fay97a] Leonid Faybusovich. Euclidean jordan algebras and interior-point algorithms. *Positivity*, 1(4):331–357, 1997.

[Fay97b] Leonid Faybusovich. Linear systems in jordan algebras and primal-dual interior-point algorithms. *Journal of Computational and Applied Mathematics*, 86(1):149–175, 1997.

[FFL+25] Shiyuan Feng, Ying Feng, George Z. Li, Zhao Song, David P. Woodruff, and Lichen Zhang. On differential privacy for adaptively solving search problems via sketching. In *International Conference on Machine Learning (ICML)*. PMLR, 2025.

[FHL+24] Jie Fu, Yuan Hong, Xinpeng Ling, Leixia Wang, Xun Ran, Zhiyu Sun, Wendy Hui Wang, Zhili Chen, and Yang Cao. Differentially private federated learning: A systematic review. *arXiv preprint arXiv:2405.08299*, 2024.

[FK94] Jacques Faraut and Adam Korányi. *Analysis on Symmetric Cones*. Oxford Mathematical Monographs. Clarendon Press, Oxford University Press, 1994.

[GKN17] Robin C. Geyer, Tassilo Klein, and Moin Nabi. Differentially private federated learning: A client level perspective. *arXiv preprint arXiv:1712.07557*, 2017.

[GLZ17] David Gamarnik, Quan Li, and Hongyi Zhang. Matrix completion from $O(n)$ samples in linear time. In Satyen Kale and Ohad Shamir, editors, *Proceedings of the 2017 Conference on Learning Theory*, volume 65 of *Proceedings of Machine Learning Research*, pages 940–947. PMLR, 07–10 Jul 2017.

[GSYZ24] Yuzhou Gu, Zhao Song, Junze Yin, and Lichen Zhang. Low rank matrix completion via robust alternating minimization in nearly linear time. In *The Twelfth International Conference on Learning Representations (ICLR)*, 2024.

[GSZ25] Yuzhou Gu, Zhao Song, and Lichen Zhang. Faster algorithms for structured linear and kernel support vector machines. In *The Thirteenth International Conference on Learning Representations (ICLR)*, 2025.

[HKM+22] Avinatan Hassidim, Haim Kaplan, Yishay Mansour, Yossi Matias, and Uri Stemmer. Adversarially robust streaming algorithms via differential privacy. *J. ACM*, 69(6), 2022.

[HRRU14] Justin Hsu, Aaron Roth, Tim Roughgarden, and Jonathan Ullman. Privately solving linear programs. In *Proceedings of the 41st International Colloquium on Automata, Languages, and Programming (ICALP)*, volume 8572 of *Lecture Notes in Computer Science*, pages 612–624. Springer, 2014.

[HRU13] Justin Hsu, Aaron Roth, and Jonathan Ullman. Differential privacy for the analyst via private equilibrium computation. In *Proceedings of the forty-fifth annual ACM symposium on Theory of computing*, pages 341–350, 2013.

[HT09] Moritz Hardt and Kunal Talwar. On the geometry of differential privacy. In *Proceedings of the 41st Annual ACM Symposium on Theory of Computing (STOC)*, pages 705–714. ACM, 2009.

[HW01] Mark Herbster and Manfred K Warmuth. Tracking the best linear predictor. *Journal of Machine Learning Research*, 1(281-309):10–1162, 2001.

[JE19] Bargav Jayaraman and David Evans. Evaluating differentially private machine learning in practice. In *28th USENIX Security Symposium (USENIX Security 19)*, pages 1895–1912, 2019.

[JNS13] Prateek Jain, Praneeth Netrapalli, and Sujay Sanghavi. Low-rank matrix completion using alternating minimization. In *Proceedings of the forty-fifth annual ACM symposium on Theory of computing*, pages 665–674, 2013.

[Joa06] Thorsten Joachims. Training linear svms in linear time. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 217–226, 2006.

[KLL+23] Jonathan Kelner, Jerry Li, Allen Liu, Aaron Sidford, and Kevin Tian. Matrix completion in almost-verification time. In *2023 IEEE 64th Annual Symposium on Foundations of Computer Science*, FOCS'23, 2023.

[KPRU14] Michael Kearns, Mallesh Pai, Aaron Roth, and Jonathan Ullman. Mechanism design in large games: Incentives and privacy. In *Proceedings of the 5th conference on Innovations in theoretical computer science*, pages 403–410, 2014.

[LHR+10] Chao Li, Michael Hay, Vibhor Rastogi, Gerome Miklau, and Andrew McGregor. Optimizing linear counting queries under differential privacy. In *Proceedings of the 29th ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems (PODS)*, pages 123–134. ACM, 2010.

[LM00] Beatrice Laurent and Pascal Massart. Adaptive estimation of a quadratic functional by model selection. *Annals of Statistics*, pages 1302–1338, 2000.

[LT91] Michel Ledoux and Michel Talagrand. *Probability in Banach Spaces: Isoperimetry and Processes*. Classics in Mathematics. Springer, 1991.

[LWAL21] Zelun Luo, Daniel J Wu, Ehsan Adeli, and Fei-Fei Li. Scalable differential privacy with sparse network finetuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5059–5068, 2021.

[Mir17] Ilya Mironov. Rényi differential privacy. In *2017 IEEE 30th Computer Security Foundations Symposium (CSF)*, pages 263–275. IEEE, 2017.

[MMR+01] K.-R. Muller, S. Mika, G. Ratsch, K. Tsuda, and B. Scholkopf. An introduction to kernel-based learning algorithms. *IEEE Transactions on Neural Networks*, 12, 2001.

[MMSVV21] Andrés Muñoz Medina, Umar Syed, Sergei Vassilvitskii, and Ellen Vitercik. Private optimization without constraint violations. In Arindam Banerjee and Kenji Fukumizu, editors, *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics*, volume 130 of *Proceedings of Machine Learning Research*, pages 2557–2565. PMLR, 13–15 Apr 2021.

[MRTZ18] H. Brendan McMahan, Daniel Ramage, Kunal Talwar, and Li Zhang. Learning differentially private recurrent language models. In *6th International Conference on Learning Representations (ICLR)*, 2018.

[MT07] Frank McSherry and Kunal Talwar. Mechanism design via differential privacy. In *48th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 94–103. IEEE, 2007.

[NBD22] Maxence Noble, Aurélien Bellet, and Aymeric Dieuleveut. Differentially private federated learning on heterogeneous data. In Gustau Camps-Valls, Francisco J. R. Ruiz, and Isabel Valera, editors, *Proceedings of The 25th International Conference on Artificial Intelligence and Statistics*, volume 151 of *Proceedings of Machine Learning Research*, pages 10110–10145. PMLR, 28–30 Mar 2022.

[Per23] Frank Permenter. A geodesic interior-point method for linear optimization over symmetric cones. *SIAM Journal on Optimization*, 33(1):1–25, 2023.

[PST95] Serge A Plotkin, David B Shmoys, and Éva Tardos. Fast approximation algorithms for fractional packing and covering problems. *Mathematics of Operations Research*, 20(2):257–301, 1995.

[Rec11] Benjamin Recht. A simpler approach to matrix completion. *J. Mach. Learn. Res.*, 12:3413–3430, dec 2011.

[SA03] S. H. Schmieta and F. Alizadeh. Extension of primal-dual interior point algorithms to symmetric cones. *Mathematical Programming*, 96(3):409–438, 2003.

[SSSSC11] Shai Shalev-Shwartz, Yoram Singer, Nathan Srebro, and Andrew Cotter. Pegasos: primal estimated sub-gradient solver for svm. *Math. Program.*, 127, 2011.

[SWYZ23] Zhao Song, Yitan Wang, Zheng Yu, and Lichen Zhang. Sketching for first order method: efficient algorithm for low-bandwidth channel and vulnerability. In *International Conference on Machine Learning (ICML)*, pages 32365–32417. PMLR, 2023.

[SYY+23] Jiankai Sun, Xin Yang, Yuanshun Yao, Junyuan Xie, Di Wu, and Chong Wang. Dpauc: differentially private auc computation in federated learning. In *Proceedings of the Thirty-Seventh AAAI Conference on Artificial Intelligence and Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence and Thirteenth Symposium on Educational Advances in Artificial Intelligence*, AAAI'23/IAAI'23/EAAI'23. AAAI Press, 2023.

[SYYZ23] Zhao Song, Xin Yang, Yuanyuan Yang, and Lichen Zhang. Sketching meets differential privacy: fast algorithm for dynamic kronecker projection maintenance. In *International Conference on Machine Learning (ICML)*, pages 32418–32462. PMLR, 2023.

[SYYZ25] Zhao Song, Mingquan Ye, Junze Yin, and Lichen Zhang. Efficient alternating minimization with applications to weighted low rank approximation. In *The Thirteenth Internation Conference on Learning Representations (ICLR)*, 2025.

[TF20] Aleksei Triastcyn and Boi Faltings. Bayesian differential privacy for machine learning. In *International Conference on Machine Learning*, pages 9583–9592. PMLR, 2020.

[TLC+20] Stacey Truex, Ling Liu, Ka-Ho Chow, Mehmet Emre Gursoy, and Wenqi Wei. Ldp-fed: Federated learning with local differential privacy. In *Proceedings of the 3rd ACM International Workshop on Edge Systems, Analytics and Networking (EdgeSys '20)*, pages 61–66. ACM, 2020.

[TWK22] Jiyuan Tao, GQ Wang, and L Kong. The araki-lieb-thirring inequality and the golden-thompson inequality in euclidean jordan algebras. *Linear and Multilinear Algebra*, 70(19):4228–4243, 2022.

[TZXL19] Shulong Tan, Zhixin Zhou, Zhaozhuo Xu, and Ping Li. On efficient retrieval of top similarity vectors. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5239–5249, 2019.

[vAG19] Joran van Apeldoorn and András Gilyén. Improvements in quantum sdp-solving with applications. In *46th International Colloquium on Automata, Languages, and Programming (ICALP 2019)*, volume 132 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 99:1–99:15. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2019.

[VCLR13] Vincent Q. Vu, Juhee Cho, Jing Lei, and Karl Rohe. Fantope projection and selection: A near-optimal convex relaxation of sparse pca. In *Advances in Neural Information Processing Systems 26 (NeurIPS)*, pages 2670–2678, 2013.

[Ver18] Roman Vershynin. *High-Dimensional Probability: An Introduction with Applications in Data Science*. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press, 2018.

[Vie07] MVC Vieira. Jordan algebraic approach to symmetric optimization. *PhD thesis*, 2007.

