# A Private and Computationally-Efficient Estimator for Unbounded Gaussians

**Gautam Kamath** G@CSAIL.MIT.EDU
**Argyris Mouzakis** AMOUZAKI@UWATERLOO.CA
**Vikrant Singhal** VIKRANT.SINGHAL@UWATERLOO.CA
*Cheriton School of Computer Science, University of Waterloo*

**Thomas Steinke** BADGAUSS@THOMAS-STEINKE.NET
*Google Research, Brain Team*

**Jonathan Ullman** JULLMAN@CCS.NEU.EDU
*Khoury College of Computer Sciences, Northeastern University*

**Editors:** Po-Ling Loh and Maxim Raginsky

## Abstract

We give the first polynomial-time, polynomial-sample, differentially private estimator for the mean and covariance of an arbitrary Gaussian distribution $\mathcal{N}(\mu, \Sigma)$ in $\mathbb{R}^d$. All previous estimators are either nonconstructive, with unbounded running time, or require the user to specify a priori bounds on the parameters $\mu$ and $\Sigma$. The primary new technical tool in our algorithm is a new differentially private preconditioner that takes samples from an arbitrary Gaussian $\mathcal{N}(0, \Sigma)$ and returns a matrix $A$ such that $A\Sigma A^T$ has constant condition number.

**Keywords:** Privacy, learning, Gaussian.

## 1. Introduction

All useful statistical estimators have the side effect of revealing information about their sample, which leads to concerns about the *privacy* of the individuals who contribute their data to the sample. In this work we study statistical estimation with the constraint of *differential privacy (DP)* (Dwork et al., 2006), a rigorous individual privacy criterion well suited to statistical estimation and machine learning.

As in classical statistical estimation, it is impossible to privately estimate even basic statistics like the mean and covariance without some restrictions on the distribution, although the assumptions made in the private setting are typically stronger both qualitatively and quantitatively. To provide some intuition for the assumptions required for private estimation, consider the simple problem of privately estimating the mean of a distribution $\mathcal{D}$ over $\mathbb{R}^d$ from a set of $n$ samples $X_1, \ldots, X_n \sim \mathcal{D}$. The standard way to solve this problem is by computing a noisy empirical mean $\hat{\mu} = \frac{1}{n}\sum_{i=1}^n X_i + Z$, where $Z$ is a suitable random variable—typically Gaussian or Laplacian. The magnitude of $Z$ must be proportional to the *sensitivity* of $\frac{1}{n}\sum_{i=1}^n X_i$, which measures how much its value can change if a single point $X_i$ is modified arbitrarily. Without further information about the underlying distribution, the sensitivity is infinite, rendering this naïve approach ineffective.

To facilitate using a low-sensitivity mean estimator, we generally make two types of assumptions on the underlying distribution $\mathcal{D}$:

1. The distribution $\mathcal{D}$ is somehow well-behaved. For example, we assume $\mathcal{D}$ is a Gaussian distribution $\mathcal{N}(\mu, \Sigma)$, while other works have assumed weaker moment bounds (Barber and Duchi, 2014; Bun and Steinke, 2019; Kamath et al., 2020).

2. The analyst has some prior knowledge about the parameters of the distribution $\mathcal{D}$. The standard assumption in this setting is that the analyst knows parameters $R > 0$ and $K > 0$ such that $\|\mu\|_2 \leq R$ and $\mathbb{I} \preceq \Sigma \preceq K\mathbb{I}$.[1][2]

These assumptions ensure that we can identify a finite subset of the domain that contains all the samples with high probability, which we can use to find a proxy for the empirical mean with finite sensitivity.

The first style of assumption is common and generally necessary to provide non-trivial guarantees even in the non-private setting. The second style of assumption however is particular to the private setting, and forces the analyst to input some prior knowledge about the location and shape of their data. This may be a minimal burden to place on the user when the domain is familiar, but can be unreasonable for unfamiliar, high-dimensional domains. In that case the analyst may only be able to give extremely loose bounds, corresponding to very large values of $R$ and $K$. This leads to a degradation of the accuracy of the final output.

For these reasons, a key goal in private algorithm design is minimizing the sample complexity's dependence on the prior knowledge in the form of the parameters $R$ and $K$. Naïve algorithms limit the empirical estimator's sensitivity by simply clipping the data based on the analyst's prior knowledge, incurring an undesirable linear dependence on $R$ and $K$. More clever approaches iteratively refine the analyst's knowledge of the shape and location of the distribution. That is, we start by finding a weak estimate of the parameters $\mu$ and $\Sigma$, which allows us to rescale the data and thereby reduce the parameters $R$ and $K$ for the next steps. This approach results in improved sample complexity compared to the naïve strategy outlined above: For the univariate case, it can be used to eliminate the dependence on $R$ and $K$ entirely (Karwa and Vadhan, 2018). For the multivariate case, existing approaches yield a polylogarithmic dependence on $R$ and $K$ (Kamath et al., 2019a)—an exponential improvement—but do not eliminate the need for a priori bounds.

Despite exponential improvements, it is natural to wonder whether a dependence on $R$ and $K$ is necessary at all. For more restrictive special cases of differential privacy, such as pure or concentrated differential privacy,[3] packing lower bounds imply that a polylogarithmic dependence is the best possible (Bun and Steinke, 2016; Bun et al., 2019). However, these lower bounds do not apply to the most general notion of approximate differential privacy, and in this model we can often eliminate the need for any a priori bounds on the distribution, which is clearly an appealing feature of an estimator.

For mean estimation, it is relatively easy to eliminate the need for a priori bounds on the mean (the parameter $R$), but the rich geometric structure of covariance matrices makes it much more challenging to eliminate the need for bounds on the covariance (the parameter $K$), even without requiring computational efficiency. Recently, building on a cover-based technique of Bun et al.

1. Here, $A \preceq B$ refers to the PSD order denoting that $x^T Ax \leq x^T Bx$ for every $x \in \mathbb{R}^d$, and $\mathbb{I}$ denotes the identity matrix.
2. By translating and rescaling the distribution, these assumptions can be relaxed to $\|\mu - c\|_2 \leq R$ for some known vector $c$ and $\mathbb{I} \preceq A\Sigma A^T \preceq K\mathbb{I}$ for some known matrix $A$.
3. Though we later define the various relevant notions of DP, we remind the reader that pure $(\varepsilon, 0)$-DP is stronger than concentrated DP, which in turn is stronger than approximate $(\varepsilon, \delta)$-DP.

(2019), Aden-Ali et al. (2021a) show the existence of an estimator that doesn't require any bounds on the covariance matrix, but their argument is non-constructive and does not give an estimator with polynomial, or even finite running time.

### 1.1. Results

Our main result is a polynomial-time algorithm for Gaussian estimation which requires no prior knowledge about the distribution parameters.

**Theorem 1 (Informal)** *There is a polynomial-time $(\varepsilon, \delta)$-differentially private estimator $M$ with the following guarantee: For every $\mu \in \mathbb{R}^d$ and positive semidefinite $\Sigma \in \mathbb{R}^{d\times d}$, if $X_1, \ldots, X_n \sim \mathcal{N}(\mu, \Sigma)$ and $n \geq \tilde{O}\left(\frac{d^2}{\alpha^2} + \frac{d^2 \cdot \mathrm{polylog}(1/\delta)}{\alpha\varepsilon} + \frac{d^{5/2}\cdot \mathrm{polylog}(1/\delta)}{\varepsilon}\right)$, then, with high probability, $M(X_1, \cdots, X_n)$ outputs $\hat{\mu} \in \mathbb{R}^d$ and $\hat{\Sigma} \in \mathbb{R}^{d\times d}$ such that $\|\hat{\Sigma} - \Sigma\|_\Sigma := \|\Sigma^{-1/2}\hat{\Sigma}\Sigma^{-1/2} - \mathbb{I}\|_F \leq \alpha$ and $\|\hat{\mu} - \mu\|_\Sigma := \|\Sigma^{-1/2}\hat{\mu} - \Sigma^{-1/2}\mu\|_2 \leq \alpha$. In particular, this guarantee implies that $\mathcal{N}(\hat{\mu}, \hat{\Sigma})$ and $\mathcal{N}(\mu, \Sigma)$ are $O(\alpha)$-close in total variation distance.*

The main advantage of our result compared to prior work is that our estimator both runs in polynomial time and requires no prior bounds on $\Sigma$, whereas all estimators from prior work lack at least one of these properties. The best known sample complexity is the result of Aden-Ali et al. (2021a), which is $n = O(d^2/\alpha^2 + d^2/\alpha\varepsilon + \log(1/\delta)/\varepsilon)$. Our estimator has a slightly worse dependence on the dimension $d$, but our running time is polynomial instead of unbounded, and it remains open to find a polynomial-time estimator with information-theoretically optimal sample complexity. Their bound is conjectured to be tight, but matching lower bounds under $(\varepsilon, \delta)$-DP are only known for the first and third terms. A lower bound of $\Omega(d^2/\alpha\varepsilon)$ has only been proven under $(\varepsilon, 0)$-DP. See Section 1.1.1 of Aden-Ali et al. (2021a) for more discussion on lower bounds. See Table 1 for more information on prior upper bounds.

Several concurrent works have appeared after the preprint of our work, which also achieve similar results. See the discussion of Simultaneous and Subsequent Work in Section 1.3.

### 1.2. Overview of Techniques

Our algorithm builds on the *private preconditioning* framework introduced in Kamath et al. (2019a). Here our goal is to privately obtain a matrix $A$ such that, after rescaling, $\mathbb{I} \preceq A\Sigma A^T \preceq O(1)\cdot\mathbb{I}$. The preceding statement implicitly assumes that $\Sigma$ is full rank, which is useful to simplify the discussion, but our methods also handle the more general case of a degenerate covariance matrix $\Sigma$. Given such a matrix $A$, we can perform the invertible transformation of replacing each sample $X_i$ with $AX_i$ and then apply the naïve private estimator to these transformed samples and finally invert the transformation to obtain our estimates $\hat{\mu}$ and $\hat{\Sigma}$. Since $X \sim \mathcal{N}(\mu, \Sigma)$ implies $AX \sim \mathcal{N}(A\mu, A\Sigma A^T)$, we now have a good a priori bound on the covariance $A\Sigma A^T$ and, hence, the naïve estimator will have small sample complexity.

The main technical ingredient in our estimator is a new *private preconditioner* that takes samples of the form $X \sim \mathcal{N}(0, \Sigma)$, for an arbitrary $\Sigma$, and outputs a matrix $A$ so that $A\Sigma A^T$ is well conditioned.[4]

4. Without loss of generality, we can restrict our attention to the case where the data is drawn from $\mathcal{N}(\mu, \Sigma)$ with $\mu = 0$. If we are given two independent samples $X, X' \sim \mathcal{N}(\mu, \Sigma)$, then $(X - X')/\sqrt{2}$ has the distribution of $\mathcal{N}(0, \Sigma)$.

| Reference | Sample Complexity | Computational Complexity |
|---|---|---|
| Non-Private | $\frac{d^2}{\alpha^2}$ | Polynomial |
| Naïve Estimator | $\frac{d^2}{\alpha^2} + \frac{Kd^2}{\alpha\varepsilon}$ | Polynomial |
| Kamath et al. (2019a) | $\frac{d^2}{\alpha^2} + \frac{d^2}{\alpha\varepsilon} + \frac{d^{3/2}\log^{1/2} K}{\varepsilon}$ | Polynomial |
| Aden-Ali et al. (2021a) | $\frac{d^2}{\alpha^2} + \frac{d^2}{\alpha\varepsilon}$ | Unbounded |
| Theorem 1 (this work) | $\frac{d^2}{\alpha^2} + \frac{d^2}{\alpha\varepsilon} + \frac{d^{5/2}}{\varepsilon}$ | Polynomial |
| Ashtiani and Liaw (2021) (concurrent) | $\frac{d^2}{\alpha^2} + \frac{d^2}{\alpha\varepsilon}$ | Polynomial |
| Kothari et al. (2021) (concurrent) | $\frac{d^8}{\alpha^4\varepsilon^8}$ | Polynomial |

Table 1: Comparing $(\varepsilon, \delta)$-differentially private covariance estimators for $\mathcal{N}(0, \Sigma)$. Here, $d$ is the dimension, $K$ is an a priori bound such that $\mathbb{I} \preceq \Sigma \preceq K\mathbb{I}$, and the accuracy guarantee is $\|\hat{\Sigma} - \Sigma\|_\Sigma \leq \alpha$. The sample-complexity bounds suppress polylogarithmic factors in $d$, $\frac{1}{\alpha}$, and $\frac{1}{\delta}$.

**Theorem 2 (Informal)** *There is a polynomial-time $(\varepsilon, \delta)$-differentially private algorithm $M$ with the following guarantee: For every positive-semidefinite, rank-$k$ matrix $\Sigma \in \mathbb{R}^{d\times d}$, if $X_1, \ldots, X_n \sim \mathcal{N}(0, \Sigma)$ and $n \geq \tilde{O}\left(\frac{d^{5/2}\cdot \mathrm{polylog}(1/\delta)}{\varepsilon}\right)$, then, with high probability, $M(X_1, \ldots, X_n)$ outputs $A \in \mathbb{R}^{d\times d}$ such that $\frac{\lambda_1(A\Sigma A^T)}{\lambda_k(A\Sigma A^T)} = O(1)$, where we write $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_d$ for the sorted eigenvalues of the matrix.*

To contrast Theorem 2 with that of Kamath et al. (2019a), their work gave a polynomial-time algorithm that takes samples from a Gaussian $\mathcal{N}(0, \Sigma)$ such that $\mathbb{I} \preceq \Sigma \preceq K\mathbb{I}$ and returns a matrix $A$ such that $\mathbb{I} \preceq A\Sigma A^T \preceq \frac{K}{2}\mathbb{I}$. Thus, iteratively applying their preconditioner $O(\log K)$ times and using composition bounds for differential privacy gives a result similar to Theorem 2, but with a $(\log K)^{1/2}$ term in the sample complexity. In contrast, very informally, our preconditioner is able to find a good estimate of $\Sigma$ one direction at a time, no matter how poorly conditioned $\Sigma$ is, so the number of iterations depends only on the dimension $d$ and not on any assumptions about $\Sigma$ itself.

