Private Estimation with Public Data*

Alex Bie
University of Waterloo
yabie@uwaterloo.ca

Gautam Kamath
University of Waterloo
g@csail.mit.edu

Vikrant Singhal
University of Waterloo
vikrant.singhal@uwaterloo.ca

Abstract

We initiate the study of differentially private (DP) estimation with access to a small amount of public data. For private estimation of d-dimensional Gaussians, we assume that the public data comes from a Gaussian that may have vanishing similarity in total variation distance with the underlying Gaussian of the private data. We show that under the constraints of pure or concentrated DP, $d + 1$ public data samples are sufficient to remove any dependence on the range parameters of the private data distribution from the private sample complexity, which is known to be otherwise necessary without public data. For separated Gaussian mixtures, we assume that the underlying public and private distributions are the same, and we consider two settings: (1) when given a dimension-independent amount of public data, the private sample complexity can be improved polynomially in terms of the number of mixture components, and any dependence on the range parameters of the distribution can be removed in the approximate DP case; (2) when given an amount of public data linear in the dimension, the private sample complexity can be made independent of range parameters even under concentrated DP, and additional improvements can be made to the overall sample complexity.

1 Introduction

Differential privacy (DP) [DMNS06] guarantees the privacy of *every* point in a dataset. This is a strong requirement, and often gives rise to qualitatively new requirements when performing private data analysis. For instance, for the problem of private mean estimation (under pure or concentrated DP), the analyst must specify a guess for the unknown parameter, and needs more data to get an accurate estimate depending on how good their guess is. This cost can be prohibitive in cases where the data domain may be unfamiliar.

Fortunately, in many cases, it is natural to assume that there exists an additional *public* dataset. This public dataset may vary in both size and quality. For example, one could imagine that a fraction of users opt out of privacy considerations, giving a small set of public in-distribution data. Alternatively, it is common to pretrain models on large amounts of public data from the web, which may be orders of magnitude larger than the private data but significantly out of distribution. In a variety of such settings, this public data can yield dramatic theoretical and empirical improvements to utility in private data analysis (see discussion in Section 1.3). We seek to answer the following question:

How can one take advantage of public data for private estimation?

We initiate the study of differentially private statistical estimation with a supplementary public dataset. In particular, our goal is to understand when a small amount of public data can significantly reduce the cost of private estimation.

*Authors are listed in alphabetical order.

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

1.1 Problem Formulation

We investigate private estimation with public data through the canonical problems of estimating Gaussians and mixtures of Gaussians. We focus on parameter estimation from i.i.d. samples drawn from public and private data distributions.

We wish to design estimation algorithms that take as input both public data and private data, and are private with respect to private data. Formally, we say a randomized algorithm $\mathcal{A} : \mathcal{X}^m \times \mathcal{X}^n \to \mathcal{Y}$ taking as input public data $\widetilde{X} \in \mathcal{X}^m$ and private data $X \in \mathcal{X}^n$ is *private with respect to private data* if for any setting of the public data $\widetilde{X}$, $\mathcal{A}(\widetilde{X}, \cdot) : \mathcal{X}^n \to \mathcal{Y}$ is differentially private.[2]

Gaussians. Let $\mu, \widetilde{\mu} \in \mathbb{R}^d$ and positive definite $\Sigma, \widetilde{\Sigma} \in \mathbb{R}^{d \times d}$ be unknown, with $\mathrm{d_{TV}}(\mathcal{N}(\mu, \Sigma), \mathcal{N}(\widetilde{\mu}, \widetilde{\Sigma})) \leq \gamma$ for some known $0 \leq \gamma < 1$. Then for $0 < \alpha, \beta < 1$, given public samples from $\mathcal{N}(\widetilde{\mu}, \widetilde{\Sigma})$ and private samples from $\mathcal{N}(\mu, \Sigma)$, we would like to output $\hat{\mu}, \widehat{\Sigma}$ such that with probability $\geq 1 - \beta$, $\mathrm{d_{TV}}(\mathcal{N}(\mu, \Sigma), \mathcal{N}(\hat{\mu}, \widehat{\Sigma})) \leq \alpha$.