[WBK19] Yu-Xiang Wang, Borja Balle, and Shiva Prasad Kasiviswanathan. Subsampled Rényi differential privacy and analytical moments accountant. In *Proceedings of the 22nd International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 1226–1235. PMLR, 2019.

[Wig58] Eugene P. Wigner. On the distribution of the roots of certain symmetric matrices. *Annals of Mathematics*, 67(2):325–327, 1958.

[WK18] Benjamin Weggenmann and Florian Kerschbaum. Syntf: Synthetic and differentially private term frequency vectors for privacy-preserving text mining. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, pages 305–314, 2018.

[WM10] Oliver Williams and Frank McSherry. Probabilistic inference and differential privacy. *Advances in Neural Information Processing Systems (NeurIPS)*, 23:2451–2459, 2010.

[XLW12] L. Xie, K. Li, and H. Wang. Stochastic gradient descent with differential privacy. In *2012 IEEE International Conference on Intelligent Control and Information Processing*, pages 47–52. IEEE, 2012.

[YDW+21] Xiang Yue, Minxin Du, Tianhao Wang, Yaliang Li, Huan Sun, and Sherman S. M. Chow. Differential privacy for text analytics via natural text sanitization. In *Findings, ACL-IJCNLP 2021*, 2021.

[ZdEG10] Youwei Zhang, Alexandre d'Aspremont, and Laurent El Ghaoui. Sparse pca: Convex relaxations, algorithms and applications. In Daniel Palomar and Yonina Eldar, editors, *Convex Optimization in Signal Processing and Communications*, pages 331–352. Springer, 2010.

[ZHT06] Hui Zou, Trevor Hastie, and Robert Tibshirani. Sparse principal component analysis. *Journal of Computational and Graphical Statistics*, 15(2):265–286, 2006.

[ZVTL24] Jiaqi Zheng, Antonios Varvitsiotis, Tiow-Seng Tan, and Wayne Lin. A primal-dual framework for symmetric cone programming. *arXiv preprint arXiv:2405.09157*, 2024.

[ZWL15] Tuo Zhao, Zhaoran Wang, and Han Liu. A nonconvex optimization framework for low rank matrix estimation. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015.

[ZYCW20] Yuqing Zhu, Xiang Yu, Manmohan Chandraker, and Yu-Xiang Wang. Private-knn: Practical differential privacy for computer vision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11854–11862, 2020.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The main contributions of this paper include a generic Gaussian mechanism for Euclidean Jordan algebra and private algorithms for symmetric cone programming. They are clearly reflected in the abstract and introduction.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The paper clearly discusses the limitations in the second paragraph of Section 6.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: All assumptions and proofs are provided, specifically, see Section 4 for the privacy proof of the generic Gaussian mechanism, Appendix B for the settings and proofs for high sensitivity constraint privacy with applications, and Appendix C for a slew of results in the low sensitivity setting.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [NA]

Justification: The paper is theoretical in nature and does not contain experiments.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [NA]

   Justification: The paper is theoretical and does not contain data and code.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [NA]

   Justification: The paper is theoretical hence contains no experiments.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [NA]

   Justification: The paper is theoretical hence contains no experiments.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [NA]

   Justification: The paper is theoretical hence contains no experiments.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

   Answer: [Yes]

   Justification: Authors have reviewed and ensured the paper adheres to the NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: Broader societal impacts have been discussed in Appendix D.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: The paper is theoretical in nature, therefore does not contain data or models.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [NA]

    Justification: The paper is theoretical in nature, therefore does not use any assets.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.

- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: The paper is theoretical in nature, therefore does not create any new assets.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: The paper is theoretical in nature, therefore does not have any crowdsourcing or research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: The paper is theoretical in nature, therefore does not require and IRB approvals.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.

- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [NA]

    Justification: LLM was only used for word editing.

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (https://neurips.cc/Conferences/2025/LLM) for what should or should not be described.

Appendix

A More Preliminaries

In this section, we provide more preliminaries on standard inequalities for EJA and multiplicative weights update for SCP.

Cauchy-Schwarz inequality is automatically satisfies given an proper inner product, we record it here.

Definition A.1 (Cauchy-Schwarz inequality for EJA)**.** *Let $x, y \in \mathcal{J}$, then we have the following standard Cauchy-Schwarz inequality:*

$$|\langle x, y\rangle| \leq \|x\|_2 \cdot \|y\|_2.$$

While Cauchy-Schwarz inequality only requires an inner product space, Hölder's inequality requires dual norms and a pairing. Nevertheless, we prove the Hölder's inequality for EJA.

Lemma A.2 (Hölder's inequality for EJA)**.** *Let $p, q \in [1, \infty]$ satisfy $1/p + 1/q = 1$, let $x, y \in \mathcal{J}$, then we have*

$$|\langle x, y\rangle| \leq \|x\|_p \cdot \|y\|_q.$$

Proof. The idea is to write x, y in their spectral decomposition and apply the standard vector Hölder's inequality. Let $x = \sum_{i=1}^r \lambda_i u_i, y = \sum_{i=1}^r \mu_i v_i$ be their respective spectral decomposition. Note that given a primitive idempotent element u, we have that $\operatorname{Tr}(c) = \|c\|_2^2 = 1$ because its rank-1. Then

$$\begin{aligned}
|\langle x, y\rangle| &= |\operatorname{Tr}(x \circ y)| \\
&= |\operatorname{Tr}(\sum_{i=1} \lambda_i u_i \circ \sum_i \mu_i v_i)| \\
&= |\sum_{i,j} \lambda_i \mu_i \operatorname{Tr}(u_i \circ v_j)| \\
&\leq \sum_{i=1}^r \sum_{j=1}^r |\lambda_i| \cdot |\mu_j| \cdot \langle u_i, v_j\rangle,
\end{aligned}$$

consider the matrix $P_{i,j} = \langle u_i, v_j\rangle$, note that P is a doubly stochastic matrix: since all u_i's, v_j's are primitive idempotent, it must be that $\langle u_i, v_j\rangle \geq 0$. Moreover, it's easy to verify that if we fix i, then

$$\begin{aligned}
\sum_{j=1}^r \langle u_i, v_j\rangle &= \langle u_i, \sum_{j=1}^r v_j\rangle \\
&= \langle u_i, e\rangle \\
&= \operatorname{Tr}(u_i) \\
&= 1,
\end{aligned}$$

and the same argument holds for rows. By Birkhoff–von Neumann theorem [Bir46], P can be written as a convex combination of permutation matrices :$P = \sum_{i=1}^r c_i \Pi_i$ where c_i's form a convex combination, so it suffices to work with any permutation matrix Π and its corresponding permutation function $\sigma : [r] \to [r]$, set $|\lambda|, |\mu|$ be the vector with entries in $|\lambda_i|, |\mu_j|$, then

$$\begin{aligned}
\sum_{i,j} |\lambda_i| \cdot |\mu_j| \cdot \Pi_{i,j} &= |\lambda|^\top \Pi |\mu| \\
&= \sum_{i=1}^r |\lambda_i| \cdot |\mu_{\sigma(i)}|. \qquad (1)
\end{aligned}$$

Thus, we can conclude that

$$\sum_{i,j} |\lambda_i| \cdot |\mu_j| \cdot \langle u_i, v_j\rangle = |\lambda|^\top P |\mu|$$

$$
\begin{aligned}
&= \sum_{i=1}^{r} c_i |\lambda|^\top \Pi_i |\mu| \\
&\leq \sum_{i=1}^{r} c_i \|\lambda\|_p \|\Pi_i \mu\|_q \\
&= \sum_{i=1}^{r} c_i \|\lambda\|_p \|\mu\|_q \\
&= \|\lambda\|_p \|\mu\|_q \\
&= \|x\|_p \|y\|_q,
\end{aligned}
$$

where we use vector Hölder's inequality in the third step. This completes the proof. □

[TWK22] proves a generalized Golden-Thompson inequality for EJA, as follows.

Lemma A.3 (Generalized Golden-Thompson inequality, [TWK22])**.** *Let $(\mathcal{J}, \circ)$ be an EJA and $x, y \in \mathcal{J}$, the generalized Golden-Thompson inequality holds:*

$$\operatorname{Tr}(\exp(x + y)) \leq \operatorname{Tr}(\exp(x) \circ \exp(y)).$$

We now describe the online linear optimization framework over symmetric cones.

Definition A.4 (Online linear optimization (OLO) framework using symmetric cone multiplicative weights update (SCMWU), [CLPV23])**.** *At each time t, the algorithm needs to pick a distributional element p^t such that $\operatorname{Tr}(p^t) = 1$ and $p^t \in \mathcal{K}$. Once the element is picked, a linear loss function $\ell^t(p) = \langle m^t, p \rangle$ where $m^t \in \mathcal{J}$. The SCMWU computes the next iterate as*

$$p^{t+1} = \frac{\exp(-\eta \sum_{\tau=1}^{t} m^\tau)}{\operatorname{Tr}(\exp(-\eta \sum_{\tau=1}^{t} m^\tau))},$$

where $p^1 = e/\operatorname{Tr}(e)$ is the uniform distribution over $\mathcal{K}$.

The main idea of SCMWU is to incur cumulative losses comparable to the losses of the best set of actions that an algorithm can make in hindsight. The discrepancy between the cumulative losses of our algorithm and the optimal algorithm is usually referred to as the regret as a function of time t. We recall the regret bound of SCMWU proved in [CLPV23].

Theorem A.5 (Theorem 5.1 of [CLPV23])**.** *Let $(\mathcal{J}, \circ)$ be an EJA of rank r, $\mathcal{K}$ be its cone of squares. For any $\eta \in (0, 1]$ and any sequence of loss vectors $\{\ell^1, \ldots, \ell^T\}$ satisfy $\|\ell^t\|_\infty \leq 1$, the iterates A^t generated by Algorithm 4 satisfy*

$$\sum_{t=1}^{T} \langle \ell^t, A^t \rangle \leq \sum_{t=1}^{T} \langle \ell^t, B \rangle + \eta T + \frac{\ln r}{\eta},$$

where B is any point in $\mathcal{K}$ satisfying $\operatorname{Tr}(B) = 1$.

B High Sensitivity Constraint Private SCP

In this section, we study algorithms for SCP in the high sensitivity constraint privacy setting. We generalize the algorithm and analysis of [HRRU14] in the LP setting.