Since the preconditioner of Kamath et al. (2019a) can already handle the case where the condition number $K$ is small or moderately large, the main technical hurdle that our work must overcome is the case where the condition number is very large, specifically exponential: $\lambda_d(\Sigma)/\lambda_1(\Sigma) \leq \exp(-\mathrm{poly}(d))$. When the eigenvalues of $\Sigma$ are so spread out, there must be a large eigenvalue gap where $\lambda_{k+1}(\Sigma)/\lambda_k(\Sigma)$ is very small, at most inverse-polynomial in $d$. Thus, the key technical ingredient we need is a private algorithm that can output an approximation to the $k$-dimensional subspace of $\Sigma$ containing the directions of large variance. Given such a subspace, we can partition the space into a $k$-dimensional subspace where the covariance is well conditioned and a lower-dimensional subspace, and then recur on the lower-dimensional subspace. This *private subspace recovery* problem has been investigated before, originally by Dwork et al. (2014), and, recently Singhal and Steinke (2021) gave an algorithm for this problem that gives dimension-independent sample complexity under the assumption of a large eigenvalue gap between the top-$k$ subspace and its complement. In order to apply their algorithm in our setting, we give a different analysis, and

along the way we make other modifications that, for our application, reduce the sample complexity by polynomial factors in the dimension.

**Theorem 3 (Informal, extension of Singhal and Steinke (2021))** *There is a polynomial-time $(\varepsilon, \delta)$-differentially private algorithm $M$ with the following guarantee: Let $\Sigma \in \mathbb{R}^{d\times d}$ such that $\lambda_{k+1}(\Sigma)/\lambda_k(\Sigma) < \gamma^2$ for some $1 \le k < d$ and $0 < \gamma \le 1$, and let $\Pi \in \mathbb{R}^{d\times d}$ be the matrix that projects onto the subspace spanned by the top-$k$ eigenvectors of $\Sigma$. If $0 < \psi \le 1$ and $X_1, \dots, X_n \sim \mathcal{N}(0, \Sigma)$ and $n \ge \tilde{O}\left(\frac{d^{3/2}k^{1/2}\cdot \mathrm{polylog}(1/\delta)}{\psi^2 \varepsilon}\right)$, then with high probability, $M(X_1, \dots, X_n)$ outputs a projection matrix $\hat{\Pi} \in \mathbb{R}^{d\times d}$ such that $\|\hat{\Pi} - \Pi\|_2 \le \psi\gamma$.*

The subspace recovery algorithm of Singhal and Steinke (2021) is tailored to allow a dimension-independent sample complexity, which is something that our modifications no longer achieve. However, in our setting, a direct application of their algorithm would be inefficient in terms of the sample complexity. Here, we are free to pick $\mathrm{poly}(d)$ samples, which gives us the option to use more accurate methods in the subspace recovery algorithm – we trade $\mathrm{poly}(d)$ sample complexity for improved accuracy. In particular, we incorporate the ball-finding algorithm of Nissim et al. (2016). Roughly speaking, if the eigengap is $\gamma^2$, then to get an error proportional to $\gamma$, Singhal and Steinke (2021) would require $O(d^2k^2)$ samples, while our modifications reduce this cost to $O(d^{3/2}k^{1/2})$.

### 1.3. Related Work

Differentially private statistical inference has been an active area of research for over a decade (e.g. Dwork and Lei (2009); Vu and Slavković (2009); Wasserman and Zhou (2010); Smith (2011)), and the literature is too broad to fully summarize here. Our work fits into two more recent trends that we survey below—designing private estimators without the need for strong prior bounds and pinning down the minimax sample complexity for differentially private estimation.

**Private Estimation without Prior Knowledge.** The influential work Karwa and Vadhan (2018) focused attention on minimizing the need for prior knowledge as a key issue for obtaining practical private estimators, providing both algorithms and lower bounds for univariate Gaussian mean and variance estimation. In particular, they designed pure DP estimators with a logarithmic dependence on the bounding parameters using a general recipe based on private histograms, and estimators with approximate DP with no dependence on these parameters. Subsequent works gave other pure DP or concentrated DP algorithms for the univariate case with a similar logarithmic dependence, based on techniques such as the exponential mechanism (Du et al., 2020), iteratively shrinking confidence intervals (Biswas et al., 2020), the trimmed mean (Bun and Steinke, 2019), and quantile estimation (Huang et al., 2021). Other techniques have been employed to deal with the bounding parameters for univariate median estimation (Avella-Medina and Brunel, 2019; Tzamos et al., 2020), including propose-test-release (Dwork and Lei, 2009) and efficient Lipschitz extensions (Cummings and Durfee, 2020; Tzamos et al., 2020).

All the above techniques for univariate mean estimation extend to multivariate mean estimation with known covariance, simply by applying a univariate estimator to each coordinate, however extending to multivariate covariance estimation is significantly more challenging. Kamath et al. (2019a) gave the first algorithm for this setting which satisfies concentrated DP or approximate DP, and incurs only a logarithmic dependence on the bounding parameters, which was subsequently refined into a more practical variant (Biswas et al., 2020). Bun et al. (2019) provides a cover-based

approach which leads to pure DP algorithms for more general settings with logarithmic dependence on the bounding parameters, but the estimators have exponential running time or worse. They further provide an approach for proving approximate DP sample complexity bounds which require no bounding parameters, contingent on the construction of a locally-sparse cover. As they describe it, their method has an infinite running time, and they are also only able to construct such a cover for multivariate Gaussians with known covariance, as the rich geometric structure makes the unknown covariance case hard to reason about. Aden-Ali et al. (2021a) extends this approach to require only a collection of sparse local covers, allowing them to prove a bound on the sample complexity of covariance estimation with no bounding parameters. Again, their approach does not provide even a finite-time algorithm, and our result is the first polynomial-time algorithm for covariance estimation with no dependence on the bounding parameters. Recent work (Brown et al., 2021) provides an approach for Gaussian mean estimation with unknown covariance, which bypasses the problem of covariance estimation to obtain better sample complexity. Specifically, they provide a computationally-inefficient approximate DP algorithm which requires no parameter knowledge. Since our goal is to estimate the covariance, their results are inapplicable to our setting.

**Minimax Sample Complexity.** Our work also falls into a broader line of work on minimax sample complexities for differentially private statistical estimation. See Kamath and Ullman (2020) for a partial survey of this line of work. The first minimax sample complexity bounds to show an asymptotic separation between private and non-private estimation for private mean estimation were proven in Bun et al. (2014), and subsequently sharpened and generalized in several respects (Dwork et al., 2015; Bun et al., 2017; Steinke and Ullman, 2017a,b; Kamath et al., 2019a). More recently, Cai et al. (2019) extended these bounds to sparse estimation and regression problems. Acharya et al. (2021) provides an alternative, user-friendly approach to proving sample complexity bounds, which is directly analogous to the classical approaches for proving minimax lower bounds in statistics. These approaches are less powerful in general, but yields tight bounds for certain statistical estimation tasks.

There are a wide variety of results pinning down the minimax sample complexity for estimation under a variety of distributional assumptions, including settings with heavy-tailed data (Barber and Duchi, 2014; Bun and Steinke, 2019; Kamath et al., 2020; Wang et al., 2020; Kamath et al., 2021; Hopkins et al., 2022), mixtures of Gaussians (Kamath et al., 2019b; Aden-Ali et al., 2021b), graphical models (Zhang et al., 2020), and discrete distributions (Diakonikolas et al., 2015). Additionally, Liu et al. (2021a,b); Hopkins et al. (2022) give algorithms for mean estimation which are simultaneously private and robust. Some recent works (Liu et al., 2020; Levy et al., 2021) focus on estimation in a setting where a single person may contribute multiple samples (but privacy must still be provided with respect to all of a person's records). One work (Avent et al., 2019) studies mean estimation in a hybrid model where some users require the more stringent local DP property, while other are content with central DP.

**Simultaneous and Subsequent Work.** The initial online posting of this work was accompanied by a flurry of simultaneous and independent papers featuring results on private covariance estimation. Most directly comparable with our work are the simultaneous and independent results of Ashtiani and Liaw (Ashtiani and Liaw, 2021), and Kothari, Manurangsi, and Velingker (Kothari et al., 2021), which obtain computationally-efficient algorithms for private estimation of unbounded Gaussians. Both are also robust to adversarial corruptions. The techniques of all three works differ from each other, and thus offer multiple perspectives on how to address this problem. While our work employs

ideas from private subspace recovery, Ashtiani and Liaw (2021) uses a framework based on privately checking whether the results of several non-private estimates resemble each other (a la Propose-Test-Release (Dwork and Lei, 2009)), and Kothari et al. (2021) privately adapts convex relaxations which have recently seen use in robust statistics. Focusing on the dependence on the dimension $d$, our algorithm has sample complexity $\tilde{O}(d^{2.5})$, while Ashtiani and Liaw (2021) is $\tilde{O}(d^2)$ and Kothari et al. (2021) is $\tilde{O}(d^8)$.

Also simultaneous to all these works, Tsfadia, Cohen, Kaplan, Mansour, and Stemmer (Tsfadia et al., 2021) provided a framework similar to that of Ashtiani and Liaw's (Ashtiani and Liaw, 2021), and applied it to the problem of mean estimation. In a subsequent update, Tsfadia et al. (2021) showed that their approach too can give an efficient (non-robust) private algorithm for estimation of unbounded Gaussian covariances.

Finally, simultaneous and independent to our work, Liu, Kong, and Oh (Liu et al., 2021b) give a framework for designing private estimators via connections with robustness. For the specific case of Gaussian covariance estimation, they give a computationally inefficient algorithm with similar guarantees as the work of Aden-Ali, Ashtiani, and Kamath (Aden-Ali et al., 2021a).

### 1.4. Organization of the Paper

We start with our main procedure to learn Gaussian covariances, which puts all of our techniques together, in Section 2. Then we state our novel and the most important component for this process – the private preconditioner – in Section 3. We give standard background on differential privacy and concentration-of-measure in Appendix A. After that, we present the algorithm for private eigenvalue estimation in Appendix B. It is followed by our extended subspace-recovery algorithm in Appendix C. We describe the final subroutine for our main algorithms, the naïve estimator, in Appendix D.

## 2. Our Estimator

In this section, we state our new estimator for Gaussian covariances (Algorithm 1) that we call, "GaussianCovarianceEstimator", which uses our novel preconditioning technique from Section 3, and the naive estimator. The algorithm first makes the Gaussian well-conditioned using the preconditioner (Algorithm 4), followed by estimating it using the naive estimator (Algorithm 7), and then it applies the inverse transformation of the preconditioning matrix. The following is the main result of the section. Then using that and Lemma 15, we would be able to conclude that $d_{TV}(\mathcal{N}(\mu,\Sigma),\mathcal{N}(\hat{\mu},\widehat{\Sigma})) \leq \alpha$.

**Theorem 4** *Let $\Sigma \in \mathbb{R}^{d\times d}$ be a symmetric, PSD matrix and $\mu \in \mathbb{R}^d$. Then for all $\varepsilon,\delta,\alpha,\beta > 0$, there exists an $(\varepsilon,\delta)$-DP algorithm that takes $n \geq \widetilde{O}\left(\frac{d^2}{\alpha^2} + \frac{d^2}{\varepsilon\alpha} + \frac{d^{2.5}}{\varepsilon}\right)$ samples from $\mathcal{N}(\mu,\Sigma)$, and outputs a symmetric, PSD matrix $\widehat{\Sigma} \in \mathbb{R}^{d\times d}$ and $\hat{\mu} \in \mathbb{R}^d$, such that with probability at least $1-O(\beta)$, $\|\Sigma-\widehat{\Sigma}\|_\Sigma \leq \alpha$ and $\|\hat{\mu}-\mu\|_\Sigma \leq \alpha$. In the above, $\widetilde{O}$ hides factors of* $\mathrm{polylog}(d,\frac{1}{\varepsilon},\frac{1}{\delta},\frac{1}{\beta})$.

**Proof** In our estimator, Algorithm 1 is one of the main components that is used to estimate the covariance of the Gaussian. The other component is the approximate DP version of the private mean estimation algorithm (PME) from Kamath et al. (2019a). We replace the preconditioning matrix in PME by our DP preconditioner that we obtain from running Algorithm 1. To prove the theorem, it is enough to show the privacy and accuracy guarantees of Algorithm 1.

Privacy follows from the privacy guarantees of Algorithm 7 (Lemma 27), Algorithm 4 (Theorem 7), and the approximate DP version of PME (Kamath et al., 2019a), followed by composition (Lemma 10) and post-processing (Lemma 9).

Now, we prove the first accuracy statement. Let $Y$ be the original dataset with $2n$ samples chosen i.i.d. from $\mathcal{N}(\mu, \Sigma)$. We construct the dataset $X$ as follows: for each $i \in [n]$, set $X_i = \frac{Y_{2i} - Y_{2i-1}}{\sqrt{2}}$. Then each $X_i$ is an independent sample from $\mathcal{N}(0, \Sigma)$. We then supply the dataset $X$ to Algorithm 1. Note that $AX$ contains points from $\mathcal{N}(0, A\Sigma A)$ by construction. This means that $\frac{\lambda_d(A\Sigma A)}{\lambda_1(A\Sigma A)} \geq \Omega(1)$. Thus, by the accuracy guarantees of NaiveEstimator (Theorem 29), we have $\|\Sigma' - A\Sigma A\|_{A\Sigma A} \leq O(\alpha)$. However, $\|\Sigma' - A\Sigma A\|_{A\Sigma A} = \|\widehat{\Sigma} - \Sigma\|_{\Sigma}$. This gives us the first result.