Mixtures of Gaussians. We represent a Gaussian mixture D by a sequence of k tuples $((\mu_1, \Sigma_1, w_1), \ldots, (\mu_k, \Sigma_k, w_k))$, each representing the mean, covariance matrix, and mixing weight of a component. Then for $0 < \alpha, \beta < 1$, given public and private samples drawn from D, we would like to output a Gaussian mixture $\widehat{D} = \{(\widehat{\mu}_1, \widehat{\Sigma}_1, \widehat{w}_1), \ldots, (\widehat{\mu}_k, \widehat{\Sigma}_k, \widehat{w}_k))\}$ such that with probability $\geq 1 - \beta$, for each $i \in [k]$, there exists a distinct $j \in [k]$, such that $\mathrm{d_{TV}}(\mathcal{N}(\mu_i, \Sigma_i), \mathcal{N}(\widehat{\mu}_j, \widehat{\Sigma}_j)) \leq \alpha$ and $|w_i - \widehat{w}_j| \leq \alpha/k$. This implies $\mathrm{d_{TV}}(D, \widehat{D}) \leq O(\alpha)$, and we refer to this stronger notion of component-wise estimation as (α, β)-learning (Definition A.10).

1.2 Results

Gaussians. In the case of pure and concentrated DP, exactly $d + 1$ public samples are sufficient to remove any dependence on "range parameters" in the private sample complexity, even when the underlying Gaussian distribution of the public data bears *almost no similarity* to that of the private data. We start with following result for the scenario when the two distributions are the same.

Theorem 1.1. *For all $\alpha, \beta, \varepsilon > 0$ and $d \geq 1$ there exists a computationally efficient $\frac{\varepsilon^2}{2}$-zCDP (and a computationally inefficient ε-DP) algorithm $\mathcal{M}$ that takes $d + 1$ public samples and n private samples from a Gaussian D over $\mathbb{R}^d$, which is private with respect to private data, such that if $n = O\left(\frac{d^2}{\alpha^2} + \frac{d^2}{\alpha\varepsilon}\right) \cdot \mathrm{polylog}\left(d, \frac{1}{\beta}, \frac{1}{\alpha}\right)$, then $\mathcal{M}$ estimates the parameters of D to within TV error α with probability $\geq 1 - \beta$.*

Without public data, the analyst would be required to specify a bound R on the ℓ_2 norm of the mean and a bound K on the condition number of the covariance, and the private sample complexity would scale logarithmically in terms of these parameters (which we denote as *range parameters*). Thus, this cost could be arbitrarily large, or even infinite, given poor *a priori* knowledge of the problem. Note that these requirements are not due to suboptimality of current private algorithms. Packing lower bounds for estimation under pure and concentrated DP show that dependence on range parameters is an inherent cost of privacy. However, this dependence can be entirely eliminated at the price of $d + 1$ public data points, which is less public data than the non-private $\Theta(\frac{d^2}{\alpha^2})$ sample complexity of the problem. For estimating Gaussians with identity or known covariance (that is, mean estimation) it turns out that just *one* public sample suffices to remove all dependence on range parameters.

The next natural question to ask is: what if our public data does not come from the same distribution as our private data? Indeed, in practical settings, there may be significant domain shift between data that is publicly available and the private data of interest. Our next result shows that we can relax this assumption: it suffices that our public data comes from another Gaussian that is at most $\gamma < 1$ away in TV distance from the underlying private data distribution.