B.1 Solving SCP with Dense Multiplicative Weights Update

Let us begin by considering constraint private SCPs over a symmmetric cone $\mathcal{K}$ in an EJA $\mathcal{J}$, with the general form

$$
\begin{aligned}
\max_{x \in \mathcal{J}} \ & \langle c, x \rangle \\
\text{s.t. } & \langle a_j, x \rangle \leq b_j, \forall j \in [m] \\
& x \in \mathcal{K},
\end{aligned}
$$

where $c, a_1, \ldots, a_m \in \mathcal{J}, b_1, \ldots, b_m \in \mathbb{R}$ and $\mathcal{K} \subseteq \mathcal{J}$.

Let $\mathcal{K}_{\mathrm{OPT}} = \mathcal{K} \cap \{x \in \mathcal{J} \mid \langle c, x\rangle = \mathrm{OPT}\}$. We reduce the SCP to the feasibility program

$$\begin{aligned} &\text{find} \;\; x \in \mathcal{K}_{\mathrm{OPT}} \\ &\text{s.t.} \;\; \langle a_j, x\rangle \le b_j, \forall j \in [m], \end{aligned}$$

then binary search the value OPT. Hence, it suffices to solve the feasibility program. As $\mathcal{K}_{\mathrm{OPT}}$ is convex, so we write $\mathcal{K}$ for $\mathcal{K}_{\mathrm{OPT}}$ for simplicity.

Let $A = \{a_1, \ldots, a_m\} \in \mathcal{J}^m$ be a collection of constraints. A database D defines an SCP as a tuple $(c(D), A(D), b(D))$, where $c(D)$ is the objective element, $A(D)$ is the set of constraints and $b(D)$ is the right hand vector. In a constraint private SCP, we have that $c(D) = c(D')$, the only differing parts are the constraints $A(D), A(D')$ and their associated scalar value. In particular, $|A(D) \cap A(D')| = m, |A(D)| = m, |A(D')| = m+1$, so $A(D), A(D')$ coincide with all m but one constraint. The scalars for the coinciding m constraints are the same, except the scalar value for the differing constraint. We define the adjacency as differing by exactly one constraint.

Definition B.1 (High sensitivity constraint privacy)**.** *Given $m \in \mathbb{N}$, vector $b \in \mathbb{R}^m$, and a constraint set $A \in \mathcal{J}^m$, a randomized mechanism $\mathcal{M}$ that outputs a vector in $\mathcal{J}$ is (ϵ, δ)-high sensitivity constraint private if for any A, A' such that $A' = A \cup \{a_{m+1}\}$, and b, b' such that $b' = [b; b_{m+1}]^\top$,*

$$\Pr[\mathcal{M}(m, b, A) \in S] \le e^\epsilon \Pr[\mathcal{M}(m+1, b', A') \in S] + \delta$$

for any subset $S \subseteq \mathcal{J}$.

Next, we introduce the dense MWU framework. Let $\mathcal{F}$ denote the universe of actions, F be the measures on the set of actions $F : \mathcal{F} \to [0, 1]$, and $\widetilde{F}$ be the respective probability distribution, defined as $\widetilde{F} = \frac{F}{|F|}$ where $|F| = \sum_{f \in \mathcal{F}} F_f$ is density of F. A key concept we will be relying on is the *Bregman projection*.

Definition B.2 (Bregman projection)**.** *Let F be a measure with $|F| \le s$ for some $s > 0$, we define $\Gamma_s F$ as the Bregman projection of F onto the set of $1/s$-dense distributions:*

$$\Gamma_s F_f := \frac{1}{s} \cdot \min\{1, cF_f\} \quad \forall f \in \mathcal{F},$$

where $c \ge 0$ is the value satisfying $s = \sum_{f \in \mathcal{F}} \min\{1, cF_f\}$.

Algorithm 2 Dense multiplicative weights update, [HW01].

procedure DENSEMWU($\mathcal{F}, \eta$)
  $F^1 \leftarrow$ the uniform distribution on $\mathcal{F}$ ▷ $F^1 \in \mathbb{R}^{|\mathcal{F}|}$
  for $t = 1 \to T - 1$ **do**
    $B^t \leftarrow \Gamma_s F^t$
    Receive loss vector ℓ^t ▷ $\ell^t \in \mathbb{R}^{|\mathcal{F}|}$
    for $f \in \mathcal{F}$ **do**
      $F_f^{t+1} \leftarrow e^{-\eta \ell_f^t} F_f^t$
    end for
  end for
  return F^T
end procedure

The following lemma due to [HW01] gives the regret bound for Algorithm 2.

Lemma B.3 ([HW01])**.** *Let F_1 denote the uniform distribution over $\mathcal{F}$ (so that $|F_1| = 1$). Consider the sequence of projected distributions $\{\widetilde{B}^t\}_{t=1}^T$ produced by Algorithm 2 under an arbitrary loss sequence $\{\ell^t\}_{t=1}^T$ with $\|\ell^t\|_\infty \le 1$ and step size $\eta \le 1/2$. Define $\widetilde{B}^*$ as the uniform distribution supported on some subset $S^* \subseteq \mathcal{F}$ of size s. Then,*

$$\frac{1}{T}\sum_{t=1}^T \langle \ell^t, \widetilde{B}^t\rangle \le \frac{1}{T}\sum_{t=1}^T \langle \ell^t, \widetilde{B}^*\rangle + \eta + \frac{\log|\mathcal{F}|}{\eta T}.$$

As standard for any MWU algorithm, we define the width of an SCP, denoted as ρ:

$$\rho \geq \max_{D} \max_{x \in \mathcal{K}} \max_{i \in [m]} |\langle a_i(D), x \rangle - b_i(D)|,$$

MWU algorithm usually assumes an oracle that can efficiently perform "simpler" minimization problem, and in our case, it corresponds to minimize over a simple linear function.

Definition B.4 ((α, β)-approximate, ρ-bounded oracle)**.** *Given a distribution $y \in \mathbb{R}^m$ and a set $A = \{a_1, \ldots, a_m\} \in \mathcal{J}^m$, an (α, β)-approximate, ρ-bounded oracle returns $x^* \in \mathcal{K}$ with probability at least $1 - \beta$ such that*

$$\langle \sum_{i=1}^{m} y_i a_i,\ x^* \rangle \leq \min_{x \in \mathcal{K}} \langle \sum_{i=1}^{m} y_i a_i,\ x \rangle + \alpha \quad \textit{and} \quad \max_{i \in [m]} \left| \langle a_i, x^* \rangle - b_i \right| \leq \rho.$$

To solve symmetric cone programs, we employ the dense symmetric cone multiplicative weights update algorithm, which maintains a distribution over the constraints and, at each iteration, selects a point $x^t \in \mathcal{K}$ that approximately minimizes the weighted violation of those constraints. Intuitively, losses increase the weight assigned to violated constraints, thereby steering subsequent iterates toward improved feasibility. Averaging the points x^t across all iterations then produces an approximately feasible solution. The full procedure is described in Algorithm 3.

Algorithm 3 SCP feasibility via DENSESCMWU.

procedure SCPDENSEMWU(A, b)
  $\widetilde{y}^1 \leftarrow \mathbf{1}_m / m$
  Let $\rho \geq \max_D \max_{x \in \mathcal{K}} \max_{i \in [m]} |\langle a_i(D), x \rangle - b_i(D)|$ be the width of the SCP, $s \in \mathbb{N}$ be the density parameter, $\alpha > 0$ be the desired accuracy
  Let ORACLE be an (α, β)-accurate, ρ-bounded oracle
  $\eta \leftarrow \sqrt{(\log m)/T}$, $T \leftarrow 36\alpha^{-2}\rho^2 \log m$
  for $t = 1 \to T$ **do**
    $x^t \leftarrow$ ORACLE($\widetilde{y}^t, A$)
    $\ell_i^t \leftarrow (1/2\rho)(b_i - \langle a_i, x^t \rangle) + 1/2$
    Update $\widetilde{y}^{t+1}$ from $\widetilde{y}^t$ and ℓ^t via dense multiplicative weights with density s
  end for
  return $\overline{x} \leftarrow (1/T) \sum_{t=1}^{T} x^t$
end procedure

Algorithm 3 differs significantly from other MWU algorithms for solving SCP such as [CLPV23, ZVTL24], where the algorithm updates the *variable* $x \in \mathcal{K}$ instead of a distribution over the constraints. In the following, we show that Algorithm 3 provides utility guarantee for solving the program. The proof is similar to [HRRU14], as the analysis focuses on the constraint, which does not exploit the structure of EJA $\mathcal{J}$.

Lemma B.5 (Approximate SCP feasibility via SCPDENSEMWU)**.** *Let $A = \{a_1, \ldots, a_m\} \in \mathcal{J}^m$ and $b \in \mathbb{R}^m$, ρ be the width of the SCP. Let $\alpha \in [0, 9\rho]$, $\beta \in (0, 1)$ and $T = 36\alpha^{-2}\rho^2 \log m$. Suppose the SCP is feasible, then Algorithm 3 with density s that utilizes an $(\alpha/3, \beta/T)$-approximate, ρ-bounded oracle can output x^* in $\mathcal{K}$ such that, with probability at least $1 - \beta$, there exists a subset of constraints $S \subseteq [m]$ with $|S| < s$ and $\langle a_i, x^* \rangle > b_i + \alpha$ for all $i \in S$.*

Proof. We will condition on the event that the oracle succeeds on all steps, note that this could be achieved via a union bound over T steps to obtain a success probability of at least $1 - \beta$.