The mean estimation result follows from the accuracy guarantees of PME, to which we supply the dataset $Y$. Note that PME is designed to provide zCDP (Bun and Steinke, 2016) and has a polylogarithmic dependence on the range parameter $R$ that bounds the magnitude of the true mean. The goal is to eliminate that dependence, which is only possible under approximate DP. The approximate DP version of this that doesn't have any dependence on $R$ can be obtained by using the approximate DP version of Karwa and Vadhan (2018) that utilises stability based histograms. With a multiplicative cost in the sample complexity in terms of $\mathrm{polylog}(1/\delta)$, this would establish the result that we need. ■

**Algorithm 1:** Differentially Private $\mathrm{GaussianCovarianceEstimator}_{\varepsilon,\delta,\alpha,\beta}(X)$

**Input:** Samples $X_1, \ldots, X_n \in \mathbb{R}^d$. Parameters $\varepsilon, \delta, \alpha, \beta > 0$.
**Output:** Matrix $\widehat{\Sigma} \in \mathbb{R}^{d \times d}$.

`// Precondition the covariance.`
Set $A \leftarrow \mathrm{Preconditioner}_{\varepsilon,\delta,\beta}(X)$.

`// Estimate the transformed covariance.`
Set $\Sigma' \leftarrow \mathrm{NaiveEstimator}_{\varepsilon,\delta,\beta}(AX)$.

`// Revert to the original space.`
Set $\widehat{\Sigma} \leftarrow A^{-1}\Sigma' A^{-1}$.

**return** $\widehat{\Sigma}$.

### 2.1. Handling the Degenerate Case

So far, we have implicitly assumed that all the eigenvalues of $\Sigma$ are strictly greater than 0. Here, we talk about the case where some of the eigenvalues of $\Sigma$ could be 0. Let $k \in [d]$ be the largest number such that the $k$-th eigenvalue of $\Sigma$ is non-zero. Then we can use Algorithm 6 to exactly recover the top $k$ subspace, and project onto that subspace, and run GaussianCovarianceEstimator within that subspace. To elaborate, this can be done in three steps: (1) detecting the non-zero eigenvalues of $\Sigma$ using Algorithm 5; (2) finding the true subspace of $\Sigma$ using Algorithm 6, which can exactly recover the subspace at a cost of $\widetilde{O}(d^2/\varepsilon)$ in the sample complexity; and (3) running Algorithm 1 on the points projected on to that subspace.

## 3. Private Preconditioning

In this section, we develop a preconditioning technique that does not rely on knowledge of a priori bounds on the eigenvalues of the covariance matrix of the underlying distribution. It is the main preprocessing step that makes the Gaussian covariance almost spherical. For the following, we assume that the eigenvalues of the covariance matrix $\Sigma$ are examined in non-increasing order $\lambda_1 \geq \cdots \geq \lambda_d > 0$.

### 3.1. Coarse Preconditioning

We describe here the function of the "coarse" preconditioner which, along with Algorithm 6, constitutes the main technical novelty of our approach. The purpose served by this subroutine is to reduce gaps between consecutive eigenvalues (say $\lambda_k$ and $\lambda_{k+1}$). Observe that, our only assumptions are that the ratio $\frac{\lambda_{k+1}(\Sigma)}{\lambda_k(\Sigma)}$ is below some threshold and that the eigenvalues that come before exhibit no significant gaps ($\frac{\lambda_k}{\lambda_1}$ is lower bounded appropriately, implying that it is larger than some absolute constant). The first condition essentially prohibits us from using the preconditioning technique from Kamath et al. (2019a), since we do not know how large the gap between $\lambda_k$ and $\lambda_{k+1}$ may be. Instead, the algorithm uses our adaptation of the subspace algorithm of Singhal and Steinke (2021) (see Algorithm 6) in order to approximate the subspace that corresponds to the eigenvalues that come before the gap. Specifically, we obtain projection matrices $\Pi_V$ onto a subspace $V$ and $\Pi_{V^\perp} = \mathbb{I} - \Pi_V$ onto its complement $V^\perp$, such that these matrices are close in spectral norm to the projections onto the top $k$ eigenspace of $\Sigma$ and its complement. Rescaling our data by a matrix of the form $A = x\Pi_V + y\Pi_{V^\perp}$ roughly results in the eigenvalues of the covariance matrix corresponding to $V$ and $V^\perp$ being rescaled by $x^2$ and $y^2$, respectively. Setting the scalars $x$ and $y$ appropriately will reduce the eigenvalue gap, even if the subspace $V$ is not perfectly aligned with the top $k$ eigenvalues. Interestingly, if the eigengap is large (i.e., the ratio $\frac{\lambda_{k+1}(\Sigma)}{\lambda_k(\Sigma)}$ is small), then our algorithm works just as well as when it is small. This is because the subspace recovery subroutine will become more accurate in this setting as it outputs a projection matrix, whose error scales with this gap. Note that this step reduces the eigengap to a large extent, but does not exactly get us in the range that we would desire, that is, the gap between the 1-st and the $(k+1)$-th eigenvalues is greatly reduced, but it is still not small enough to maintain the loop invariant of Algorithm 4, which says that in iteration $i$, the gap between the 1-st and the $i$-th eigenvalues is bounded. We address this issue in Section 3.2.

Having described the algorithm above, we now present the corresponding pseudocode, followed by its analysis.

**Algorithm 2:** Differentially Private $\mathrm{CoarsePreconditioner}_{\varepsilon,\delta,\beta,k,\hat{\gamma}}(X)$

**Input:** Samples $X_1, \ldots, X_n \in \mathbb{R}^d$. Parameters $\varepsilon, \delta, \beta, k > 0, \hat{\gamma} \geq 0$.
**Output:** Matrix $A \in \mathbb{R}^{d \times d}$.

Set $1 - \eta \leftarrow \hat{\gamma}$.
Set $\hat{\Pi}_{1:k} \leftarrow \mathrm{SubspaceRecovery}_{\varepsilon,\delta,\beta,k,\hat{\gamma}}(X)$ and $\hat{\Pi}_{k+1:d} \leftarrow \mathbb{I} - \hat{\Pi}_{1:k}$.
Set $A \leftarrow (1-\eta)\hat{\Pi}_{1:k} + \hat{\Pi}_{k+1:d}$.

**return** $A$.

**Theorem 5 (Coarse Preconditioner)** *Let $0 < \overline{\gamma} \leq 1$ and $0 < \hat{\gamma} < 1$ be arbitrary parameters. Then for all $\varepsilon, \delta, \beta > 0$ and $n \geq O\left(\frac{d^2 \cdot \mathrm{polylog}(d, \frac{1}{\varepsilon}, \frac{1}{\delta}, \frac{1}{\beta})}{\varepsilon \overline{\gamma}^4}\right)$, there exists an $(\varepsilon, \delta)$-DP algorithm, such that the following holds. Let $X = (X_1, \ldots, X_n)$ be i.i.d. samples from $\mathcal{N}(0, \Sigma)$, where, for some $1 \leq k < d$, $\frac{\lambda_k(\Sigma)}{\lambda_1(\Sigma)} \geq \overline{\gamma}^2$, and $\gamma^2 := \frac{\lambda_{k+1}(\Sigma)}{\lambda_k(\Sigma)} \in \left[\frac{\hat{\gamma}^2}{4}, 4\hat{\gamma}^2\right]$. Then with probability at least $1 - O(\beta)$, the algorithm takes $X$ and $\hat{\gamma}$ as input, and outputs $A \in \mathbb{R}^{d \times d}$ that satisfies $\frac{\lambda_{k+1}(A\Sigma A)}{\lambda_1(A\Sigma A)} \geq \frac{\overline{\gamma}^2}{40}$.*

**Proof** We prove the privacy and accuracy guarantees of Algorithm 2. Privacy follows from the privacy guarantees of SubspaceRecovery (Theorem 26) and post-processing of DP (Lemma 9).

Now, we prove the accuracy guarantees. Suppose $\Sigma = U \Lambda U^\top$ and $U, \Lambda, \Sigma \in \mathbb{R}^{d \times d}$, where $U^\top U = I$ and $\Lambda$ is diagonal with entries $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_d \geq 0$.

We know that there is a large eigengap – i.e., $\lambda_{k+1} = \gamma^2 \cdot \lambda_k$ for some $k \in [d]$ and $0 < \gamma \ll 1$. Consider the subspace spanned by the eigenvectors corresponding to $\lambda_1, \ldots, \lambda_k$ and let $\Pi_{1:k}$ be the corresponding projection matrix. We then run the subspace algorithm SubspaceRecovery (Singhal and Steinke, 2021) with parameters $\varepsilon, \delta, \beta, k, \hat{\gamma}$ to obtain $\hat{\Pi}_{1:k} \in \mathbb{R}^{d \times d}$ satisfying $\|\hat{\Pi}_{1:k} - \Pi_{1:k}\| \leq \phi \iff \|\hat{\Pi}_{k+1:d} - \Pi_{k+1:d}\| \leq \phi$ with probability at least $1 - O(\beta)$, where, because of our sample complexity $\phi \leq \frac{\hat{\gamma}\overline{\gamma}^2}{100}$.

Now let $y_i = [(1-\eta)\hat{\Pi}_{1:k} + \hat{\Pi}_{k+1:d}]X_i$ for all $i \in [n]$. Here, $0 \leq \eta = 1 - \hat{\gamma}$. Then $y_1, \cdots, y_n \in \mathbb{R}^d$ are $n$ independent draws from $\mathcal{N}(0, \hat{\Sigma})$, where

$$\hat{\Sigma} = (\hat{\gamma}\hat{\Pi}_{1:k} + \hat{\Pi}_{k+1:d})\Sigma(\hat{\gamma}\hat{\Pi}_{1:k} + \hat{\Pi}_{k+1:d}).$$

We set $\xi = \xi_1 = \hat{\Pi}_{1:k} - \Pi_{1:k}$ and $\xi_2 = \hat{\Pi}_{k+1:d} - \Pi_{k+1:d} = -\xi$ where $\|\xi\| \leq \phi$. We have for $\hat{\Sigma}$:

$$\begin{aligned}
\hat{\Sigma} &= (\hat{\gamma}\hat{\Pi}_{1:k} + \hat{\Pi}_{k+1:d})\Sigma(\hat{\gamma}\hat{\Pi}_{1:k} + \hat{\Pi}_{k+1:d}) \\
&= (\hat{\gamma}\xi_1 + \xi_2 + \hat{\gamma}\Pi_{1:k} + \Pi_{k+1:d})\Sigma(\hat{\gamma}\xi_1 + \xi_2 + \hat{\gamma}\Pi_{1:k} + \Pi_{k+1:d}) \\
&= (\hat{\gamma}\xi_1 + \xi_2 + \hat{\gamma}\Pi_{1:k} + \Pi_{k+1:d})\Sigma(\hat{\gamma}\xi_1 + \xi_2 + \hat{\gamma}\Pi_{1:k} + \Pi_{k+1:d}) \\
&= (\hat{\gamma}\xi_1 + \xi_2)\Sigma(\hat{\gamma}\xi_1 + \xi_2) + (\hat{\gamma}\Pi_{1:k} + \Pi_{k+1:d})\Sigma(\hat{\gamma}\xi_1 + \xi_2) \\
&+ (\hat{\gamma}\xi_1 + \xi_2)\Sigma(\hat{\gamma}\Pi_{1:k} + \Pi_{k+1:d}) + (\hat{\gamma}\Pi_{1:k} + \Pi_{k+1:d})\Sigma(\hat{\gamma}\Pi_{1:k} + \Pi_{k+1:d}) \\
&= (\hat{\gamma}\xi_1 + \xi_2)\Sigma(\hat{\gamma}\xi_1 + \xi_2) + \hat{\gamma}\Pi_{1:k}\Sigma(\hat{\gamma}\xi_1 + \xi_2) + \Pi_{k+1:d}\Sigma(\hat{\gamma}\xi_1 + \xi_2) \\
&+ \hat{\gamma}(\hat{\gamma}\xi_1 + \xi_2)\Sigma\Pi_{1:k} + (\hat{\gamma}\xi_1 + \xi_2)\Sigma\Pi_{k+1:d} + \hat{\gamma}^2\Pi_{1:k}\Sigma\Pi_{1:k} + \Pi_{k+1:d}\Sigma\Pi_{k+1:d}.
\end{aligned}$$

Now, we need to find an upper limit for $\lambda_1(\hat{\Sigma})$, and a lower limit for $\lambda_{k+1}(\hat{\Sigma})$.