Theorem 1.2. *For all $\alpha, \beta, \varepsilon > 0$, $d \geq 1$, and $0 \leq \gamma < 1$, there exists a computationally efficient $\frac{\varepsilon^2}{2}$-zCDP (and a computationally inefficient ε-DP) algorithm $\mathcal{M}$ that takes $d + 1$ public samples from*

[2] Previous work exploring private algorithms with additional public data, such as [BNS16a; BCMNUW20], adopt this privacy requirement for their algorithms.

a Gaussian $\widetilde{D}$ over $\mathbb{R}^d$ and n private samples from a Gaussian D over $\mathbb{R}^d$ with $\mathrm{d_{TV}}(D, \widetilde{D}) \leq \gamma$, which is private with respect to private data, such that if $n = O\left(\frac{d^2}{\alpha^2} + \frac{d^2}{\alpha\varepsilon}\right) \cdot \mathrm{polylog}\left(d, \frac{1}{\beta}, \frac{1}{\alpha}, \frac{1}{1-\gamma}\right)$, then $\mathcal{M}$ estimates the parameters of D to within TV error α with probability $\geq 1 - \beta$.

The upper bound on the TV gap γ results in the sample complexity picking up an extra $\mathrm{polylog}(\frac{1}{1-\gamma})$ factor. For instance, even when the public and private distributions have total variation distance $1 - 1/d^{100}$ (that is, they are *almost entirely dissimilar*), we incur only an extra $\mathrm{poly}\log d$ factor in the sample complexity. In particular, note that the public dataset may be simultaneously quite small and significantly out of distribution with respect to the private data.

Mixtures of Gaussians. We focus on estimation of mixtures of k *separated* Gaussians in d dimensions. Specifically, we assume that, given mixture components $\mathcal{N}(\mu_i, \Sigma_i)$ with respective mixing weights w_i for $i \in [k]$ (with $w_i \geq w_{\min}$), then for each $i \neq j$,

$$\|\mu_i - \mu_j\|_2 \geq \widetilde{\Omega}\left(\sqrt{k} + \frac{1}{\sqrt{w_i}} + \frac{1}{\sqrt{w_j}}\right) \cdot \max\left\{\sqrt{\|\Sigma_i\|_2}, \sqrt{\|\Sigma_j\|_2}\right\}. \tag{1}$$

Private estimation of such mixtures of Gaussians (without public data) was previously studied by [KSSU19]. Their algorithms proceed in two stages: first, a clustering step to isolate the samples from each component, and an estimation step, which estimates the parameters of each individual Gaussian using the isolated samples from each component. Our algorithms follow a similar structure, but employ the public data at appropriate points in the procedure to enable substantial savings.

In all the results that follow, $\widetilde{O}$ hides the $\mathrm{polylog}$ factors of d, k, the inverse of error parameters $\frac{1}{\alpha}, \frac{1}{\beta}$, and the inverse of privacy parameters $\frac{1}{\varepsilon}, \frac{1}{\delta}, \frac{1}{\rho}$. We provide two flavours of results for learning mixtures of Gaussians. First, we investigate the setting when we have a very small, dimension-independent amount of public data.

Theorem 1.3. *For all $\alpha, \beta, \varepsilon, \delta > 0$, there exists an (ε, δ)-DP algorithm $\mathcal{M}$ that takes n private samples and m public samples from a Gaussian mixture D, satisfying (1), which is private with respect to private data, such that if $m = O\left(\frac{\log(k/\beta)}{w_{\min}}\right)$ and $n = \widetilde{O}\left(\frac{d^2}{w_{\min}\alpha^2} + \frac{d^2}{w_{\min}\alpha\varepsilon} + \frac{d^{1.5}k^{2.5}}{\varepsilon}\right)$, then $\mathcal{M}$ (α, β)-learns D.*

Theorem 1.4. *For all $\alpha, \beta, \rho > 0$, there exists a ρ-zCDP algorithm $\mathcal{M}$ that takes n private samples and m public samples from a Gaussian mixture D satisfying (1), which is private with respect to private data, such that if $\forall i$, $\|\mu_i\|_2 \leq R$ and $\mathbb{I} \preceq \Sigma_i \preceq K\mathbb{I}$, and $m = O\left(\frac{\log(k/\beta)}{w_{\min}}\right)$ and $n = \widetilde{O}\left(\frac{d^2}{w_{\min}\alpha^2} + \frac{d^2}{w_{\min}\alpha\sqrt{\rho}} + \frac{d^{1.5}k^{2.5}}{\sqrt{\rho}}\right) \cdot \mathrm{polylog}\,(K, R)$, then $\mathcal{M}$ (α, β)-learns D.*