Let $\mathcal{K}_s = \{y \in \mathbb{R}^m \mid \langle \mathbf{1}_m, y \rangle = 1, \|y\|_\infty \leq 1/s\}$ be the set of $1/s$-dense distribution. The oracle finds x^t with $\sum_{i=1}^m y_i \langle a_i, x^t \rangle \leq \sum_{i=1}^m y_i b_i + \alpha/3$. Define the i-th item in the loss vector is $\ell_i^t = (1/2\rho)(b_i - \langle a_i, x_i^t \rangle) + 1/2$. Then we have

$$\begin{aligned} \langle \ell^t, y^t \rangle &= \sum_{i=1}^{m} \ell_i^t y_i^t \\ &= (1/2\rho) \sum_{i=1}^{m} (b_i - \langle a_i, x_i^t \rangle) y_i^t + \frac{1}{2} \|y\|_1 \end{aligned}$$

$$\geq (1/2\rho)((\sum_{i=1}^{m} y_i^t b_i - y_i^t b_i) - \alpha/3) + \frac{1}{2}\|y\|_1$$
$$= \frac{1}{2} - \frac{\alpha}{6\rho},$$

if $\alpha \leq 9\rho$, then $\langle \ell^t, y^t \rangle \geq -1$. To prove an upper bound, we recall that the oracle is ρ-bounded:

$$\begin{aligned}\langle \ell^t, y^t \rangle =& (1/2\rho) \sum_{i=1}^{m} (b_i - \langle a_i, x_i^t \rangle) y_i^t + \frac{1}{2}\|y\|_1 \\ &\leq \frac{1}{2\rho} \sum_{i=1}^{m} |b_i - \langle a_i, x_i^t \rangle| y_i^t + \frac{1}{2} \\ &\leq \frac{1}{2} \sum_{i=1}^{m} y_i + \frac{1}{2} \\ &= 1,\end{aligned}$$

therefore, we can apply Lemma B.3 for the following bound, let $p \in \mathcal{K}_s$, then

$$\begin{aligned}\frac{1}{2} - \frac{\alpha}{6\rho} &\leq \frac{1}{T} \sum_{t=1}^{T} \langle \ell^t, p \rangle + \eta + \frac{\log m}{\eta T} \\ &= \frac{1}{T} \sum_{t=1}^{T} p_i \cdot (\frac{1}{2\rho}(b_i - \langle a_i, x^t \rangle) + \frac{1}{2}) + \eta + \frac{\log m}{\eta T},\end{aligned}$$

rearranging gives

$$-\frac{\alpha}{6\rho} \leq \frac{1}{T} \sum_{t=1}^{T} \sum_{i=1}^{m} p_i \cdot \frac{1}{2\rho}(b_i - \langle a_i, x^t \rangle) + \eta + \frac{\log m}{\eta T},$$

recall that we set the final output as $\overline{x} = 1/T \sum_{t=1}^{T} x^t$, the above bound could further be written as

$$\sum_{i=1}^{m} p_i \cdot \langle a_i, \overline{x} \rangle \leq \sum_{i=1}^{m} p_i \cdot b_i + 2\rho\eta + \frac{2\rho \log m}{\eta T} + \frac{\alpha}{3},$$

picking $\eta = \frac{\log m}{T}$ and $T = \frac{100\rho^2 \log m}{\alpha^2}$ gives

$$\sum_{i=1}^{m} p_i \cdot \langle a_i, \overline{x} \rangle \leq \sum_{i=1}^{m} p_i \cdot b_i + \alpha.$$

As the above bound holds for any $p \in \mathcal{K}_s$, it must be the case that $\overline{x}$ satisfies all but at most $s-1$ constraints with additive error α. Suppose otherwise, there are s constraints violate the α-additive error condition, then we could set p as the uniform distribution over these s constraints and this implies that these s constraints satisfy the α-additive error condition, a contradiction. □

B.2 Privacy Guarantee of Dense Multiplicative Weights Update

Next, we prove that Algorithm 3 is private as long as the oracle is private. The algorithm only utilizes the private data via the oracle minimization, hence, as long as the oracle is privately minimizes over $\mathcal{K}$ at each step $t \in [T]$, the final output will automatically be private as $\mathcal{K}$ is convex. To do so, we recall a lemma first proved in [HRU13], showing that for two neighboring databases that differ by one action, then their projected distributions $\widetilde{y}$ and $\widetilde{y}'$ satisfy $\|\widetilde{y} - \widetilde{y}'\|_1 \leq 2/s$.

Lemma B.6 ([HRU13])**.** *Let $F : \mathcal{F} \to [0,1]$ and $F' : \mathcal{F} \cup \{f'\} \to [0,1]$ be two measures over their respective set of actions. Let density parameter $s \in \mathbb{N}$ and it satisfies that (1) $|F|, |F'| \leq s$, (2) $F_f = F'_f$ for every $f \in \mathcal{F}$. Let $\widetilde{F}, \widetilde{F}'$ be the corresponding Bregman projections onto the set of $1/s$-dense distributions, then we have*

$$\|\widetilde{F} - \widetilde{F}'\|_1 \leq 2/s$$

Now we are able to present our main theorem for high sensitivity constraint privacy:

Theorem B.7 (Privacy guarantee)**.** *Let $\epsilon, \delta > 0$. Consider a symmetric cone program with constraint matrix $A \in \mathcal{J}^m$, vector $b \in \mathbb{R}^m$, and width ρ. Fix parameters $\alpha \in [0, 9\rho]$, $\beta \in (0, 1)$, and $T = 3, \alpha^{-2}\rho^2 \log m$. Given access to an $(\alpha/3, \beta/T)$-approximate, ρ-bounded oracle that is also ϵ'-private where $\epsilon' = \frac{\epsilon}{\sqrt{8T \log(1/\delta)}}$. For any neighboring instances, the oracle inputs satisfy*

$$\|\widetilde{y}\|_\infty \leq \tfrac{1}{s}, \qquad \|\widetilde{y}'\|_\infty \leq \tfrac{1}{s}, \qquad \|\widetilde{y} - \widetilde{y}'\|_1 \leq \tfrac{2}{s}.$$

Then Algorithm 3, run with density parameter s, is (ϵ, δ)-high sensitivity constraint private.

Proof. Since the oracle is ϵ'-private, the overall (ϵ, δ)-constraint private is achieved via adaptive composition (Lemma 3.13). When adding or removing a constraint, we note that A and A' are identical except for one additional constraint. Since we project the distribution onto $\mathcal{K}_s$, it must be the case $\|\widetilde{y}\|_\infty, \|\widetilde{y}'\|_\infty \leq 1/s$, and $\|\widetilde{y} - \widetilde{y}'\|_1 \leq 2/s$ follows from Lemma B.6. Thus, since two distributions are identical except for the probability associated with the additional constraint, we have completed the proof. □

B.3 Application I: Private Solver for Covering SDP

As an application, we show how to develop a private solver for covering semidefinite programming:

$$\begin{aligned} \min_{X \in \mathbb{R}^{r \times r}} \ & \mathrm{Tr}(X) \\ \text{s.t. } & \langle A_i, X \rangle \geq 1, \forall i \in [m] \\ & X \succeq 0 \end{aligned}$$

where $A_1, \ldots, A_m \succeq 0$. We without loss of generality assume $\max_{i \in [m]} \|A_i\| \leq 1$, note that this can be done via scaling all the A_i's. Again, we consider the convex set $\mathcal{K}_{\mathrm{OPT}} = \{X \succeq 0 : \mathrm{Tr}(X) = \mathrm{OPT}\}$ and we will assume OPT is known, as otherwise it could be found by binary search. Thus, we can consider the feasibility problem

$$\begin{aligned} & \text{find } X \in \mathcal{K}_{\mathrm{OPT}} \\ & \text{s.t. } \langle A_i, X \rangle \geq 1, \forall i \in [m]. \end{aligned}$$

It is natural to study high sensitivity constraint privacy in this setting, as one could view A_i as the positive semidefinite constraint matrix associated with each individual data point as in the case of robust mean estimation [CDG19], robust covariance estimation [CDGW19] and E-optimal experimental design [vAG19].

We need to design an oracle to solve the minimization problem

$$O(y) = \arg \min_{X \in \mathcal{K}_{\mathrm{OPT}}} \langle \sum_{i=1}^m y_i A_i, X \rangle,$$

this is a linear minimization problem, and $\mathcal{K}_{\mathrm{OPT}}$ is the intersection of a hyperplane and the positive semidefinite cone, and the solutions to the linear minimization are the extreme rays of the cone:

$$X^* = uu^\top,$$

where $\|u\|_2^2 = \mathrm{OPT}$. To implement a private oracle, it is tempting to use exponential mechanism directly, but note that there are infinitely many r-dimensional vectors u satisfying $\|u\|_2^2 = \mathrm{OPT}$, so the accuracy guarantee of Lemma 3.12 because meaningless as $|\mathcal{R}| = \infty$. To address this issue, we use a γ-net argument to quantize the ball $B = \{u \in \mathbb{R}^r : \|u\|_2^2 \leq \mathrm{OPT}\}$ into finitely many points.

Lemma B.8 ([LT91, Ver18])**.** *Let $B = \{u \in \mathbb{R}^r : \|u\|_2 \leq R\}$, there exists a finite collection of points $N \subset \mathbb{R}^r$ such that for any $u \in B$, there exists $v \in N$ such that $\|u - v\|_2 \leq \gamma$ for $\gamma > 0$. Moreover, $|N| = O((R/\gamma)^r)$.*

Our private oracle will then be the exponential mechanism, over the net N.

Lemma B.9. *Let $\mathrm{OPT}, \epsilon > 0$, $\beta \in (0, 1)$, $s \in \mathbb{N}$, and define $B = \{u \in \mathbb{R}^r : \|u\|_2^2 \leq \mathrm{OPT}\}$. Assume neighboring inputs $y, y' \in \mathbb{R}^m$ satisfy $\|y\|_\infty \leq \frac{1}{s}$, $\|y'\|_\infty \leq \frac{1}{s}$ and $\|y - y'\|_1 \leq \frac{2}{s}$. Let N be*

a $\sqrt{\mathrm{OPT}/2}$-net of B (Lemma B.8), and let $O(y)$ denote the ϵ-private exponential mechanism on N with quality score

$$Q(u, y) = \langle \sum_{i=1}^{m} y_i A_i, uu^\top \rangle - 1.$$

Then $O(y)$ is an (α, β)-approximate, ρ-bounded oracle with

$$\alpha = \frac{6r\,\mathrm{OPT}}{s\epsilon} \log(1/\beta), \qquad \rho \leq 3\,\mathrm{OPT} - 1.$$

Proof. We first need to prove the width of the oracle. For any point $u \in N$, we know that there exists a point $v \in B$ with $\|u - v\|_2 \leq \gamma$, therefore

$$\begin{aligned}
\langle A_i, uu^\top \rangle - 1 &= u^\top A_i u - 1 \\
&\leq \|u\|_2^2 - 1 \\
&\leq 2\|v\|_2^2 + 2\|u - v\|_2^2 - 1 \\
&\leq 2(\mathrm{OPT} + \gamma^2) - 1.
\end{aligned}$$

For accuracy guarantee, we first need to compute the sensitivity of the quality score. To do so, note that for the first m entries, we have $\sum_{i=1}^{m} |y_i - y_i'| \leq 2/s$ and D' might have one more constraint, therefore the extra entry of y' has its magnitude being at most $1/s$. We let $y, y' \in \mathbb{R}^{m+1}$ and set $y_{m+1} = 0$, the sensitivity is

$$\begin{aligned}
|\langle \sum_i y_i A_i - \sum_i y_i' A_i, uu^\top \rangle| &= |u^\top (\sum_i y_i A_i - \sum_i y_i' A_i) u| \\
&= |\sum_i (y_i - y_i') u^\top A_i u| \\
&\leq \max_i |u^\top A_i u| \cdot \|y - y'\|_1 \\
&\leq \max_i \|A_i\| \cdot \|u\|_2^2 \cdot 3/s \\
&\leq \frac{3\,\mathrm{OPT}}{s}.
\end{aligned}$$

To obtain the final bound, we need a handle on $|N|$. Pick $\gamma = \sqrt{\frac{\mathrm{OPT}}{2}}$, then $|N| \leq \exp(r)$ and by Lemma 3.12, we can choose α as

$$\begin{aligned}
\alpha &= \frac{6\,\mathrm{OPT}}{s \cdot \epsilon} \log(|N|/\beta) \\
&= \frac{6r \cdot \mathrm{OPT}}{s \cdot \epsilon} \log(1/\beta),
\end{aligned}$$

this ensures a success probability of at least $1 - \beta$. We hence complete the proof. □

We are set in a position to state the final privacy and utility guarantee of our algorithm for high sensitivity constraint privacy.