We start with the upper bound on $\lambda_1(\hat{\Sigma})$.

$$\begin{aligned}
\left\|\hat{\Sigma}\right\| &\leq \|\hat{\gamma}\xi_1 + \xi_2\|^2 \|\Sigma\| + 2\hat{\gamma}\|\Pi_{1:k}\Sigma\| \|\hat{\gamma}\xi_1 + \xi_2\| + 2\|\Pi_{k+1:d}\Sigma\| \|\hat{\gamma}\xi_1 + \xi_2\| \\
&\quad + \hat{\gamma}^2\|\Pi_{1:k}\Sigma\Pi_{1:k}\| + \|\Pi_{k+1:d}\Sigma\Pi_{k+1:d}\| \\
&\leq (1-\hat{\gamma})^2 \tfrac{\hat{\gamma}^2\overline{\gamma}^4}{10000}\lambda_1(\Sigma) + 2(1-\hat{\gamma})\tfrac{\hat{\gamma}^2\overline{\gamma}^2}{100}\lambda_1(\Sigma) + 2(1-\hat{\gamma})\tfrac{\hat{\gamma}\overline{\gamma}^2}{100}\lambda_{k+1}(\Sigma) \\
&\quad + \hat{\gamma}^2\lambda_1(\Sigma) + \lambda_{k+1}(\Sigma) \\
&\leq \frac{\gamma^2\overline{\gamma}^2}{2500}\lambda_k(\Sigma) + \frac{2\gamma^2}{25}\lambda_k(\Sigma) + 2(1-\hat{\gamma})\tfrac{\hat{\gamma}\overline{\gamma}^2}{100}\lambda_{k+1}(\Sigma) + \frac{4\gamma^2}{\overline{\gamma}^2}\lambda_k(\Sigma) + \lambda_{k+1}(\Sigma) \\
&\leq \frac{\overline{\gamma}^2}{2500}\lambda_{k+1}(\Sigma) + \frac{2}{25}\lambda_{k+1}(\Sigma) + 2(1-\hat{\gamma})\tfrac{\hat{\gamma}\overline{\gamma}^2}{100}\lambda_{k+1}(\Sigma) + \frac{4}{\overline{\gamma}^2}\lambda_{k+1}(\Sigma) + \lambda_{k+1}(\Sigma)
\end{aligned}$$

$$\leq \frac{5}{\overline{\gamma}^2}\lambda_{k+1}(\Sigma).$$

Now, we prove a lower bound on $\lambda_{k+1}(\hat{\Sigma})$.

$$\begin{aligned}
\lambda_{k+1}(\hat{\Sigma}) &\geq \lambda_{k+1}\left(\hat{\gamma}^2\Pi_{1:k}\Sigma\Pi_{1:k} + \Pi_{k+1:d}\Sigma\Pi_{k+1:d}\right) \\
&\quad + (1-\hat{\gamma})^2\lambda_d\left(\xi\Sigma\xi\right) - \hat{\gamma}(1-\hat{\gamma})\lambda_d\left(\Pi_{1:k}\Sigma\xi\right) \\
&\quad - (1-\hat{\gamma})\lambda_d\left(\Pi_{k+1:d}\Sigma\xi\right) - \hat{\gamma}(1-\hat{\gamma})\lambda_d\left(\xi\Sigma\Pi_{1:k}\right) \\
&\quad - (1-\hat{\gamma})\lambda_d\left(\xi\Sigma\Pi_{k+1:d}\right) && \text{(Lemma 19)} \\
&\geq \frac{\lambda_{k+1}}{4} - 2\hat{\gamma}(1-\hat{\gamma})\left\|\xi\right\|\left\|\Pi_{1:k}\Sigma\right\| - 2(1-\hat{\gamma})\left\|\xi\right\|\left\|\Sigma\Pi_{k+1:d}\right\| \\
&\geq \frac{\lambda_{k+1}}{4} - \frac{\hat{\gamma}^2\overline{\gamma}^2}{50}\lambda_1(\Sigma) - \frac{\hat{\gamma}\overline{\gamma}^2}{50}\lambda_{k+1}(\Sigma) \\
&\geq \frac{\lambda_{k+1}}{4} - \frac{2}{25}\lambda_{k+1}(\Sigma) - \frac{\hat{\gamma}\overline{\gamma}^2}{50}\lambda_{k+1}(\Sigma) \\
&\geq \frac{\lambda_{k+1}}{8}
\end{aligned}$$

Therefore, $\frac{\lambda_{k+1}(\hat{\Sigma})}{\lambda_1(\hat{\Sigma})} \geq \frac{\overline{\gamma}^2}{40}$. ■

### 3.2. Fine Preconditioning

In this section, we present our second preconditioning constituent (the "fine" preconditioner) that is used in the presence of small cumulative gaps. This component of our preconditioning process is similar to the one that appears in Kamath et al. (2019a). It first uses the naive estimator (i.e., clipping data based on the covariance matrix's spectrum and noising the empirical covariance, Algorithm 7) to get a rough estimate of the covariance. This gives us enough information about the top $k+1$ eigenvectors and eigenvalues to operate (approximately) within the top-$(k+1)$ subspace, allowing us to shrink down the top $k$ eigenvalues by a small multiplicative factor. We initially assume that the gap between the 1-st and the $(k+1)$-th eigenvalues is large, but not too large, essentially the setting that we will be in after running the coarse preconditioner described in Section 3.1. In other words, when the gap between the 1-st and the $(k+1)$-th eigenvalues is loosely bounded, the fine preconditioner tightens that gap. We now present our algorithm and its analysis.

**Theorem 6 (Fine Preconditioner)** *Let $X = (X_1, \ldots, X_n)$ be i.i.d. samples from $\mathcal{N}(0, \Sigma)$, such that for some $1 \leq k < d$, $\frac{\lambda_{k+1}(\Sigma)}{\lambda_1(\Sigma)} \geq \tau^2\overline{\gamma}^2$ for $\overline{\gamma} \leq 1$. Then for all $\varepsilon, \delta > 0$, there exists an $(\varepsilon, \delta)$-DP algorithm, such that if $n \geq O\left(\frac{d^{3/2}\cdot\mathrm{polylog}(d, \frac{1}{\varepsilon}, \frac{1}{\delta}, \frac{1}{\beta})}{\varepsilon\tau^2\overline{\gamma}^2}\right)$, then with probability at least $1 - O(\beta)$, it takes $X$ as input, and outputs a matrix $A$ that satisfies $\frac{\lambda_{k+1}(A\Sigma A)}{\lambda_1(A\Sigma A)} \geq \overline{\gamma}^2$.*

**Proof** We prove the privacy and accuracy guarantees of Algorithm 3. Privacy follows from the guarantees of Lemma 27.

Now, we prove the accuracy. Let $\hat{\Pi}_S$ and $\hat{\Pi}_{\overline{S}}$ be matrices as defined in Algorithm 3. We first show an upper bound on $\|A\Sigma A\|$. For this, by Lemma 22, it is enough to prove an upper bound on

**Algorithm 3:** Differentially Private $\mathrm{FinePreconditioner}_{\varepsilon,\delta,\beta,k,\overline{\gamma},\kappa}(X)$

**Input:** Samples $X_1, \ldots, X_n \in \mathbb{R}^d$. Parameters $\varepsilon, \delta, \beta, k, \overline{\gamma}, \kappa > 0$.
**Output:** Matrix $A \in \mathbb{R}^{d \times d}$.

Set $Z \leftarrow \mathrm{NaiveEstimator}_{\varepsilon,\delta,\beta,\kappa}(X)$.
Let $S \leftarrow \{i : \lambda_i(Z) \geq \frac{\lambda_{k+1}(Z)}{16\overline{\gamma}^2}\}$.
Let $g_i \leftarrow \sqrt{\frac{\lambda_i(Z)}{\lambda_{k+1}(Z)}}$.
Let $v_i$ be the $i$-th eigenvector of $Z$.
Set $\hat{\Pi}_S \leftarrow \sum_{i \in S} \frac{v_i v_i^\top}{4 g_i \overline{\gamma}}$ and $\hat{\Pi}_{\overline{S}} \leftarrow \sum_{i \notin S} v_i v_i^\top$.
Set $A \leftarrow \hat{\Pi}_S + \hat{\Pi}_{\overline{S}}$.
**return** $A$.

$\|A(Z-N)A\|$.

$$\begin{aligned}
\|A(Z-N)A\| &\leq \|AZA\| + \|ANA\| \\
&\leq \|\hat{\Pi}_S Z \hat{\Pi}_S + \hat{\Pi}_{\overline{S}} Z \hat{\Pi}_{\overline{S}}\| + \|N\| \\
&\leq \frac{\lambda_{k+1}(Z)}{16\overline{\gamma}^2} + \frac{\lambda_{k+1}(Z)}{16\overline{\gamma}^2} \\
&= \frac{\lambda_{k+1}(Z)}{8\overline{\gamma}^2}
\end{aligned}$$

In the above, the third inequality comes from Corollary 28 and our sample complexity. This shows that $\|A\Sigma A\| \leq \frac{\lambda_{k+1(Z)}}{4\overline{\gamma}^2}$.

Now, we show a lower bound on $\lambda_{k+1}(A\Sigma A)$. As before, by Lemma 22, it is enough to show a lower bound on $\lambda_{k+1}(A(Z-N)A)$.

$$\begin{aligned}
\lambda_{k+1}(A(Z-N)A) &\geq \lambda_{k+1}(AZA) - \|ANA\| && \text{(Lemma 19)} \\
&\geq \lambda_{k+1}(\hat{\Pi}_S Z \hat{\Pi}_S + \hat{\Pi}_{\overline{S}} Z \hat{\Pi}_{\overline{S}}) - \|N\| \\
&\geq \lambda_{k+1}(Z) - \frac{\lambda_{k+1}(Z)}{2} \\
&\geq \frac{\lambda_{k+1}(Z)}{2}
\end{aligned}$$

In the above, the third inequality again follows from Corollary 28 and our sample complexity. This gives us $\lambda_{k+1}(A\Sigma A) \geq \frac{\lambda_{k+1}(Z)}{4}$.

Therefore, $\frac{\lambda_{k+1}(A\Sigma A)}{\lambda_1(A\Sigma A)} \geq \overline{\gamma}^2$. ■

### 3.3. Putting Everything Together

We are now ready to present our overall preconditioning algorithm (Algorithm 4). The algorithm essentially relies on a *dynamic programming* approach. In particular, the $i-$th iteration always starts

under the assumption that the cumulative gap of the eigenvalues $\lambda_1 \geq \cdots \geq \lambda_i$ is (relatively) small, so the focus is on the gaps involving the eigenvalue $\lambda_{i+1}$, namely the ratios $\frac{\lambda_{i+1}}{\lambda_i}$ and $\frac{\lambda_{i+1}}{\lambda_1}$. Based on how small these ratios are, the algorithm may use either the coarse or the fine preconditioner, or both. Doing so, it ensures that, at the start of the next iteration, the loop's invariant will be preserved. At the end of a run of this algorithm, we get a linear transformation that reduces the multiplicative gap between the 1-st and the $d$-th eigenvalues of $\Sigma$ to $\Omega(1)$. The algorithm and its analysis follow.

**Algorithm 4:** Differentially Private $\mathrm{Preconditioner}_{\varepsilon,\delta,\beta}(X)$

**Input:** Samples $X_1, \ldots, X_n \in \mathbb{R}^d$. Parameters $\varepsilon, \delta, \beta > 0$.
**Output:** Matrix $A \in \mathbb{R}^{d \times d}$.

Set parameter: $\tau^2 \leftarrow \frac{1}{10000}$ $\quad$ $\overline{\gamma}^2 \leftarrow \frac{40}{10000}$

Let $A \leftarrow \mathbb{I}$.
$\hat{\lambda}_1, \ldots, \hat{\lambda}_d \leftarrow \mathrm{EigenvalueEstimator}_{\varepsilon,\delta,\beta}(X)$.
Set $i \leftarrow 1$.
**while** $i < d$ **do**
  **if** $\frac{\hat{\lambda}_{i+1}}{\hat{\lambda}_i} < 4\tau^2$ **then**
    $B \leftarrow \mathrm{CoarsePreconditioner}_{\frac{\varepsilon}{\sqrt{6d\log(1/\delta)}}, \frac{\delta}{d+1}, \frac{\beta}{d}, i, \sqrt{\frac{\hat{\lambda}_{i+1}}{\hat{\lambda}_i}}}(X)$.
    $A \leftarrow BA$.
    $X \leftarrow AX$.
    $Z \leftarrow \mathrm{NaiveEstimator}_{\frac{\varepsilon}{\sqrt{6d\log(1/\delta)}}, \frac{\delta}{d+1}, \frac{\beta}{d}}(X)$.
    **if** $\frac{\lambda_{i+1}(Z)}{\lambda_1(Z)} < 4\overline{\gamma}^2$ **then**
      $C \leftarrow \mathrm{FinePreconditioner}_{\frac{\varepsilon}{\sqrt{6d\log(1/\delta)}}, \frac{\delta}{d+1}, \frac{\beta}{d}, i, \overline{\gamma}, \lambda_1(Z)}(X)$.
      $A \leftarrow CA$.
      $X \leftarrow AX$.
    **end**
  **end**
  **else if** $\frac{\hat{\lambda}_{i+1}}{\hat{\lambda}_1} < 4\overline{\gamma}^2$ **then**
    $D \leftarrow \mathrm{FinePreconditioner}_{\frac{\varepsilon}{\sqrt{6d\log(1/\delta)}}, \frac{\delta}{d+1}, \frac{\beta}{d}, i, \overline{\gamma}, \lambda_1(Z)}(X)$.
    $A \leftarrow DA$.
    $X \leftarrow AX$.
  **end**
  $Z \leftarrow \mathrm{NaiveEstimator}_{\frac{\varepsilon}{\sqrt{6d\log(1/\delta)}}, \frac{\delta}{d+1}, \frac{\beta}{d}}(X)$.
  $\hat{\lambda}_1, \ldots, \hat{\lambda}_d \leftarrow \mathrm{EigenvalueEstimator}_{\frac{\varepsilon}{\sqrt{6d\log(1/\delta)}}, \frac{\delta}{d+1}, \frac{\beta}{d}}(X)$.
  $i \leftarrow i + 1$.
**end**
**return** $A$.

**Theorem 7 (DP Preconditioner)** *Let $\Sigma \in \mathbb{R}^{d\times d}$ be a symmetric, positive-definite matrix. There exists an $(\varepsilon,\delta)$-DP algorithm, such that if $X = (X_1,\ldots,X_n) \sim \mathcal{N}(0,\Sigma)$ and $n \geq O\left(\frac{d^{2.5}\cdot \mathrm{polylog}(d,\frac{1}{\varepsilon},\frac{1}{\delta},\frac{1}{\beta})}{\varepsilon}\right)$, then with probability at least $1-\beta$, the algorithm outputs a matrix $A$ that satisfies $\frac{\lambda_k(A\Sigma A)}{\lambda_1(A\Sigma A)} \geq \Omega(1)$.*

**Proof** We prove the theorem by proving the privacy and accuracy of Algorithm 4. Privacy follows from Theorems 5, 6, and 23, Lemma 27, and composition of DP (Lemma 10).