Our results differ from the prior work of [KSSU19] in the following ways: (1) we have results for both approximate DP and zCDP, as opposed to just approximate DP in [KSSU19]; (2) the private sample complexity due to the clustering step does not involve any dependence on the range parameters; and (3) the sample complexity due to the clustering step is now $O(d^{1.5}k^{2.5})$, as opposed to $O(d^{1.5}k^{9.06})$, which is a significant improvement, and shows up only due to applications of private PCA under better sensitivity guarantees. As far as the private sample complexity in the component estimation process is concerned, in the case of approximate DP, we have expressions independent of distribution parameters owing to the recent Gaussian learner by [AL21].

Finally, we investigate Gaussian mixture estimation with an amount of public data which is linear in the dimension.

Theorem 1.5. *For all $\alpha, \beta, \varepsilon, \delta > 0$, there exists an (ε, δ)-DP algorithm $\mathcal{M}$ that takes n private samples and m public samples from a Gaussian mixture D satisfying (1), which is private with respect to private data, such that if $m = O\left(\frac{d\log(k/\beta)}{w_{\min}}\right)$ and $n = \widetilde{O}\left(\frac{d^2}{w_{\min}\alpha^2} + \frac{d^2}{w_{\min}\alpha\varepsilon}\right)$, then $\mathcal{M}$ (α, β)-learns D.*

Theorem 1.6. *For all $\alpha, \beta, \rho > 0$, there exists a ρ-zCDP algorithm $\mathcal{M}$ that takes n private samples and m public samples from a Gaussian mixture D satisfying (1), which is private with respect to private data, such that if $m = O\left(\frac{d\log(k/\beta)}{w_{\min}}\right)$ and $n = \widetilde{O}\left(\frac{d^2}{w_{\min}\alpha^2} + \frac{d^2}{w_{\min}\alpha\sqrt{\rho}}\right)$, then $\mathcal{M}$ (α, β)-learns D.*

Using $\widetilde{O}(d/w_{\min})$ public samples establishes all the advantages that using $\widetilde{O}(1/w_{\min})$ public samples gives, but additionally improves the private sample complexity even further. As we will describe in the technical overview, the clustering process is done entirely using the public data, and ends up removing the term due to private PCA from the private sample complexity in both approximate DP and zCDP regimes. Additionally, in the zCDP case, we do not have any dependence on distribution range parameters even in the component estimation process because we can use results from Theorem 1.1 this time.

1.3 Related Work

A large part of our work focuses on using public data to obviate the dependence on range parameters in private estimation. Understanding these dependences without public data has been a topic of significant study. [KV18] investigated private estimation of univariate Gaussians, showing that logarithmic dependences on the range parameters were both necessary and sufficient for pure differential privacy, but by using stability-based histograms [KKMN09; BNS16b], they could be removed under approximate differential privacy. Similar results have been shown in the multivariate setting: while logarithmic dependences are necessary and sufficient under pure or concentrated DP [KLSU19], they can be removed under approximate DP [AAK21; KMSSU21; TCKMS21; AL21; KMV21; LKO21]. More broadly, a common theme is that the dependences which are intrinsic to pure or concentrated DP can be eliminated by relaxing to approximate DP. Our results demonstrate that instead of relaxing the privacy definition for all the data, one can achieve the same goal by employing a much smaller amount of public data.