Theorem B.10 (Formal version of Theorem 5.1)**.** *Let $\beta, \delta \in (0, 1), \epsilon > 0$. Algorithm 3 with the oracle $O(y)$ as in Lemma B.9 solves a covering SDP with m constraints by outputting $X^* \succeq 0$ such that with probability at least $1 - \beta$, we have $\langle A_i, X^* \rangle \geq 1 - \alpha$ for all but s constraints where*

$$s = \Omega\left(\frac{r}{\epsilon} \cdot \log^{1/2}(1/\delta) \log(1/\beta) \log m\right)$$

and $\alpha = O(\mathrm{OPT})$. Moreover, Algorithm 3 is ϵ-private with respect to high sensitivity constraint privacy.

Proof. To apply Theorem B.7, we require $\alpha \leq 9\rho$, and need to use an $(\alpha/3, \beta/T)$-approximate oracle with ϵ'-private. As $T = 36\alpha^{-2}\rho^2 \log m$ and $\epsilon' = \frac{\epsilon}{\sqrt{8T \log(1/\delta)}}$, we plug in these choices into Lemma B.9:

$$\alpha = \frac{18r \cdot \mathrm{OPT}}{s \cdot \epsilon'} \log(T/\beta)$$

$$
\begin{aligned}
&\leq \frac{54r \cdot \mathrm{OPT} \cdot \sqrt{T \log(1/\delta)}}{s \cdot \epsilon} \log(T/\beta) \\
&= O\left(\frac{r \cdot \mathrm{OPT} \cdot \sqrt{T}}{s \cdot \epsilon} \sqrt{\log(1/\delta)} \log(1/\beta) \right) \\
&= O\left(\frac{r \cdot \mathrm{OPT} \cdot \rho}{s \cdot \epsilon \cdot \alpha} \cdot \sqrt{\log(1/\delta)} \log(1/\beta) \log m \right),
\end{aligned}
$$

solve for α, we get

$$
\alpha = C \cdot \sqrt{\frac{r \cdot \mathrm{OPT} \cdot \rho}{s \cdot \epsilon}} \cdot \log^{1/4}(1/\delta) \log^{1/2}(1/\beta) \log^{1/2} m,
$$

and the requirement $\alpha \leq 9\rho$ forces that

$$
\sqrt{\frac{r \cdot \mathrm{OPT} \cdot \rho}{s \cdot \epsilon}} \cdot \log^{1/4}(1/\delta) \log^{1/2}(1/\beta) \log^{1/2} m \leq C' \rho,
$$

rearranging gives

$$
\begin{aligned}
\sqrt{s} &\geq \frac{1}{C'} \cdot \sqrt{\frac{r \cdot \mathrm{OPT}}{\rho \cdot \epsilon}} \cdot \log^{1/4}(1/\delta) \log^{1/2}(1/\beta) \log^{1/2} m \\
s &= \Omega\left(\frac{r \cdot \mathrm{OPT}}{\rho \cdot \epsilon} \cdot \log^{1/2}(1/\delta) \log(1/\beta) \log m \right) \\
&= \Omega\left(\frac{r}{\epsilon} \cdot \log^{1/2}(1/\delta) \log(1/\beta) \log m \right).
\end{aligned}
$$

This completes the proof. □

B.4 Application II: Private Solver for Covering SCP

More generally, consider the covering symmetric cone programming:

$$
\begin{aligned}
\min_{x \in \mathcal{J}} \ & \mathrm{Tr}(x) \\
\text{s.t. } & \langle a_i, x \rangle \geq 1, \forall i \in [m] \\
& x \in \mathcal{K},
\end{aligned}
$$

where $a_1, \ldots, a_m \in \mathcal{K}$. We can again without loss of generality assume $\max_{i \in [m]} \|a_i\|_\infty \leq 1$. Set $\mathcal{K}_{\mathrm{OPT}} = \{x \in \mathcal{K} : \mathrm{Tr}(x) = \mathrm{OPT}\}$, we reduce the optimization to feasibility:

$$
\begin{aligned}
\text{find } & x \in \mathcal{K}_{\mathrm{OPT}} \\
\text{s.t. } & \langle a_i, x \rangle \geq 1, \forall i \in [m].
\end{aligned}
$$

Recall that we need to design an oracle to solve

$$
O(y) = \arg \min_{x \in \mathcal{K}_{\mathrm{OPT}}} \langle \sum_{i=1} y_i a_i, x \rangle,
$$

in the case of SDP, we noted that the optimal solutions to a linear minimization over the positive semidefinite cone are the extreme rays of the cone, i.e., rank-1 positive semidefinite matrices. For SCP, we consider the case where $\mathcal{J}$ is a simple Jordan algebra, which covers the interesting cases including real symmetric matrices, Hermitian symmetric matrices and spin factors (the algebra whose cone of the squares is the second-order cone). In this scenario, it's easy to see that the optimal solutions are the primitive idempotent q with $\|q\|_2^2 = \mathrm{OPT}$. To use a γ-net argument, we need to understand the dimension of all the primitive idempotents in $\mathcal{J}$. These primitive idempotents form a connected and compact manifold, whose dimension can be characterized as follows:

Lemma B.11 ([FK94]). *Let $\mathcal{J}$ be a simple Jordan algebra with rank r and Peirce constant d. Let $Q = \{q \in \mathcal{J} : q \text{ is a primitive idempotent}\}$, then $\dim(Q) = d(r-1)$. Let $C = \{c \cdot q : c \in \mathbb{R}, q \in Q\}$, then $\dim(C) = d(r-1) + 1$.*

Algebra $\mathcal{J}$	Rank r	Peirce constant d
$r \times r$ real symmetric matrices	r	1
$r \times r$ complex Hermitian matrices	r	2
$r \times r$ quaternionic Hermitian matrices	r	4
r-dimensional spin factors	2	r
Albert Algebra	3	8

Table 1: Five types of simple Jordan algebras, their rank and Peirce constant.

Peirce constant is a parameter related to the Pierce decomposition of a Jordan frame. We list r and d for all the simple Jordan algebras.

We note that r-dimensional spin factors are $(r+1)$-dimensional vectors whose cone of the square is the second-order cone. We can hence conclude that for all simple Jordan algebras, the dimension of C is at most $4r$. We could then use exactly the same argument for Lemma B.9 and B.10 for covering SCP.

Lemma B.12. *Let* $\mathrm{OPT}, \epsilon > 0$, $\beta \in (0,1)$, $s \in \mathbb{N}$, *let* $C_{\mathrm{OPT}} = \{c \cdot q : c^2 \leq \mathrm{OPT}, q \in Q\}$. *Assuming neighboring inputs* $y, y' \in \mathbb{R}^m$ *satisfying* $\|y\|_\infty, \|y'\|_\infty \leq 1/s, \|y - y'\|_1 \leq 2/s$. *Let* N *be a* $\sqrt{\mathrm{OPT}/2}$*-net of* C_{OPT} *and let* $O(y)$ *denote the* ϵ*-private exponential mechanism on* N *with quality score*

$$Q(p, y) = \langle \sum_{i=1}^{m} y_i a_i, p \rangle - 1.$$

Then, $O(y)$ *is an* (α, β)*-approximate,* ρ*-bounded oracle with*

$$\alpha = \frac{24r \cdot \mathrm{OPT}}{s \cdot \epsilon} \log(1/\beta), \quad \rho \leq 3\,\mathrm{OPT} - 1.$$

Theorem B.13 (Formal version of Theorem 5.2)**.** *Let* $\beta, \delta \in (0,1), \epsilon > 0$. *Algorithm 3 with the oracle* $O(y)$ *as in Lemma B.12 solves a covering SCP with* m *constraints by outputting* $x^* \in \mathcal{K}$ *such that with probability at least* $1 - \beta$*, we have* $\langle a_i, x^* \rangle \geq 1 - \alpha$ *for all but* s *constraints where*

$$s = \Omega\left(\frac{r}{\epsilon} \cdot \log^{1/2}(1/\delta) \log(1/\beta) \log m\right)$$

and $\alpha = O(\mathrm{OPT})$. *Moreover, Algorithm 3 is* ϵ*-private with respect to high sensitivity constraint privacy.*

C Low Sensitivity SCPs

For low sensitivity SCPs, the divergence between neighboring inputs diminishes as the database size increases. We continue to study the feasibility program in the following form:

$$\begin{aligned} &\text{find } x \in \mathcal{K} \\ &\text{s.t. } \langle a_i, x \rangle \leq b_i, \forall i \in [m]. \end{aligned}$$

We further without loss of generality normalize the constraints so that the feasible solutions are distributions over $\mathcal{J}$. The reduction is simple, if the optimal solution has trace L, then consider

$$\begin{aligned} &\text{find } x \in \mathcal{K} \\ &\text{s.t. } \langle a_i, x \rangle \leq b_i/L, \forall i \in [m] \end{aligned}$$

has a distribution solution. Once computing a solution x^*, we can scale back to obtain a solution for the unscaled program. The only downside is that if the constraints are only approximately satisfied: $\langle a_i, x^* \rangle \leq b_i/L + \alpha$ for all $i \in [m]$, by setting $\alpha = \alpha'/L$, then $\langle a_i, Lx^* \rangle \leq b_i + \alpha'$.