For the accuracy argument, it is enough to show that at the beginning of each iteration $1 \leq i \leq k$, $\frac{\lambda_i(A\Sigma A)}{\lambda_1(A\Sigma A)} \geq O(\overline{\gamma}^2)$. We prove this via induction on $i$.

For the basis step, it is trivial because $A\Sigma A = \Sigma$. Therefore, the ratio equals 1.

Now, we move on to the inductive step. Suppose for $i > 1$, the claim holds for all $j < i$. Let the matrix $A$ be equal to $A_{i-1}$ at the beginning of iteration $i-1$. This implies that for iteration $i-1$,

$$\frac{\lambda_{i-1}(A_{i-1}\Sigma A_{i-1})}{\lambda_1(A_{i-1}\Sigma A_{i-1})} \geq \overline{\gamma}^2. \tag{1}$$

According to the **If**-block, if the privately estimated eigenvalue ratio is less than $4\tau^2$, then it must be the case that with high probability (Theorem 23), $\frac{\lambda_{i-1}(A_{i-1}\Sigma A_{i-1})}{\lambda_i(A_{i-1}\Sigma A_{i-1})} < 16\tau^2$. Then because of (1), Theorem 5, and Corollary 28, it must be the case that with probability $1-O(\beta/d)$, at the beginning of the nested **If**-block, $\frac{\lambda_i(BA_{i-1}\Sigma(BA_{i-1})^\top)}{\lambda_1(BA_{i-1}\Sigma(BA_{i-1})^\top)} \geq \frac{\overline{\gamma}^2}{40}$. Now, if $\frac{\lambda_i(Z)}{\lambda_1(Z)} < 4\overline{\gamma}^2$, then by Corollary 28, $\frac{\lambda_i(BA_{i-1}\Sigma(BA_{i-1})^\top)}{\lambda_1(BA_{i-1}\Sigma(BA_{i-1})^\top)} < 16\overline{\gamma}^2$. By the guarantees of Theorem 6, with probability at least $1-O(\beta/d)$, at the end of the nested **If**-block (hence, at the end of the loop and the starting of the $i$-th iteration), $\frac{\lambda_i(CBA_{i-1}\Sigma(CBA_{i-1})^\top)}{\lambda_1(CBA_{i-1}\Sigma(CBA_{i-1})^\top)} \geq \overline{\gamma}^2$. Suppose, the algorithm skips the first **If**-block. Then with high probability, it must be the case that $\frac{\lambda_{i-1}(A_{i-1}\Sigma A_{i-1})}{\lambda_i(A_{i-1}\Sigma A_{i-1})} \geq \tau^2$. If it enters the **ElIf**-block, then it mean that with high probability, $\frac{\lambda_i(A_{i-1}\Sigma A_{i-1})}{\lambda_1(A_{i-1}\Sigma A_{i-1})} < 16\overline{\gamma}^2$. Then again, by the guarantees of Theorem 6, with probability at least $1-O(\beta/d)$, at the end of the iteration, $\frac{\lambda_i(DA_{i-1}\Sigma(DA_{i-1})^\top)}{\lambda_1(DA_{i-1}\Sigma(DA_{i-1})^\top)} \geq \overline{\gamma}^2$.
This proves the inductive step. If neither of the **If** or **ElIf**-blocks are entered, it would mean that the ratio is already at least $\overline{\gamma}^2$. Applying the union bound over all $i$, we get the required result. ■

## Acknowledgments

GK was supported by an NSERC Discovery Grant, and a University of Waterloo startup grant. AM was supported by an NSERC Discovery Grant and a David R. Cheriton Graduate Scholarship. Part of VS' research was performed at the Khoury College of Computer Sciences, Northeastern University – supported by NSF grants CCF-1750640, CNS-1816028, and CNS-1916020, and an NSERC Discovery Grant. JU was affiliated with the Institute for Experiential AI and the Cybersecurity & Privacy Institute – supported by NSF grants CCF-1750640, CNS-1816028, and CNS-1916020.

## References


Jayadev Acharya, Ziteng Sun, and Huanyu Zhang. Differentially private assouad, fano, and le cam. In *Proceedings of the 32nd International Conference on Algorithmic Learning Theory*, ALT '21, pages 48–78. JMLR, Inc., 2021.

Ishaq Aden-Ali, Hassan Ashtiani, and Gautam Kamath. On the sample complexity of privately learning unbounded high-dimensional gaussians. In *Proceedings of the 32nd International Conference on Algorithmic Learning Theory*, ALT '21, pages 185–216. JMLR, Inc., 2021a.

Ishaq Aden-Ali, Hassan Ashtiani, and Christopher Liaw. Privately learning mixtures of axis-aligned gaussians. In *Advances in Neural Information Processing Systems 34*, NeurIPS '21. Curran Associates, Inc., 2021b.

Hassan Ashtiani and Christopher Liaw. Private and polynomial time algorithms for learning Gaussians and beyond. *arXiv preprint arXiv:2111.11320*, 2021.

Marco Avella-Medina and Victor-Emmanuel Brunel. Differentially private sub-Gaussian location estimators. *arXiv preprint arXiv:1906.11923*, 2019.

Brendan Avent, Yatharth Dubey, and Aleksandra Korolova. The power of the hybrid model for mean estimation. *Proceedings on Privacy Enhancing Technologies*, 2020(4):48–68, 2019.

Rina Foygel Barber and John C Duchi. Privacy and statistical risk: Formalisms and minimax bounds. *arXiv preprint arXiv:1412.4451*, 2014.

Sourav Biswas, Yihe Dong, Gautam Kamath, and Jonathan Ullman. Coinpress: Practical private mean and covariance estimation. In *Advances in Neural Information Processing Systems 33*, NeurIPS '20, pages 14475–14485. Curran Associates, Inc., 2020.

Gavin Brown, Marco Gaboardi, Adam Smith, Jonathan Ullman, and Lydia Zakynthinou. Covariance-aware private mean estimation without private covariance estimation. In *Advances in Neural Information Processing Systems 34*, NeurIPS '21. Curran Associates, Inc., 2021.

Mark Bun and Thomas Steinke. Concentrated differential privacy: Simplifications, extensions, and lower bounds. In *Proceedings of the 14th Conference on Theory of Cryptography*, TCC '16-B, pages 635–658, Berlin, Heidelberg, 2016. Springer.

Mark Bun and Thomas Steinke. Average-case averages: Private algorithms for smooth sensitivity and mean estimation. In *Advances in Neural Information Processing Systems 32*, NeurIPS '19, pages 181–191. Curran Associates, Inc., 2019.

Mark Bun, Jonathan Ullman, and Salil Vadhan. Fingerprinting codes and the price of approximate differential privacy. In *Proceedings of the 46th Annual ACM Symposium on the Theory of Computing*, STOC '14, pages 1–10, New York, NY, USA, 2014. ACM.

Mark Bun, Kobbi Nissim, and Uri Stemmer. Simultaneous private learning of multiple concepts. In *Proceedings of the 7th Conference on Innovations in Theoretical Computer Science*, ITCS '16, pages 369–380, New York, NY, USA, 2016. ACM.

Mark Bun, Thomas Steinke, and Jonathan Ullman. Make up your mind: The price of online queries in differential privacy. In *Proceedings of the 28th Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '17, pages 1306–1325, Philadelphia, PA, USA, 2017. SIAM.

Mark Bun, Gautam Kamath, Thomas Steinke, and Zhiwei Steven Wu. Private hypothesis selection. In *Advances in Neural Information Processing Systems 32*, NeurIPS '19, pages 156–167. Curran Associates, Inc., 2019.

T. Tony Cai, Yichen Wang, and Linjun Zhang. The cost of privacy: Optimal rates of convergence for parameter estimation with differential privacy. *arXiv preprint arXiv:1902.04495*, 2019.

Rachel Cummings and David Durfee. Individual sensitivity preprocessing for data privacy. In *Proceedings of the 31st Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '20, Philadelphia, PA, USA, 2020. SIAM.

Ilias Diakonikolas, Moritz Hardt, and Ludwig Schmidt. Differentially private learning of structured discrete distributions. In *Advances in Neural Information Processing Systems 28*, NIPS '15, pages 2566–2574. Curran Associates, Inc., 2015.

Ilias Diakonikolas, Gautam Kamath, Daniel M. Kane, Jerry Li, Ankur Moitra, and Alistair Stewart. Robust estimators in high dimensions without the computational intractability. In *Proceedings of the 57th Annual IEEE Symposium on Foundations of Computer Science*, FOCS '16, pages 655–664, Washington, DC, USA, 2016. IEEE Computer Society.

Wenxin Du, Canyon Foot, Monica Moniot, Andrew Bray, and Adam Groce. Differentially private confidence intervals. *arXiv preprint arXiv:2001.02285*, 2020.

Cynthia Dwork and Jing Lei. Differential privacy and robust statistics. In *Proceedings of the 41st Annual ACM Symposium on the Theory of Computing*, STOC '09, pages 371–380, New York, NY, USA, 2009. ACM.

Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Proceedings of the 3rd Conference on Theory of Cryptography*, TCC '06, pages 265–284, Berlin, Heidelberg, 2006. Springer.

Cynthia Dwork, Guy N. Rothblum, and Salil Vadhan. Boosting and differential privacy. In *Proceedings of the 51st Annual IEEE Symposium on Foundations of Computer Science*, FOCS '10, pages 51–60, Washington, DC, USA, 2010. IEEE Computer Society.

Cynthia Dwork, Kunal Talwar, Abhradeep Thakurta, and Li Zhang. Analyze Gauss: Optimal bounds for privacy-preserving principal component analysis. In *Proceedings of the 46th Annual ACM Symposium on the Theory of Computing*, STOC '14, pages 11–20, New York, NY, USA, 2014. ACM.

Cynthia Dwork, Adam Smith, Thomas Steinke, Jonathan Ullman, and Salil Vadhan. Robust traceability from trace amounts. In *Proceedings of the 56th Annual IEEE Symposium on Foundations of Computer Science*, FOCS '15, pages 650–669, Washington, DC, USA, 2015. IEEE Computer Society.

Samuel B Hopkins, Gautam Kamath, and Mahbod Majid. Efficient mean estimation with pure differential privacy via a sum-of-squares exponential mechanism. 2022.

Ziyue Huang, Yuting Liang, and Ke Yi. Instance-optimal mean estimation under differential privacy. In *Advances in Neural Information Processing Systems 34*, NeurIPS '21. Curran Associates, Inc., 2021.

Gautam Kamath and Jonathan Ullman. A primer on private statistics. *arXiv preprint arXiv:2005.00010*, 2020.

Gautam Kamath, Jerry Li, Vikrant Singhal, and Jonathan Ullman. Privately learning high-dimensional distributions. In *Proceedings of the 32nd Annual Conference on Learning Theory*, COLT '19, pages 1853–1902, 2019a.

Gautam Kamath, Or Sheffet, Vikrant Singhal, and Jonathan Ullman. Differentially private algorithms for learning mixtures of separated Gaussians. In *Advances in Neural Information Processing Systems 32*, NeurIPS '19, pages 168–180. Curran Associates, Inc., 2019b.

Gautam Kamath, Vikrant Singhal, and Jonathan Ullman. Private mean estimation of heavy-tailed distributions. In *Proceedings of the 33rd Annual Conference on Learning Theory*, COLT '20, pages 2204–2235, 2020.

Gautam Kamath, Xingtu Liu, and Huanyu Zhang. Improved rates for differentially private stochastic convex optimization with heavy-tailed data. *arXiv preprint arXiv:2106.01336*, 2021.

Vishesh Karwa and Salil Vadhan. Finite sample differentially private confidence intervals. In *Proceedings of the 9th Conference on Innovations in Theoretical Computer Science*, ITCS '18, pages 44:1–44:9, Dagstuhl, Germany, 2018. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

Aleksandra Korolova, Krishnaram Kenthapadi, Nina Mishra, and Alexandros Ntoulas. Releasing search queries and clicks privately. In *Proceedings of the 18th International World Wide Web Conference*, WWW '09, pages 171–180, New York, NY, USA, 2009. ACM.

Pravesh K Kothari, Pasin Manurangsi, and Ameya Velingker. Private robust estimation by stabilizing convex relaxations. *arXiv preprint arXiv:2112.03548*, 2021.

Beatrice Laurent and Pascal Massart. Adaptive estimation of a quadratic functional by model selection. *The Annals of Statistics*, 28(5):1302–1338, 2000.

Daniel Levy, Ziteng Sun, Kareem Amin, Satyen Kale, Alex Kulesza, Mehryar Mohri, and Ananda Theertha Suresh. Learning with user-level privacy. In *Advances in Neural Information Processing Systems 34*, NeurIPS '21. Curran Associates, Inc., 2021.

Xiyang Liu, Weihao Kong, Sham Kakade, and Sewoong Oh. Robust and differentially private mean estimation. In *Advances in Neural Information Processing Systems 34*, NeurIPS '21. Curran Associates, Inc., 2021a.

Xiyang Liu, Weihao Kong, and Sewoong Oh. Differential privacy and robust statistics in high dimensions. *arXiv preprint arXiv:2111.06578*, 2021b.

Yuhan Liu, Ananda Theertha Suresh, Felix Yu, Sanjiv Kumar, and Michael Riley. Learning discrete distributions: User vs item-level privacy. In *Advances in Neural Information Processing Systems 33*, NeurIPS '20. Curran Associates, Inc., 2020.

Kobbi Nissim, Uri Stemmer, and Salil Vadhan. Locating a small cluster privately. In *Proceedings of the 35th ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, PODS '16, pages 413–427, New York, NY, USA, 2016. ACM.