C.1 Solving SCPs with Multiplicative Weights

For convenience, we present a standard algorithm framework for multiplicative weights update. We note Algorithm 4 is quite different from the standard multiplicative weights update, which would

take the form

$$F^{t+1} \leftarrow \frac{F^t \circ \exp(-\eta \ell^t)}{\mathrm{Tr}(F^t \circ \exp(-\eta \ell^t))},$$

however, this turns out to be insufficient since $\exp(A) \circ \exp(B) \neq \exp(A+B)$ in general unless A and B commute, see [CLPV23] for more details. The regret bound of Algorithm 4 has been recorded in Lemma B.3.

Algorithm 4 Multiplicative weights update algorithm.

procedure MWU($\mathcal{F}, \eta$)
  $F^1 \leftarrow$ the uniform distribution on $\mathcal{F}$
  for $t = 1 \to T-1$ **do**
    Receive loss $\ell^t \in \mathcal{J}$ (may depend on $F^1, \ldots, F^t$)
    $F^{t+1} \leftarrow \frac{\exp(-\eta \sum_{i=1}^t \ell^t)}{\mathrm{Tr}(\exp(-\eta \sum_{i=1}^t \ell^t))}$
  end for
  return F^T
end procedure

Our MWU algorithm for solving SCP is somewhat different from the variants introduced in [ZVTL24], as it relies on a dual oracle that returns an approximately most violated constraint. This should be treated as a generalization of the dual oracle for LP introduced in [HRRU14].

Definition C.1 (Dual oracle)**.** *For $\gamma > 0$, an (α, γ)-dual oracle, given $A = \{a_1, \ldots, a_m\} \in \mathcal{J}^m$, $b \in \mathbb{R}^m$, and $x \in \mathcal{K}$ as input, returns an index $i \in [m]$ such that*

$$\langle a_i, x \rangle - b_i \geq \max_{j \in [m]} (\langle a_j, x \rangle - b_j) - \alpha,$$

with probability at least $1 - \gamma$, provided that

$$\max_{j \in [m]} (\langle a_j, x \rangle - b_j) \geq 0.$$

The full algorithm is given in 5.

Algorithm 5 Solving SCP feasibility via MWU over primal variables.

procedure SCPMWUPRIMAL($A \in \mathcal{J}^m, b \in \mathbb{R}^m$)
  $x^1 \leftarrow e/r$
  Let $\rho = \max_{i \in [m]} \|a_i\|_\infty$ be the width of the SCP, $\alpha > 0$ be the desired accuracy
  Let ORACLE be a (α, γ)-dual oracle,
  $\eta \leftarrow \frac{\alpha}{4\rho}, T \leftarrow \frac{16\rho^2 \log r}{\alpha^2}$
  for $t = 0 \to T-1$ **do**
    $p^t \leftarrow$ ORACLE(A, b, x^t)
    $\ell^t \leftarrow \frac{1}{\rho} a_{p^t}$
    $x^{t+1} \leftarrow \frac{\exp(-\sum_{i=1}^t \eta \ell^i)}{\mathrm{Tr}(\exp(-\sum_{i=1}^t \eta \ell^i))}$.
  end for
  return $\overline{x} = (1/T) \sum_{t=1}^T x^t$
end procedure

Our MWU framework is based on SCP variant introduced in [CLPV23, ZVTL24], hence its proof differs significantly from [PST95] as the update is different. Moreover, the SCP solver due to [ZVTL24] utilizes *primal oracle*. Hence, we prove a convergence theorem for the dual oracle.

Theorem C.2 (Formal version of Theorem 5.3)**.** *Suppose the SCP admits a distributional feasible solution $x \in \mathcal{K}$ with $\mathrm{Tr}(x) = 1$. Then Algorithm 5, when equipped with an $(\alpha/2, \gamma)$-dual oracle, returns $x \in \mathcal{K}$ such that*

$$\langle a_i, x \rangle \leq b_i + \alpha \qquad \forall i \in [m],$$

with probability at least $1 - T\gamma$.

Proof. Our proof will be applying Theorem A.5, to do so, we first need to verify that the sequence of loss satisfy that $\|\ell_t\|_\infty \leq 1$. As we set ρ to be the max infinity norm over all constraints, it's easy to see that $\|\ell^t\|_\infty \leq 1$. Next, we assume the oracle is exact, i.e., it returns $\max_{j\in[m]}\langle a_j, x\rangle - b_j$. Then we show how to generalize the argument to approximate oracles. By Theorem A.5, we have that

$$\sum_{t=1}^{T}\langle \ell^t, x^t\rangle \leq \sum_{t=1}^{T}\langle \ell^t, y\rangle + \eta T + \frac{\ln r}{\eta}, \tag{2}$$

for any probability distribution $y \in \mathcal{K}$. Set $r^t := \langle a_{p^t}, x^t\rangle - b_{p^t}$, we can explicitly compute the LHS:

$$\begin{aligned}\sum_{t=1}^{T}\langle \ell^t, x^t\rangle &= \frac{1}{\rho}\sum_{t=1}^{T}\langle a_{p^t}, x^t\rangle \\ &= \frac{1}{\rho}\sum_{t=1}^{T}(b_{p^t} + r^t),\end{aligned}$$

meanwhile since we can choose any distribution $y \in \mathcal{K}$, we choose a feasible y so that for any $i \in [m]$, $\langle a_i, y\rangle \leq b_i$, thus

$$\begin{aligned}\sum_{t=1}^{T}\langle \ell^t, y\rangle &= \frac{1}{\rho}\sum_{t=1}^{T}\langle a_{p^t}, y\rangle \\ &\leq \frac{1}{\rho}\sum_{t=1} b_{p^t},\end{aligned}$$

combining these inequalities yields

$$\frac{1}{\rho}\sum_{t=1}^{T} r^t \leq \eta T + \frac{\ln r}{\eta},$$

multiplying both sides by $\frac{\rho}{T}$ gives

$$\frac{1}{T}\sum_{t=1}^{T} r_t \leq \eta\rho + \frac{\rho \ln r}{\eta T},$$

setting $\eta = \frac{\epsilon}{2\rho}$ and $T = \frac{4\rho^2 \ln r}{\epsilon^2}$, the above bound simplifies to

$$\frac{1}{T}\sum_{t=1} r_t \leq \epsilon,$$

to obtain a final bound for any constraint, fix $i \in [m]$, we note that for any $t \in [T]$, it must be the case that $\langle a_i, x^t\rangle - b_i \leq \langle a_{p^t}, x^t\rangle - b_{p^t}$, and if we average it gives

$$\begin{aligned}\frac{1}{T}\sum_{t=1}^{T}\langle a_i, x^t\rangle - b_i &= \langle a_i, \overline{x}\rangle - b_i \\ &\leq \frac{1}{T}\sum_{t=1}^{T} r_t \\ &\leq \epsilon,\end{aligned}$$

as desired. Now recall in our algorithm we are using an approximate dual oracle, and we slightly modify the analysis by setting $\epsilon = \alpha/2$, and note that the only place we are using the most violated constraint is to prove all constraints, where using an $(\alpha/2, \gamma)$-dual oracle would blow up the final error by a factor of $\alpha/2$. Together with the choice of ϵ, we conclude that for any $i \in [m]$, $\langle a_i, \overline{x}\rangle - b_i \leq \alpha$ holds with probability at least $1 - T\gamma$ (via a union bound). □

C.2 Scalar Private SCPs

We first consider the "simpler" case of scalar private SCPs. This is when the objective and constraints are public data, and only the scalar vector b is private. A private database D is mapped to a tuple $(A(D), c(D), b(D))$. For neighboring databases D, D', the mappings satisfy

$$c(D) = c(D'),$$
$$A(D) = A(D'),$$

i.e., both the objective vector c and the constraint matrix A are independent of the data. Moreover, the right-hand side vectors differ only by a bounded amount:

$$\|b(D) - b(D')\|_\infty \leq \Delta_\infty.$$

Note that since $b \in \mathbb{R}^m$, here the infinity norm is the standard vector ℓ_∞ norm. We again focus on the feasibility SCP. Formally:

Definition C.3. *Given a vector $b \in \mathbb{R}^m$, and a constraint set $A \in \mathcal{J}^m$, a randomized mechanism $\mathcal{M}$ that outputs a vector in $\mathcal{J}$ is (ϵ, δ)-low sensitivity scalar private if for any b, b' with $\|b - b'\|_\infty \leq \Delta_\infty$,*

$$\Pr[\mathcal{M}(b, A) \in S] \leq e^\epsilon \Pr[\mathcal{M}(b', A) \in S] + \delta$$

for any subset $S \subseteq \mathcal{J}$.

To achieve privacy, we will implement a private dual oracle via exponential mechanism, henceforce privatize Algorithm 5.

Lemma C.4. *Let $\epsilon > 0$ and $\gamma \in (0, 1)$. Suppose that on neighboring instances the vector b changes by at most Δ_∞ in ℓ_∞ norm. Then, the ϵ-private exponential mechanism with quality score*

$$Q(i, b) = \langle a_i, x \rangle - b_i$$

is an (α, γ)-dual oracle, where

$$\alpha = \frac{2\Delta_\infty}{\epsilon} \log(\frac{m}{\gamma}).$$

Proof. Note that our oracle will be following the procedure:

- Given A, x, b, computing $Q(i, b)$ for all $i \in [m]$;
- Sample constraint i with probability $\exp(\frac{\epsilon}{2\Delta_\infty} \cdot Q(i, b))$.

This procedure is automatically ϵ-private following the definition of exponential mechanism (Def. 3.11), so it remains to prove the procedure indeed implements an (α, γ)-dual oracle. By Lemma 3.12, this is true if $\alpha = \frac{2\Delta_\infty}{\epsilon} \cdot \log(\frac{m}{\gamma})$, which completes the proof. □

We are now in the position to provide privacy and accuracy guarantees for scalar private SCPs.

Theorem C.5. *Let $\epsilon > 0$, $\delta, \beta \in (0, 1)$, and define $\rho = \max_{i \in [m]} \|a_i\|_\infty$. Then Algorithm 5 is (ϵ, δ)-low sensitivity scalar private with sensitivity Δ_∞, and with probability at least $1 - \beta$ outputs $x^* \in \mathcal{K}$ such that $\langle a_i, x^* \rangle \leq b_i + \alpha$ for all $i \in [m]$, where*

$$\alpha = \widetilde{O}\left(\frac{\rho^{1/2} \Delta_\infty^{1/2}}{\epsilon^{1/2}} \log^{1/4} r \log^{1/4}(1/\delta) \log^{1/2}(1/\beta) \log^{1/2} m \right).$$

Proof. Let $\epsilon' = \frac{\epsilon}{\sqrt{8T \log(1/\delta)}}$, we use exponential mechanism to implement an ϵ'-private dual oracle, and by adaptive composition (Lemma 3.13), this gives a final (ϵ, δ)-private algorithm. For success probability, we set $\gamma = \beta/T$. By Lemma C.4, we have

$$\begin{aligned} \alpha &= \frac{2\Delta_\infty}{\epsilon'} \cdot \log(\frac{mT}{\beta}) \\ &= \frac{2\Delta_\infty \sqrt{8T \log(1/\delta)}}{\epsilon} \cdot \log(\frac{mT}{\beta}) \end{aligned}$$

$$= \frac{8\Delta_\infty \rho \log^{1/2} r \log^{1/2}(1/\delta)}{\alpha\epsilon} \cdot \log(\frac{16m\rho^2 \log r}{\alpha^2\beta}),$$

solving for α gives

$$\alpha = \widetilde{O}\left(\frac{\rho^{1/2}\Delta_\infty^{1/2}}{\epsilon^{1/2}} \log^{1/4} r \log^{1/4}(1/\delta) \log^{1/2}(1/\beta) \log^{1/2} m\right),$$

as desired. This completes the proof. □

C.3 Low Sensitivity Constraint Private SCPs

Given the feasibility program

$$\text{find } x \in \mathcal{K}$$
$$\text{s.t. } \langle a_i, x\rangle \leq b_i, \forall i \in [m],$$

and a neighboring instance would perturb some constraints by small amounts. More specifically, for two set of constraints $A, A' \in \mathcal{J}^m$, we define the global infinity norm as

$$\|A - A'\|_\infty = \max_{i\in[m]} \|a_i - a_i'\|_\infty,$$

and we assume for two neighboring instances, their distance in global infinity norm is at most Δ_∞.

Similarly to the scalar private setting, a private database D is mapped to a tuple $(A(D), c(D), b(D))$. For every pair of neighboring databases D, D', the mappings satisfy

$$c(D) = c(D'),$$
$$b(D) = b(D'),$$

i.e., both the objective vector c and the scalar b are independent of the data. Moreover, the constraints differ by a bounded amount

$$\|A(D) - A(D')\|_\infty \leq \Delta_\infty.$$

Formally:

Definition C.6. *Given a vector $b \in \mathbb{R}^m$, and a constraint set $A \in \mathcal{J}^m$, a randomized mechanism $\mathcal{M}$ that outputs a vector in $\mathcal{J}$ is (ϵ, δ)-low sensitivity constraint private if for any A, A' such that $\|A - A'\|_\infty \leq \Delta_\infty$,*

$$\Pr[\mathcal{M}(b, A) \in S] \leq e^\epsilon \Pr[\mathcal{M}(b, A') \in S] + \delta$$

for any subset $S \subseteq \mathcal{J}$.

We without loss of generality normalize the constraints so that the spectrum of each a_i lies in $[-1, 1]$. Our algorithm will again be implementing the dual oracle with exponential mechanism. However, as the oracle returns a constraint and it will be used to compute the loss, we have to add one more layer of privacy via the generic Gaussian mechanism.

Theorem C.7 (Privacy guarantee of Theorem 5.4)**.** *Let ϵ, ϵ', and Δ_∞ be as defined in Algorithm 1, and suppose the algorithm employs an ϵ'-private dual oracle. Then Algorithm 1 is (ϵ, δ)-low sensitivity constraint private with sensitivity Δ_∞.*

Proof. We note the privacy comes from two sources: the Gaussian mechanism and the oracle operation. For Gaussian mechanism, by Lemma 4.2, we know that each operation is $(\epsilon', \delta/T)$-private, and each oracle is ϵ'-private. By an adaptive composition over $2T$ operations, we see that the algorithm is (ϵ, δ)-private. □

We prove that the exponential mechanism is a private dual oracle under this setting of neighboring.

Lemma C.8. *Let $\epsilon > 0$ and $\gamma \in (0, 1)$. Suppose that on neighboring instances the constraint set A changes by at most Δ_∞ in ℓ_∞ norm. Let $x \in \mathcal{K}$ be any distributional element. Then, the ϵ-private exponential mechanism with quality score*

$$Q(i, A) = \langle a_i, x\rangle - b_i$$

is an (α, γ)-dual oracle, for

$$\alpha = \frac{2\Delta_\infty}{\epsilon} \cdot \log(\frac{m}{\gamma}).$$

Proof. Since x is a distribution, the quality score Q changes by at most Δ_∞ on neighboring inputs, and hence is Δ_∞-sensitive. The claimed accuracy bound then follows directly from the accuracy guarantee of the exponential mechanism (Lemma 3.12). □

To prove the convergence, we need to slightly change the original argument, as we have to perturb each constraint by a Gaussian noise element. We give a customized proof based on standard regret bound.

Theorem C.9 (Utility guarantee of Theorem 5.4)**.** *Let $A \in \mathcal{J}^m$ satisfy $\lambda(a_i) \subseteq [-1,1]$ for all $i \in [m]$, and let $b \in \mathbb{R}^m$. Fix $\beta, \epsilon > 0$ and $\delta \in (0,1)$. Let r denote the rank of $\mathcal{J}$ and k its dimension. Then, with probability at least $1-\beta$, Algorithm 1, when run with the exponential mechanism as a dual oracle (Lemma C.8), returns a distribution x^* such that*

$$\langle a_i, x^* \rangle \leq b_i + \alpha \qquad \forall i \in [m],$$

where

$$\alpha = \widetilde{O}\left(\frac{\Delta_\infty^{1/2} r^{1/4} k^{1/4}}{\epsilon^{1/2}} \cdot \operatorname{poly}\log(r, 1/\beta, 1/\delta) \right).$$

Proof. Let ϵ' be as in Theorem C.7, T be the number of iterations in Algorithm 1, and set $\gamma = \beta/(2T)$. By Lemma C.8, with probability at least $1-\gamma$, the oracle returns p^t such that for all constraints $i \in [m]$,

$$(\langle a_i, x^t \rangle - b_i) - (\langle a_{p^t}, x^t \rangle - b_{p^t}) \leq \frac{2\Delta_\infty}{\epsilon'} \cdot \log(\frac{m}{\gamma}) \tag{3}$$

Here p^t is chosen as the constraint that is nearly the most violated, up to an additive factor of α. Define the vanilla loss $\ell^t = a_{p^t}$. Note that the left-hand side of (3) is exactly $(\langle a_i, x^t \rangle - b_i) - (\langle \ell^t, x^t \rangle - b_{p^t})$. Applying a union bound over the T oracle calls, inequality (3) holds simultaneously for all $t \in [T]$ with probability at least $1 - \beta/2$. We henceforth condition on this event.

We next need to bound the norm of the noise element z. Again, we utilize the isometric between $\mathcal{J}$ and $\mathbb{R}^k$ in ℓ_2 norm: recall that $\phi(z^t)$ is a Gaussian vector sampled from $\mathcal{N}(0, \sigma^2 I_k)$, by Lemma 3.10, with probability at least $1 - \beta/(2T)$,

$$\begin{aligned} \|z^t\|_2 &\leq \sigma \cdot (\sqrt{k} + \sqrt{\log(\frac{2T}{\beta})}) \\ &= \frac{\Delta_\infty(\sqrt{2rk\log(T/\delta)} + \sqrt{2r\log(T/\delta)\log(1/\gamma)})}{\epsilon'} \end{aligned} \tag{4}$$

union bound over T noise elements, it succeeds with probability at least $1 - \beta/2$, condition on this event happen. Note that the bound on $\|z^t\|_2$ (Eq. (4)) subsumes the error introduced by the oracle (Eq. (3)).

We proceed by assuming the norm of Gaussian noise is small, in particular,

$$\frac{\Delta_\infty(\sqrt{2rk\log(T/\delta)} + \sqrt{2r\log(T/\delta)\log(1/\gamma)})}{\epsilon'} \leq \frac{\alpha}{6}, \tag{5}$$

since $\alpha < 1$, this implies that the norm of the noise is at most $1/6$. Hence,

$$\begin{aligned} \|\widehat{\ell}^t\|_\infty &\leq \frac{\|\ell^t\|_\infty + \|z^t\|_\infty}{2} \\ &\leq \frac{\|\ell^t\|_\infty + \|z^t\|_2}{2} \\ &\leq 1. \end{aligned}$$

We can then apply Theorem A.5: pick $y \in \mathcal{K}$ be a feasible distribution element, then

$$\frac{1}{T}\sum_{t=1}^{T} \langle \widehat{\ell}^t, x^t \rangle \leq \frac{1}{T}\sum_{t=1}^{T} \langle \widehat{\ell}^t, y \rangle + \eta + \frac{\log r}{\eta T},$$

set $r_t := \frac{\langle a_{p^t}, x^t\rangle - b_{p^t}}{2} + \frac{\langle z^t, x^t\rangle}{2}$, we compute

$$\begin{aligned}\frac{1}{T}\sum_{t=1}^{T} r_t &= \frac{1}{T}\sum_{t=1}^{T}\langle \widehat{\ell}^t, x^t\rangle - \frac{b_{p^t}}{2} \\ &\leq \frac{1}{T}\sum_{t=1}^{T}\langle \widehat{\ell}^t, y\rangle - \frac{b_{p^t}}{2} + \eta + \frac{\log r}{\eta T}.\end{aligned}$$

Fix any constraint a_i, since we assume the error of the exponential mechanism is at most $\frac{2\Delta_\infty}{\epsilon'}\log(\frac{m}{\gamma})$ by Eq. (5) and x^t is a distribution, it must be the case that

$$\begin{aligned}\frac{1}{T}\sum_{t=1}^{T}\frac{\langle a_i, x^t\rangle - b_i}{2} + \frac{\langle z^t, x^t\rangle}{2} &\leq \frac{1}{T}\sum_{t=1}^{T} r_t + \frac{\alpha}{6} \\ &\leq \frac{1}{T}\sum_{t=1}^{T}\langle \widehat{\ell}^t, y\rangle - \frac{b_{p^t}}{2} + \eta + \frac{\log r}{\eta T} + \frac{\alpha}{6},\end{aligned}$$

where the first step is by Eq. (3). Recall that we also have shown that the norm of Gaussian noise is small and we could bound the term introduced by it:

$$\begin{aligned}|\langle z^t, x^t\rangle| &\leq \|x^t\|_1 \cdot \|z^t\|_\infty \\ &\leq \alpha/6,\end{aligned}$$

where the first step is by Hölder's inequality (Lemma A.2) and the second step is by Eq. (5). Thus, we have

$$\frac{1}{T}\sum_{t=1}^{T}\langle a_i, x^t\rangle - b_i \leq \frac{1}{T}\sum_{t=1}^{T} 2\langle \widehat{\ell}^t, y\rangle - b_{p^t} + \frac{\alpha}{2} + 2\eta + \frac{2\log r}{\eta T}.$$