Vikrant Singhal and Thomas Steinke. Privately learning subspaces. In *Advances in Neural Information Processing Systems 34*, NeurIPS '21. Curran Associates, Inc., 2021.

Adam Smith. Privacy-preserving statistical estimation with optimal convergence rates. In *Proceedings of the 43rd Annual ACM Symposium on the Theory of Computing*, STOC '11, pages 813–822, New York, NY, USA, 2011. ACM.

Thomas Steinke and Jonathan Ullman. Between pure and approximate differential privacy. *The Journal of Privacy and Confidentiality*, 7(2):3–22, 2017a.

Thomas Steinke and Jonathan Ullman. Tight lower bounds for differentially private selection. In *Proceedings of the 58th Annual IEEE Symposium on Foundations of Computer Science*, FOCS '17, pages 552–563, Washington, DC, USA, 2017b. IEEE Computer Society.

Terence Tao. *Topics in random matrix theory*, volume 132 of *Graduate Studies in Mathematics*. American Mathematical Society, Providence, RI, 2012.

Eliad Tsfadia, Edith Cohen, Haim Kaplan, Yishay Mansour, and Uri Stemmer. Friendlycore: Practical differentially private aggregation. *arXiv preprint arXiv:2110.10132*, 2021.

Christos Tzamos, Emmanouil-Vasileios Vlatakis-Gkaragkounis, and Ilias Zadik. Optimal private median estimation under minimal distributional assumptions. In *Advances in Neural Information Processing Systems 33*, NeurIPS '20, pages 3301–3311. Curran Associates, Inc., 2020.

Salil Vadhan. The complexity of differential privacy. In Yehuda Lindell, editor, *Tutorials on the Foundations of Cryptography: Dedicated to Oded Goldreich*, chapter 7, pages 347–450. Springer International Publishing AG, Cham, Switzerland, 2017.

Duy Vu and Aleksandra Slavković. Differential privacy for clinical trial data: Preliminary evaluations. In *2009 IEEE International Conference on Data Mining Workshops*, ICDMW '09, pages 138–143. IEEE, 2009.

Di Wang, Hanshen Xiao, Srinivas Devadas, and Jinhui Xu. On differentially private stochastic convex optimization with heavy-tailed data. In *Proceedings of the 37th International Conference on Machine Learning*, ICML '20, pages 10081–10091. JMLR, Inc., 2020.

Larry Wasserman and Shuheng Zhou. A statistical framework for differential privacy. *Journal of the American Statistical Association*, 105(489):375–389, 2010.

Huanyu Zhang, Gautam Kamath, Janardhan Kulkarni, and Zhiwei Steven Wu. Privately learning Markov random fields. In *Proceedings of the 37th International Conference on Machine Learning*, ICML '20, pages 11129–11140. JMLR, Inc., 2020.

## Appendix A. Preliminaries

### A.1. Differential Privacy Preliminaries

A *dataset* $X = (X_1, \ldots, X_n) \in \mathcal{X}^n$ is a collection of elements from some *universe*. We say that two datasets $X, X' \in \mathcal{X}^n$ are *neighboring* if they differ on at most a single entry, and denote this by $X \sim X'$.

**Definition 8 (Differential Privacy (DP) (Dwork et al., 2006))** *A randomized algorithm $M : \mathcal{X}^n \to \mathcal{Y}$ satisfies $(\varepsilon, \delta)$-*differential privacy *($(\varepsilon, \delta)$-DP) if for every pair of neighboring datasets $X, X' \in \mathcal{X}^n$,*

$$\forall Y \subseteq \mathcal{Y} \quad \mathbb{P}\left(M(X) \in Y\right) \leq e^{\varepsilon}\mathbb{P}\left(M(X') \in Y\right) + \delta.$$

This definition is closed under post-processing

**Lemma 9 (Post-Processing (Dwork et al., 2006))** *If $M : \mathcal{X}^n \to \mathcal{Y}$ is $(\varepsilon, \delta)$-DP and $P : \mathcal{Y} \to \mathcal{Z}$ is any randomized function, then the algorithm $P \circ M$ is $(\varepsilon, \delta)$-DP.*

A crucial property of all the variants of differential privacy is that they can be composed adaptively. By adaptive composition, we mean a sequence of algorithms $M_1(X), \ldots, M_T(X)$ where the algorithm $M_t(X)$ may also depend on the outcomes of the algorithms $M_1(X), \ldots, M_{t-1}(X)$.

**Lemma 10 (Composition of DP (Dwork et al., 2006, 2010; Bun and Steinke, 2016))** *If $M$ is an adaptive composition of differentially private algorithms $M_1, \ldots, M_T$, then the following all hold:*

1. *If $M_1, \ldots, M_T$ are $(\varepsilon_1, \delta_1), \ldots, (\varepsilon_T, \delta_T)$-DP then $M$ is $(\varepsilon, \delta)$-DP for $\varepsilon = \sum_t \varepsilon_t$ and $\delta = \sum_t \delta_t$*
2. *If $M_1, \ldots, M_T$ are $(\varepsilon_0, \delta_1), \ldots, (\varepsilon_0, \delta_T)$-DP for some $\varepsilon_0 \leq 1$, then for every $\delta_0 > 0$, $M$ is $(\varepsilon, \delta)$-DP for*

$$\varepsilon = \varepsilon_0 \cdot \sqrt{6T \log(1/\delta_0)} \quad \text{and} \quad \delta = \delta_0 + \sum_t \delta_t$$

Note that the first property says that $(\varepsilon, \delta)$-DP composes linearly—the parameters simply add up. The second property says that $(\varepsilon, \delta)$-DP actually composes sublinearly—the parameter $\varepsilon$ grows roughly with the square root of the number of steps in the composition, provided we allow a small increase in $\delta$.

#### A.1.1. Useful Differentially Private Mechanisms

Our algorithms will extensively use the well known and standard Gaussian mechanism to ensure differential privacy.

**Definition 11 ($\ell_2$-Sensitivity)** *Let $f : \mathcal{X}^n \to \mathbb{R}^d$ be a function, its $\ell_2$-*sensitivity *is*

$$\Delta_f = \max_{X \sim X' \in \mathcal{X}^n} \|f(X) - f(X')\|_2$$

**Lemma 12 (Gaussian Mechanism)** *Let $f : \mathcal{X}^n \to \mathbb{R}^d$ be a function with $\ell_2$-sensitivity $\Delta_f$. Then the Gaussian mechanism*

$$M(X) = f(X) + \mathcal{N}\left(0, \frac{2\Delta_f^2 \ln(2/\delta)}{\varepsilon^2} \cdot \mathbb{I}_{d\times d}\right)$$

*satisfies $(\varepsilon, \delta)$-DP.*

Next, we describe a tool to privately estimate histograms.

**Lemma 13 (Stability-based Histograms (Korolova et al., 2009; Bun et al., 2016; Vadhan, 2017))** *Let $(X_1, \ldots, X_n)$ be samples in some data universe $U$, and let $\Omega = \{h_u\}_{u\subset U}$ be a collection of disjoint histogram buckets over $U$. Then we have an $(\varepsilon, \delta)$-DP histogram algorithm with the following guarantees:*

- *With probability at least $1 - \beta$, the $\ell_\infty$ error is $O\left(\frac{\log(1/\delta\beta)}{\varepsilon}\right)$.*
- *The algorithm runs in time* $\text{poly}\left(n, \log\left(\frac{1}{\varepsilon\beta}\right)\right)$.

Finally, we provide a tool to find an approximately smallest ball that contains all the points in the dataset with high probability.

**Theorem 14 (**GoodCenter **from Nissim et al. (2016))** *Let $X = (X_1, \ldots, X_n) \in \mathbb{R}^D$ be the dataset such that*

$$n \geq O\left(\frac{\sqrt{d} \cdot \text{polylog}(D, \frac{1}{\varepsilon}, \frac{1}{\delta}, \frac{1}{\beta})}{\varepsilon}\right).$$

*Suppose the smallest ball in $\mathbb{R}^D$ that contains all the points in $X$ has radius $R_{\text{opt}}$. Then for all $\varepsilon, \delta, \beta > 0$, there exists an $(\varepsilon, \delta)$-DP algorithm (*GoodCenter*) that takes $X, R_{\text{opt}}$ as input, and outputs a point $c \in \mathbb{R}^D$, such that $B_{CR_{\text{opt}}\sqrt{\log n}}(c)$ (for a universal constant $C$) contains at least $\frac{n}{2}$ points from $X$ with probability at least $1 - \beta$.*

### A.2. Distribution Estimation Preliminaries

In this work, our goal is to estimate some underlying distribution in total variation distance. We will achieve this by estimating the parameters of the distribution, and we argue that a distribution from the class with said parameters will be accurate in total variation distance. For a vector $x$, define $\|x\|_\Sigma = \|\Sigma^{-1/2}x\|_2$. Similarly, for a matrix $X$, define $\|X\|_\Sigma = \|\Sigma^{-1/2}X\Sigma^{-1/2}\|_F$. With these two norms in place, we have the following lemma, which is a combination of Corollaries 2.13 and 2.14 of Diakonikolas et al. (2016).

**Lemma 15** *Let $\alpha \geq 0$ be smaller than some absolute constant. Suppose that $\|\mu - \hat{\mu}\|_\Sigma \leq \alpha$, and $\|\Sigma - \hat{\Sigma}\|_\Sigma \leq \alpha$, where $\mathcal{N}(\mu, \Sigma)$ is a Gaussian distribution in $\mathbb{R}^d$, $\hat{\mu} \in \mathbb{R}^d$, and $\Sigma \in \mathbb{R}^{d\times d}$ is a PSD matrix. Then $\mathrm{d_{TV}}(\mathcal{N}(\mu, \Sigma), \mathcal{N}(\hat{\mu}, \hat{\Sigma})) \leq O(\alpha)$.*

### A.2.1. USEFUL INEQUALITIES

We will need several facts about Gaussians and Gaussian matrices. Throughout this section, let $\mathrm{GUE}(\sigma^2)$ denote the distribution over $d \times d$ symmetric matrices $M$ where for all $i \leq j$, we have $M_{ij} \sim \mathcal{N}(0, \sigma^2)$ i.i.d.. From basic random matrix theory, we have the following guarantee.

**Theorem 16 (see e.g. Tao (2012) Corollary 2.3.6)** *For $d$ sufficiently large, there exist absolute constants $C, c > 0$ such that*

$$\mathop{\mathbb{P}}_{M \sim \mathrm{GUE}(\sigma^2)} \left( \|M\|_2 > A\sigma\sqrt{d} \right) \leq C \exp(-cAd)$$

*for all $A \geq C$.*

We also require the following, well known tail bound on quadratic forms on Gaussians.

**Theorem 17 (Hanson-Wright Inequality (see e.g. Laurent and Massart (2000)))** *Let $X \sim \mathcal{N}(0, \mathbb{I})$ and let $A$ be a $d \times d$ matrix. Then, for all $t > 0$, the following two bounds hold:*

$$\mathbb{P}\left( X^\top A X - \mathrm{tr}(A) \geq 2\|A\|_F \sqrt{t} + 2\|A\|_2 t \right) \leq \exp(-t) \tag{2}$$

$$\mathbb{P}\left( X^\top A X - \mathrm{tr}(A) \leq -2\|A\|_F \sqrt{t} \right) \leq \exp(-t) \tag{3}$$

As a special case of the above inequality, we also have the following.

**Fact 18 (Laurent and Massart (2000))** *Fix $\beta > 0$, and let $X_1, \ldots, X_m \sim \mathcal{N}(0, \sigma^2)$ be independent. Then*

$$\mathbb{P}\left( \left| \frac{1}{m} \sum_{i=1}^{m} X_i^2 - \sigma^2 \right| > 4\sigma^2 \left( \sqrt{\frac{\log(1/\beta)}{m}} + \frac{2\log(1/\beta)}{m} \right) \right) \leq \beta$$

Now, we state an inequality bounding the eigenvalues of sum of two matrices.

**Lemma 19 (Weyl's Inequality)** *Let $M, N, R$ be $d \times d$ Hermitian matrices, such that $M = N + R$. Then for each $1 \leq i \leq d$,*

$$\lambda_i(N) + \lambda_d(R) \leq \lambda_i(M) \leq \lambda_i(N) + \lambda_1(R).$$

In order to prove accuracy, we will use the following standard tail bounds for Gaussian random variables.

**Lemma 20** *If $Z \sim \mathcal{N}(0, \sigma^2)$ then for every $t > 0$, $\mathbb{P}(|Z| > t\sigma) \leq 2e^{-t^2/2}$.*

### A.2.2. DETERMINISTIC REGULARITY CONDITIONS FOR GAUSSIANS

We will rely on certain regularity properties of i.i.d. samples from a Gaussian. These are standard concentration inequalities, and a reference for these facts is Section 4 of Diakonikolas et al. (2016).

**Fact 21** *Let $X_1, \ldots, X_n \sim \mathcal{N}(0, \Sigma)$ i.i.d. for $\kappa_1 \mathbb{I} \preceq \Sigma \preceq \kappa_2 \mathbb{I}$. Let $Y_i = \Sigma^{-1/2} X_i$ and let*

$$\widehat{\Sigma}_Y = \frac{1}{n} \sum_{i=1}^{n} Y_i Y_i^\top$$

*Then for every $\beta > 0$, the following conditions hold except with probability $1 - O(\beta)$.*

$$\forall i \in [n] \quad \|Y_i\|_2^2 \le O\left(d \log(n/\beta)\right) \tag{4}$$

$$\left(1 - O\left(\sqrt{\frac{d + \log(1/\beta)}{n}}\right)\right) \cdot \mathbb{I} \preceq \widehat{\Sigma}_Y \preceq \left(1 + O\left(\sqrt{\frac{d + \log(1/\beta)}{n}}\right)\right) \cdot \mathbb{I} \tag{5}$$

$$\left\|\mathbb{I} - \widehat{\Sigma}_Y\right\|_F \le O\left(\sqrt{\frac{d^2 + \log(1/\beta)}{n}}\right) \tag{6}$$

We now note some simple consequences of these conditions. These inequalities follow from simple linear algebra and we omit their proof for conciseness.

**Lemma 22** *Let $Y_1, \dots, Y_n$ satisfy (4)–(6). Fix $M \succ 0$, and for all $i = 1, \dots, n$, let $Z_i = M^{1/2} Y_i$, and let $\widehat{\Sigma}_Z = \frac{1}{n} \sum_{i=1}^n Z_i Z_i^\top$. Let $\kappa'$ be the top eigenvalue of $M$. Then*

$$\forall i \in [n] \quad \|Z_i\|_2^2 \le O\left(\kappa' d \log(n/\beta)\right)$$

$$\left(1 - O\left(\sqrt{\frac{d + \log(1/\beta)}{n}}\right)\right) \cdot M \preceq \widehat{\Sigma}_Z \preceq \left(1 + O\left(\sqrt{\frac{d + \log(1/\beta)}{n}}\right)\right) \cdot M$$

$$\left\|M - \widehat{\Sigma}_Z\right\|_M \le O\left(\sqrt{\frac{d^2 + \log(1/\beta)}{n}}\right)$$

## Appendix B. Eigenvalue Estimation

In this section, we present an algorithm that estimates the eigenvalues of a covariance matrix of a Gaussian distribution up to a constant factor, under the constraint of approximate differential privacy. This algorithm's function is important for the following sections, since it helps us overcome the issue that we have no prior bounds on the eigenvalues, as well as identify gaps between them. The algorithm performs a subsample-and-aggregate process. The samples are split into $t$ subsets and for each of them, the eigenvalues of the empirical covariance are computed. Denoting the $i$-th eigenvalue (in decreasing order of magnitude) of the $j$-th subsample by $\lambda_i^j$, for each $i$, we construct stability-based histograms and output an estimate of $\lambda_i$ based on the bucket where $\lambda_i^j$ tend to concentrate most.

**Theorem 23** *For every $\varepsilon, \delta, \beta > 0$, there exists an $(\varepsilon, \delta)$-DP algorithm, that takes*

$$n = O\left(\frac{d^{3/2} \cdot \mathrm{polylog}(d, 1/\delta, 1/\varepsilon, 1/\beta)}{\varepsilon}\right)$$

*samples from $\mathcal{N}(0, \Sigma)$, for an arbitrary symmetric, positive-semidefinite $\Sigma \in \mathbb{R}^{d \times d}$, and outputs $\hat{\lambda}_1 \ge \cdots \ge \hat{\lambda}_d$, such that with probability at least $1 - O(\beta)$, $\hat{\lambda}_i \in \left[\frac{\lambda_i(\Sigma)}{\sqrt{2}}, \sqrt{2}\lambda_i(\Sigma)\right]$ for all $i$.*

**Proof** We show this by proving privacy and accuracy guarantees of Algorithm 5.

Fix an $i \in [d]$. Then by changing one sample in $X$, only one subsample of $X$ (say, $X^j$) gets changed, hence, only one $\lambda_i^j$ gets affected. This can change at most two histogram buckets, leading to sensitivity 2. Therefore, by the privacy of private histograms Lemma 13, we have

**Algorithm 5:** Differentially Private $\text{EigenvalueEstimator}_{\varepsilon,\delta,\beta}(X)$

**Input:** Samples $X_1, \ldots, X_n \in \mathbb{R}^d$. Parameters $\varepsilon, \delta, \beta > 0$.
**Output:** Noisy eigenvalues of $X$: $(\hat{\lambda}_1, \ldots, \hat{\lambda}_d) \in \mathbb{R}^d$.

Set parameters: $t \leftarrow \frac{C_1 \log(d/\delta\beta)}{\varepsilon}$ $\quad m \leftarrow \lfloor n/t \rfloor$

Split $X$ into $t$ datasets of size $m$: $X^1, \ldots, X^t$.

`// Estimate the eigenvalues via DP Histograms.`

**for** $i \leftarrow 1, \ldots, d$ **do**
  **for** $j \leftarrow 1, \ldots, t$ **do**
    Let $\lambda_i^j$ be the $i$-th eigenvalue of $\frac{1}{m} \cdot X^{j\top} X^j$.
  **end**
  Divide $[0, \infty)$ into
  $\Omega \leftarrow \{\ldots, [1/\sqrt{2}, 1/2^{1/4})[1/2^{1/4}, 1)[1, 2^{1/4}), [2^{1/4}, \sqrt{2}), \ldots\} \cup \{[0,0]\}$.
  Run $\left(\frac{\varepsilon}{\sqrt{6d\log(1/\delta)}}, \frac{\delta}{d+1}\right)$-DP histogram on all $\lambda_i^j$ over $\Omega$.
  **if** *no bucket is returned* **then**
    **return** $\perp$.
  **end**
  Let $[l, r]$ be a non-empty bucket returned.
  Set $\bar{\lambda}_i \leftarrow l$.
**end**

Sort $(\bar{\lambda}_1, \ldots, \bar{\lambda}_d)$ to get $\hat{\lambda}_1, \ldots, \hat{\lambda}_d$.

**return** $(\hat{\lambda}_1, \ldots, \hat{\lambda}_d)$

$\left(O\left(\frac{\varepsilon}{\sqrt{d\log(1/\delta)}}\right), O\left(\frac{\delta}{d}\right)\right)$-DP for this fixed $i$. Applying Lemma 10 gives us the final privacy guarantee.

Now, we move on to the accuracy guarantees. It is sufficient to show that with probability at least $1 - O(\beta/d)$, for each $1 \le i \le d$, $\bar{\lambda}_i \in \left[\frac{\lambda_i(\Sigma)}{\sqrt{2}}, \sqrt{2}\lambda_i(\Sigma)\right]$. Fix an $i$. Now, by Lemma 22, with probability at least $1 - O(\beta/d)$, the non-private estimates of $\lambda_i(\Sigma)$ must be within a factor of $2^{1/8}$ of $\lambda_i(\Sigma)$ due to our sample complexity. Therefore, at most two consecutive buckets would be filled with $\lambda_i^j$'s. Due to our sample complexity and Lemma 13, those buckets are released with probability at least $1 - O(\beta/d)$. Since they are built at a multiplicative width of $2^{1/4}$, they approximate the non-private estimate to within a factor of $2^{1/4}$. Therefore, the total multiplicative error is at most a factor of 2. Taking the union bound over all $i$, we get the required result. ■

## Appendix C. Subspace Recovery

We improve the guarantees of the subspace algorithm from Singhal and Steinke (2021) for our problem, where we are willing to pay $\text{poly}(d)$ in the sample complexity. In our version, the algorithm's aggregation step uses the ball-finding algorithm from Nissim et al. (2016), followed by noisy mean

estimation, instead of using high-dimensional stability-based histograms as in Singhal and Steinke (2021). For completeness, we restate the entire algorithm, but just point out the differences in the proof of the final accuracy lemma from Singhal and Steinke (2021).

**Algorithm 6:** DP Subspace Estimator $\mathrm{SubspaceRecovery}_{\varepsilon,\delta,\alpha,\gamma,k}(X)$

**Input:** Samples $X_1, \ldots, X_n \in \mathbb{R}^d$. Parameters $\varepsilon, \delta, \alpha, \gamma, k > 0$.
**Output:** Projection matrix $\widehat{\Pi} \in \mathbb{R}^{d \times d}$ of rank $k$.

Set parameters: $t \leftarrow \frac{C_0\sqrt{dk}\cdot\mathrm{polylog}(d,k,\frac{1}{\varepsilon},\frac{1}{\delta})}{\varepsilon}$ $\quad m \leftarrow \lfloor n/t \rfloor$ $\quad q \leftarrow C_1 k$

$r \leftarrow \frac{C_2\gamma\sqrt{d}(\sqrt{k}+\sqrt{\ln(kt)})}{\sqrt{m}}$

Sample reference points $p_1, \ldots, p_q$ from $\mathcal{N}(\mathbf{0}, \mathbb{I})$ independently.

`// Subsample from` $X$`, and form projection matrices.`
**for** $j \in 1, \ldots, t$ **do**
  Let $X^j = (X_{(j-1)m+1}, \ldots, X_{jm}) \in \mathbb{R}^{d \times m}$.
  Let $\Pi_j \in \mathbb{R}^{d \times d}$ be the projection matrix onto the subspace spanned by the eigenvectors of $X^j (X^j)^\top \in \mathbb{R}^{d \times d}$ corresponding to the largest $k$ eigenvalues.
  **for** $i \in 1, \ldots, q$ **do**
    $p_i^j \leftarrow \Pi_j p_i$
  **end**
**end**

`// Aggregate using a ball-finding algorithm.`
**for** $i \in [q]$ **do**
  Let $P_i \in \mathbb{R}^{d \times t}$ be the dataset, where column $j$ is $p_i^j$.
  Set $c_i \leftarrow \mathrm{GoodCenter}_{\frac{\varepsilon}{\sqrt{q\ln(1/\delta)}}, \frac{\delta}{q}, r}(P_i)$.
**end**
Set $R \leftarrow C_3 r\sqrt{\log(t)}$

`// Return the subspace.`
Let $\sigma \leftarrow \frac{4R\sqrt{q}\ln(q/\delta)}{\varepsilon t}$.
**for** *each* $i \in [q]$ **do**
  Truncate all $p_i^j$'s to within $B_R(c_i)$.
  Let $\widehat{p}_i \leftarrow \sum_{j=1}^{t} p_i^j + \mathcal{N}(0, \sigma^2 \mathbb{I}_{d\times d})$.
**end**
Let $\widehat{P} \leftarrow (\widehat{p}_i, \ldots, \widehat{p}_q)$.
Let $\widehat{\Pi}$ be the projection matrix of the top-$k$ subspace of $\widehat{P}$.
**return** $\widehat{\Pi}$.

**Lemma 24** *Algorithm 6 is $(2\varepsilon, 2\delta)$-DP.*

**Proof** The first aggregation step of finding $c_i$ is $(\varepsilon, \delta)$-DP by Theorem 14 and Lemma 10. In the mean estimation step, because we are restricting all the $p_i^j$'s to within $B_R(c_i)$, the sensitivity is $2R$,

since by changing one point in $X$, we can change exactly one $p_i^j$ by $2R$ in $\ell_2$ norm. Therefore, by Lemmata 12 and 10, this step is $(\varepsilon, \delta)$-DP. The final privacy guarantee follows from Lemma 10. ■

**Lemma 25 (Lemma 4.9 of Singhal and Steinke (2021) Modified)** *Let $\widehat{\Pi}$ be the projection matrix as defined in Algorithm 6, $n$ be the total number of samples, and $0 < \psi < 1$. If*

$$t \geq O\left(\frac{\sqrt{dk} \cdot \operatorname{polylog}(d, k, \frac{1}{\varepsilon}, \frac{1}{\delta})}{\varepsilon}\right) \quad \textit{and} \quad m \geq O\left(\frac{d \cdot \operatorname{polylog}(d, k, \frac{1}{\varepsilon}, \frac{1}{\delta})}{\psi^2}\right),$$

*which implies that*

$$n \geq O\left(\frac{d^{1.5}\sqrt{k} \cdot \operatorname{polylog}(d, k, \frac{1}{\varepsilon}, \frac{1}{\delta})}{\varepsilon\psi^2}\right),$$

*then $\|\Pi - \widehat{\Pi}\| \leq \psi\gamma$ with probability at least 0.7.*

**Proof** For each $i \in [q]$, let $p_i^*$ be the projection of $p_i$ on to the subspace spanned by $\Sigma_k$, $\widehat{p}_i$ be as defined in the algorithm, and $p_i^j$ be the projection of $p_i$ on to the subspace spanned by the $j^{\text{th}}$ subset of $X$. From the analysis in Singhal and Steinke (2021), we know that for a fixed $i$, all $p_i^j$'s are contained in a ball of radius $r$. Therefore, all points in $P_i$ lie in a ball of radius $r$. Therefore, by the guarantees of GoodCenter (Theorem 14), $B_R(c_i)$ contains all of $p_i^j$'s, such that $R \in O(r\sqrt{\ln(t)})$. This implies that $p_i^*$ is also contained within $B_R(c_i)$.

Now, let $P = (p_1^*, \ldots, p_q^*)$. Suppose $\widehat{P} = (\widehat{p}_1, \ldots, \widehat{p}_q)$ as defined in the algorithm. Then by above, $\widehat{P} = P + E$ for some $E \in \mathbb{R}^{d \times q}$. The goal is to show that $\|\Pi - \widehat{\Pi}\| \leq O(\frac{\|E\|}{\sqrt{k}}) \leq O(\gamma\psi)$. We set $E = E_0 + E_1$, where $E_0$ is the sampling error, and $E_1$ is the error due to privacy, In other words, let $\overline{p}_i = \frac{1}{t}\sum_{j=1}^{t} p_i^j$ and $\overline{P} = (\overline{p}_1, \ldots, \overline{p}_q)$; then $E_0 = \overline{P} - P$ and $E_1 = \widehat{P} - \overline{P}$.