It remains to examine $2\langle \widehat{\ell}^t, y\rangle - b_{p^t}$, since y is feasible, it must be that for any i, $\langle a_i, y\rangle - b_i \leq 0$, therefore

$$\begin{aligned}2\langle \widehat{\ell}^t, y\rangle - b_{p^t} &= \langle a_{p^t}, y\rangle + \langle z^t, y\rangle - b_{p_t} \\ &\leq \langle z^t, y\rangle \\ &\leq \|y\|_1 \cdot \|z^t\|_\infty \\ &\leq \alpha/6,\end{aligned}$$

put things together, we have

$$\begin{aligned}\frac{1}{T}\sum_{t=1}^{T}\langle a_i, x^t\rangle - b_i &= \langle a_i, \overline{x}\rangle - b_i \\ &\leq \frac{2\alpha}{3} + 2\eta + \frac{2\log r}{\eta T} \\ &\leq \alpha,\end{aligned}$$

where the last step is by the choice of η and T. It remains to provide a value of α that satisfies Eq. (5), and we can take

$$\begin{aligned}\alpha &\geq \frac{6\Delta_\infty(\sqrt{2rk\log(T/\delta)} + \sqrt{2r\log(T/\delta)\log(1/\gamma)})}{\epsilon'} \\ &\geq \frac{24\Delta_\infty\sqrt{rT}\cdot(\sqrt{k} + \sqrt{\log(1/\gamma)})\cdot\log(T/\delta)}{\epsilon} \\ &= \frac{288\Delta_\infty\sqrt{r\log r}\cdot(\sqrt{k} + \sqrt{\log(T/\beta)})\cdot\log(T/\delta)}{\alpha\epsilon},\end{aligned}$$

rearranging gives

$$\alpha \geq \frac{12\Delta_\infty^{1/2} r^{1/4}(\log r)^{1/4} k^{1/4}}{\epsilon^{1/2}} \cdot (\log\frac{288\log r}{\beta})^{1/4} \cdot (\log\frac{288\log r}{\delta})^{1/2}.$$

This completes the proof. □

Remark C.10. *Our algorithm differs significantly from the row private LP algorithm of [HRRU14], where they ensure the constraints are private by injecting Laplace noises to each entry. This comes from the fact that for $\mathcal{J} = \mathbb{R}^r$, the only Jordan frame is the standard basis, hence to develop a differential private mechanism, it is enough to perturb the eigenvalues. For other $\mathcal{J}$, this is no longer the case. Consider $\mathcal{J}$ to be the set of all $r \times r$ real symmetric matrices, then a Jordan frame can be formed by taking any set of orthonormal vectors $u_1, \ldots, u_r \in \mathbb{R}^r$ and computing $u_1 u_1^\top, \ldots, u_r u_r^\top$. This large degree of freedom means that private mechanism must also perturb the Jordan frame to ensure the basis information is preserved. We achieve so by implicitly resorting to the isomorphism between $\mathcal{J}$ and $\mathbb{R}^k$, as such a random Gaussian element would possess both random eigenvalues and Jordan frame. Note that we choose Gaussian mechanism instead of Laplace mechanism, as we will only provide either ℓ_2 or ℓ_1 norm guarantee instead of ℓ_∞, and ℓ_2 norm only distorts ℓ_∞ by a factor of $\sqrt{k}$ instead of k that is given by the ℓ_1 norm.*

C.4 Objective Private SCPs

Finally, we consider objective private SCP, where instead of solving a feasibility problem, we try to solve the optimization version. In contrast to [HRRU14], we again consider two neighboring objectives differ in ℓ_∞ norm.

$$\begin{aligned} \max_{x \in \mathcal{J}} \ & \langle c, x \rangle \\ \text{s.t. } & \langle a_i, x \rangle \leq b_i, \forall i \in [m] \\ & x \in \mathcal{K}. \end{aligned}$$

Given two neighboring databases D, D', we have $A(D) = A(D'), b(D) = b(D')$ and $\|c(D) - c(D')\|_\infty \leq \Delta_\infty$. Formally,

Definition C.11. *Given a vector $b \in \mathbb{R}^m$, $c \in \mathcal{K}$ and a constraint set $A \in \mathcal{J}^m$, a randomized mechanism $\mathcal{M}$ that outputs a vector in $\mathcal{J}$ is (ϵ, δ)-low sensitivity objective private if for any c, c' such that $\|c - c'\|_\infty \leq \Delta_\infty$,*

$$\Pr[\mathcal{M}(c, b, A) \in S] \leq e^\epsilon \Pr[\mathcal{M}(c', b, A) \in S] + \delta$$

for any subset $S \subseteq \mathcal{J}$.

Next, we present a simple algorithm, based similarly on the Gaussian mechanism introduced in Lemma 4.2, but applied to the objective element. We further assume a somewhat unusual but necessary condition on the SCP: there exists an optimal solution with unit ℓ_2 norm.

Theorem C.12. *Let the objective private SCP has optimal value* OPT *and the optimal solution has unit ℓ_2 norm. Suppose $\dim(\mathcal{J}) = k$ and $\phi : \mathcal{J} \to \mathbb{R}^k$ is an isomorphism between $\mathcal{J}$ and $\mathbb{R}^k$. Consider the following mechanism:*

- *Set $\sigma = \frac{\Delta_\infty \sqrt{2r \log(1/\delta)}}{\epsilon}$;*
- *Generate a Gaussian noise vector $\nu \sim \mathcal{N}(0, \sigma^2 I_k)$;*
- *Set $z = \phi^{-1}(\nu)$.*

Then, let $\widetilde{c} := c + z$, and the perturbed SCP

$$\begin{aligned} \max_{x \in \mathcal{J}} \ & \langle \widetilde{c}, x \rangle \\ \textit{s.t. } & \langle a_i, x \rangle \leq b_i, \forall i \in [m] \\ & \|x\|_2 = 1 \\ & x \in \mathcal{K} \end{aligned}$$

is released. Then, the algorithm is (ϵ, δ)-low sensitivity objective private with sensitivity Δ_∞. With probability $1 - \beta$, solving the perturbed SCP non-privately produces x^* such that $\langle a_i, x^* \rangle \leq b_i, \forall i \in [m]$ and $\langle c, x^* \rangle \geq \text{OPT} - \alpha$, where

$$\alpha = \frac{4\Delta_\infty \sqrt{r \log(1/\delta)}(\sqrt{k} + \sqrt{\log(1/\beta)})}{\epsilon}.$$

Proof. Recall that $\|c\|_2 \leq \sqrt{r} \cdot \|c\|_\infty$, by Corollary 4.4, indeed the mechanism is (ϵ, δ)-private. For accuracy, by Lemma 3.10, with probability at least $1 - \beta$,

$$\begin{aligned}\|z\|_2 &\leq \sigma \cdot (\sqrt{k} + \sqrt{\log(1/\beta)}) \\ &= \frac{\Delta_\infty(\sqrt{2rk\log(1/\delta)} + \sqrt{2r\log(1/\delta)\log(1/\beta)})}{\epsilon}.\end{aligned}$$

Suppose that

$$\frac{\alpha}{2} = \frac{\Delta_\infty(\sqrt{2rk\log(1/\delta)} + \sqrt{2r\log(1/\delta)\log(1/\beta)})}{\epsilon}.$$

Let $\widetilde{x}^*$ be the optimal solution to the perturbed SCP, and let x^* be the optimal solution to the original SCP. Note that if

$$\langle c, \widetilde{x}^* \rangle < \mathrm{OPT} - \alpha,$$

then

$$\begin{aligned}\langle \widetilde{c}, \widetilde{x}^* \rangle &= \langle c + \phi^{-1}(z), \widetilde{x}^* \rangle \\ &< \mathrm{OPT} - \alpha + \langle \phi^{-1}(z), \widetilde{x}^* \rangle \\ &\leq \mathrm{OPT} - \alpha + \|\phi^{-1}(z)\|_2 \cdot \|\widetilde{x}^*\|_2 \\ &= \mathrm{OPT} - \alpha/2,\end{aligned}$$

where we use ϕ is an isometry (can be achieved by picking an isomorphism with respect to an orthonormal basis), Cauchy-Schwarz inequality and the definition of $\alpha/2$. Meanwhile, note that

$$\begin{aligned}\langle \widetilde{c}, x^* \rangle &= \mathrm{OPT} + \langle \phi^{-1}(z), x^* \rangle \\ &\geq \mathrm{OPT} - \|z\|_2 \cdot \|x^*\|_2 \\ &= \mathrm{OPT} - \alpha/2,\end{aligned}$$

contradicts the definition of x^*. Thus, it must be the case that $\langle c, \widetilde{x}^* \rangle \geq \mathrm{OPT} - \alpha$, and $\widetilde{x}^*$ is feasible since it's a feasible solution to the perturbed SCP, and the perturbation does not change feasibility. This completes the proof. □

Remark C.13. *In [HRRU14], they impose a much standard assumption that $\|x\|_1 = 1$. This again comes from the fact that for $\mathcal{J} = \mathbb{R}^k$, the isomorphism is just the identity map, and in addition to the isometry in ℓ_2 norm $\|\phi(x)\|_2 = \|x\|_2$, all norms are preserved. This is particularly important when applying Hölder's inequality: when $\|x\|_1 = 1$, we could upper or lower bound the inner product by the ℓ_∞ norm of the noise z. This is no longer true for $\mathcal{J} \neq \mathbb{R}^k$, as $\phi(\cdot)$ only preserves the ℓ_2 norm, translating between different norms would incur blowup or shrinkage factors dependent on r. Of course, imposing an ℓ_2 norm constraint makes the constraint set no longer an affine subspace, which would require solvers that could handle quadratic constraints.*

D Impact Statement

Our work concerns the privacy of symmetric cone programming, we believe its development would help protect the data and model privacy for various machine learning tasks. At its current stage, our work is theoretical, so it does not lead to any direct negative societal consequences.