We first analyse $\|E_0\|$. Let $\Pi^j$ be the subspace spanned by the $j$-th subsample. We know that the subspaces spanned by $P^j = (p_1^j, \ldots, p_q^j)$ and the $j$-th subsample are the same. Therefore, $\|\Pi - \Pi^j\| \in \Theta(\frac{\|P^j - P\|}{\sqrt{k}}) \leq \gamma\sqrt{\frac{d}{m}}$ by Lemmata 2.4 and 4.5, and Corollary 2.7 of Singhal and Steinke (2021). Therefore,

$$\begin{aligned}
\frac{\|E_0\|}{\sqrt{k}} &\leq O\left(\frac{\|\overline{P} - P\|}{\sqrt{k}}\right) \\
&= O\left(\frac{\|\frac{1}{t}\sum_{j=1}^{t} P^j - P\|}{\sqrt{k}}\right) \\
&\leq O\left(\frac{\frac{1}{t}\sum_{j=1}^{t}\|P^j - P\|}{\sqrt{k}}\right) \\
&\leq O\left(\frac{1}{t} \cdot \sum_{j=1}^{t} \gamma\sqrt{\frac{d}{m}}\right)
\end{aligned}$$

$$\leq O\left(\gamma\sqrt{\frac{d}{m}}\right)$$
$$\in O(\gamma\psi). \qquad \text{(By our sample complexity.)}$$

Next, we analyse $\|E_1\|$. $E_1$ is a matrix with i.i.d. entries from $\mathcal{N}(0, \sigma^2)$. Therefore, by Lemma 2.4 of Singhal and Steinke (2021), we have

$$\frac{\|E_1\|}{\sqrt{k}} \in O\left(\frac{\sigma\sqrt{d}}{\sqrt{k}}\right)$$
$$\in O\left(\frac{r\sqrt{\log(t)kd\log(k/\delta)}}{\varepsilon t\sqrt{k}}\right)$$
$$\in O\left(\frac{r}{\sqrt{k}}\right) \qquad \text{(By our sample complexity.)}$$
$$\in O\left(\gamma\sqrt{\frac{d}{m}}\right)$$
$$\in O(\gamma\psi). \qquad \text{(By our sample complexity.)}$$

Therefore, we have $\|E\| \in O(\gamma\psi)$.

Let $E = E_P + E_{\overline{P}}$, where $E_P$ is the component of $E$ in the subspace spanned by $P$, and $E_{\overline{P}}$ be the orthogonal component. Let $P' = P + E_P$. We will be analysing $\widehat{P}$ with respect to $P'$.

As before, we will try to bound the distance between the subspaces spanned by $P'$ and $\widehat{P}$. The quantities $a, z_{12}$ remain unchanged, but $b, z_{21}$ change.

$$b \leq \|E_{\overline{P}}\|$$
$$z_{21} \leq \|E_{\overline{P}}\|$$

Therefore, we get the final error:

$$\|\Pi - \widehat{\Pi}\| \leq \frac{az_{21} + bz_{12}}{a^2 - b^2 - \min\{z_{12}^2, z_{21}^2\}}$$
$$\leq \gamma\psi.$$

This completes our proof. ■

This gives us the following theorem about Algorithm 6.

**Theorem 26** *Let $\Sigma \in \mathbb{R}^{d\times d}$ be a symmetric, PSD matrix, such that for $1 \leq k < d$ and $\gamma < 1$, $\frac{\lambda_{k+1}(\Sigma)}{\lambda_k(\Sigma)} < \gamma^2$. Suppose $\Pi$ is the subspace spanned by the top $k$ eigenvectors of $\Sigma$. Then for all $\varepsilon, \delta, \beta, \psi > 0$, there exists an $(\varepsilon, \delta)$-DP algorithm, that takes*

$$n \geq O\left(\frac{d^{1.5}\sqrt{k}\cdot \mathrm{polylog}(d, k, \frac{1}{\varepsilon}, \frac{1}{\delta}, \frac{1}{\beta})}{\varepsilon\psi^2}\right)$$

*samples from $\mathcal{N}(0, \Sigma)$, and outputs a projection matrix $\widehat{\Pi}$, such that with probability at least $1 - O(\beta)$, $\|\Pi - \widehat{\Pi}\| \leq \psi\gamma$.*

**Proof** The claim, but with error probability $0.35$, is guaranteed from Lemma 25. Now, we just have to boost the success probability. This can be done using Theorem 4.10 of Singhal and Steinke (2021). ■

## Appendix D. Naive Estimator

In this section, we revisit the naive estimator presented in Kamath et al. (2019a) for well-conditioned gaussians. We present a slightly modified version of the algorithm and its analysis that is tailored to our setting.

**Algorithm 7:** Naive Private Gaussian Covariance Estimation $\mathrm{NaiveEstimator}_{\varepsilon,\delta,\beta}(X)$

**Input:** A set of $n$ samples $X_1, \ldots, X_n$ from an unknown Gaussian. Parameters $\varepsilon, \delta, \beta > 0$
**Output:** A covariance matrix $M$.

Set $\hat{\lambda}_1, \ldots, \hat{\lambda}_d \leftarrow \mathrm{EigenvalueEstimator}_{\varepsilon,\delta,\beta}(X)$.
Set $\kappa \leftarrow 4\hat{\lambda}_1$.
Let $S \leftarrow \left\{i \in [n] : \|X_i\|_2^2 \le O(d\kappa_2 \log(n/\beta))\right\}$
Let

$$\sigma \leftarrow \Theta\left(\frac{d\kappa_2 \log(\frac{n}{\beta})\sqrt{\log(1/\delta)}}{n\varepsilon}\right)$$

Let $M' \leftarrow \frac{1}{n}\sum_{i\in S} X_i X_i^\top + N$ where $N_{ij} \sim \mathcal{N}(0, \sigma^2)$
Let $M$ be the Euclidean projection of $M'$ on the PSD cone.
**return** $M$

**Lemma 27 (Analysis of** NaiveEstimator**)** *For every $\varepsilon, \delta, \beta, \kappa_1, \kappa_2, n$, $\mathrm{NaiveEstimator}_{\varepsilon,\delta,\beta}(X)$ satisfies $(\varepsilon, \delta)$-DP, and if $X_1, \ldots, X_n$ are sampled i.i.d. from $\mathcal{N}(0, \Sigma)$ for $\kappa_1 \mathbb{I} \preceq \Sigma \preceq \kappa_2 \mathbb{I}$ and satisfy (4)–(6), then with probability at least $1 - O(\beta)$, it outputs $M$ so that:*

1. $\|\Sigma - M\|_\Sigma \le O\left(\frac{\kappa_2 d^2 \log(n/\beta)\log(1/\beta)\sqrt{\log(1/\delta)}}{\kappa_1 n\varepsilon} + \sqrt{\frac{d^2+\log(1/\beta)}{n}}\right)$.
2. $\|\Sigma - M\|_2 \le O\left(\kappa_2\sqrt{\frac{d+\log(1/\beta)}{n}} + \frac{\kappa_2 d^{3/2}\log(n/\beta)\log(1/\beta)\sqrt{\log(1/\delta)}}{n\varepsilon}\right)$.

**Proof** We prove the lemma by proving the privacy and accuracy guarantees of Algorithm 7. We first prove the privacy guarantee. Given two neighboring data sets $X, X'$ of size $n$ which differ in that one contains $X_i$ and the other contains $X_i'$, the truncated empirical covariance of these two data sets can change in Frobenius norm by at most

$$\left\|\frac{1}{n}\left(X_i X_i^\top - X_i'(X_i')^\top\right)\right\|_F \le \frac{1}{n}\|X_i\|_2^2 + \frac{1}{n}\|X_i'\|_2^2 \le O\left(\frac{d\kappa_2 \log(n/\beta)}{n}\right).$$

Thus the privacy guarantee follows immediately from Lemma 12.

We now prove correctness. First, we have:

$$\begin{aligned}
\|\Sigma - M\|_\Sigma &\le \left\|M - M'\right\|_\Sigma + \left\|M' - \Sigma\right\|_\Sigma \\
&\le \|M - M'\|_F \|\Sigma^{-1}\|_2 + \|M' - \Sigma\|_\Sigma \\
&\le \sqrt{d}\kappa_1^{-1} \left\|M - M'\right\|_2 + \left\|M' - \Sigma\right\|_\Sigma \\
&\overset{(a)}{\le} \sqrt{d}\kappa_1^{-1} \|N\|_2 + \left\|M' - \Sigma\right\|_\Sigma \\
&\le \sqrt{d}\kappa_1^{-1} \|N\|_2 + \left\|\frac{1}{n}\sum_{i=1}^n X_i X_i^\top - \Sigma\right\|_\Sigma + \|N\|_\Sigma \\
&\overset{(b)}{\le} \sqrt{d}\kappa_1^{-1} \|N\|_2 + \left\|\frac{1}{n}\sum_{i=1}^n X_i X_i^\top - \Sigma\right\|_\Sigma + \frac{1}{\kappa_1}\|N\|_F \\
&\overset{(c)}{\le} O\left(\frac{\kappa_2 d^2 \log(n/\beta)\log(1/\beta)\sqrt{\log(1/\delta)}}{\kappa_1 n\varepsilon}\right) \\
&\quad + O\left(\sqrt{\frac{d^2 + \log(1/\beta)}{n}}\right) + O\left(\frac{\kappa_2 d^2 \log(n/\beta)\log^{1/2}(1/\beta)\sqrt{\log(1/\delta)}}{n\kappa_1\varepsilon}\right) \\
&= O\left(\frac{\kappa_2 d^2 \log(n/\beta)\log(1/\beta)\sqrt{\log(1/\delta)}}{\kappa_1 n\varepsilon} + \sqrt{\frac{d^2 + \log(1/\beta)}{n}}\right),
\end{aligned}$$

where $(a)$ holds because $\frac{1}{n}\sum_{i\in S} X_i X_i^\top$ is PSD, and $M$ is the projection of $M' = \frac{1}{n}\sum_{i\in S} X_i X_i^\top + N$ onto the PSD cone, so by Weyl's inequality, the zeroed out eigenvalues have to be at most $\|N\|_2$; $(b)$ is by the inequality $\left\|B^{\frac{1}{2}} A B^{\frac{1}{2}}\right\|_F \le \|B\|_2 \|A\|_F$ and the fact that $\Sigma \succeq \kappa_1 \mathbb{I}$; and $(c)$ is due to Facts 21 and 18.

Additionally, we have:

$$\begin{aligned}
\|\Sigma - M\|_2 &\le \left\|\Sigma - M'\right\|_2 + \left\|M' - M\right\|_2 \\
&\le \left(\left\|\frac{1}{n}\sum_{i=1}^n X_i X_i^\top - \Sigma\right\|_2 + \|N\|_2\right) + \|N\|_2 \\
&\overset{(c)}{\le} \|\Sigma\| \left\|\frac{1}{n}\sum_{i=1}^n \left(\Sigma^{-\frac{1}{2}} X_i\right)\left(\Sigma^{-\frac{1}{2}} X_i\right)^\top - \mathbb{I}\right\|_2 + 2\|N\|_2 \\
&\overset{(d)}{\le} O\left(\kappa_2\sqrt{\frac{d + \log(1/\beta)}{n}}\right) + 2\|N\|_2 \\
&\overset{(e)}{\le} O\left(\kappa_2\sqrt{\frac{d + \log(1/\beta)}{n}} + \frac{\kappa_2 d^{3/2}\log(n/\beta)\log(1/\beta)\sqrt{\log(1/\delta)}}{n\varepsilon}\right).
\end{aligned}$$

where $(c)$ is by the sub-multiplicative property of the spectral norm, $(d)$ is by Fact 21 and $(e)$ is by Theorem 16. ■

**Corollary 28** *Suppose $X_1, \ldots, X_n$ are sampled i.i.d. from $\mathcal{N}(0, \Sigma)$ for $\kappa_1 \mathbb{I} \preceq \Sigma \preceq \kappa_2 \mathbb{I}$ and satisfy (4)–(6). Let $1 \leq k \leq d$ be the largest number, such that $\lambda_k(\Sigma) \geq \overline{\gamma}^2 \lambda_1(\Sigma)$ for $0 < \overline{\gamma} \leq 1$. If*

$$n \geq O\left(\frac{d^{3/2} \cdot \mathrm{polylog}(1/\beta, 1/\delta)}{\varepsilon \overline{\gamma}^2}\right),$$

*then with probability at least $1 - O(\beta)$, $\mathrm{NaiveEstimator}_{\varepsilon,\delta,\beta,\kappa}(X)$ outputs $M$ so that for each $1 \leq i \leq k$, $\lambda_i(M) \in \left[\frac{\lambda_i(\Sigma)}{2}, 2\lambda_i(\Sigma)\right]$.*

**Proof** By Lemma 22 and our sample complexity, each eigenvalue of $\Sigma$ is estimated correctly by the empirical covariance up to a factor of $\sqrt{2}$. Now, by Lemma 16 and our sample complexity, $\|N\|_2 \in \tilde{O}(\kappa_2 \overline{\gamma}^2)$. By applying Weyl's inequality (Lemma 19) for each eigenvalue $1 \leq i \leq k$, the claim follows. Note, that the eigenvalues corresponding to $i > k$ may not be estimated accurately, but because $\|N\|_2$ is bounded, the corresponding estimates in $Z$ cannot be more than $2\lambda_k(\Sigma)$ by Weyl's inequality. ■

The following is an immediate consequence of Lemma 27.

**Theorem 29** *For every $\varepsilon, \delta, \alpha, \beta, > 0, \kappa_2 \geq \kappa_1 > 0$, the algorithm $\mathrm{NaiveEstimator}_{\varepsilon,\delta,\beta}$ is $(\varepsilon, \delta)$-DP, and when given*

$$n \geq O\left(\frac{d^2 + \log(1/\beta)}{\alpha^2} + \frac{\kappa_2 d^2 \log(n/\beta) \log(1/\beta) \sqrt{\log(1/\delta)}}{\kappa_1 \alpha \varepsilon}\right),$$

*samples from $\mathcal{N}(0, \Sigma)$ satisfying $\kappa_1 \mathbb{I} \preceq \Sigma \preceq \kappa_2 \mathbb{I}$, with probability at least $1 - O(\beta)$, it returns $M$ such that $\|\Sigma - M\|_\Sigma \leq O(\alpha)$.*