Our investigation fits more broadly into a line of work employing public data for private data analysis. Works, both theoretical and empirical, investigate the role of public data in private query release, synthetic data generation, and prediction [JE13; BNS16a; ABM19; NB20; BCMNUW20; BMN20; LVSUW21]. For distribution-free classification with public and private data, it was shown in [ABM19] that, roughly speaking, if a hypothesis class is privately learnable with a small amount of public data ($o(1/\alpha)$), it is privately learnable with no public data. For density estimation, we show that the family of unbounded Gaussians is learnable with a small amount of public data (α-independent), despite the fact that it is not learnable without public data.

Additionally, a number of empirical works study the efficacy of public data in private machine learning, including methods such as public pre-training [ACGMMTZ16; PCSTE19; TB21; LWAFF21; YZCL21; LTLH22; Yu+22], using public data to compute useful statistics about the private gradients [ZWB21; YZCL21; KRRT21; AGMRSSSTT21], or using unlabeled public data to train a student model [PAEGT17; PSMRTE18; BTGT18].

Another setting with mixed privacy guarantees is when different users may require local versus central differential privacy [AKZHL17]. Mean estimation has also been studied in this setting [ADK19].

1.4 Organization

The remainder of the paper is organized as follows. In Section 2, we discuss our results for estimating multivariate Gaussians. After that in Section 3, we give a technical overview of our results for estimating mixtures of Gaussians. We present all the relevant notations and preliminaries about differential privacy, Gaussian mixtures, and techniques like PCA, along with a few useful concentration inequalities in Appendix A. In Appendix B, we provide further technical details that are missing from Section 2. We provide all our technical results about mixtures of Gaussians from Section 3, including algorithms, theorems, and proofs, in Appendix C. Finally, in Appendix D we present some proof-of-concept numerical simulations demonstrating the effectivenss of public data for private estimation.

2 Estimating Gaussians

We discuss our results for estimating multivariate Gaussians with access to a limited amount of public data. First, as a warm-up, we consider the case of estimating the mean of an identity (or known) covariance Gaussian, and give a ρ-zCDP private algorithm that uses 1 public sample. We then generalize the high-level approach taken there to arrive at a private algorithm that uses $(d+1)$ public samples for general Gaussians. Furthermore, we show that we can relax the condition that the

public and the private data come from the same distribution – the public data can come from another Gaussian that is at most γ far away in TV distance.

2.1 Warmup: Identity Covariance Gaussians with 1 Public Sample

In this setting, we are given a single public sample $\widetilde{X}$, along with private samples $X_1, \ldots, X_n$, where $\widetilde{X}$ and the X_j are drawn from a d-dimensional Gaussian $\mathcal{N}(\mu, \mathbb{I})$ independently. Our main observation is that a single public sample is sufficient to get a "good enough" coarse estimate of μ, allowing us to apply existing private algorithms for a finer estimate. This is captured in the following claim, which is a standard fact regarding norm concentration of Gaussian vectors (Lemma A.3).

Claim 2.1. *Suppose we get a sample $\widetilde{X} \sim \mathcal{N}(\mu, \mathbb{I})$. Then with probability at least $1-\beta$, $\|\mu - \widetilde{X}\|_2 \leq \sqrt{d + 2\sqrt{d\log(1/\beta)} + 2\log(1/\beta)}$.*

We restate the guarantees of an existing folklore ρ-zCDP "clip-and-noise" algorithm via the Gaussian mechanism for estimating identity covariance Gaussians.

Lemma 2.2 (zCDP Private Gaussian Mean Estimation)**.** *For all $\alpha, \beta, \rho > 0$, there exists a ρ-zCDP algorithm that takes $n = \widetilde{O}\left(\frac{d+\log(1/\beta)}{\alpha^2} + \frac{\sqrt{d}(R+\sqrt{d})\log(1/\beta)}{\alpha\sqrt{\rho}}\right)$ private samples from $\mathcal{N}(\mu, \mathbb{I})$ over $\mathbb{R}^d$, where $\|\mu\|_2 \leq R$, and outputs an estimate $\widehat{\mu} \in \mathbb{R}^d$, such that with probability at least $1-\beta$, $\mathrm{d_{TV}}(\mathcal{N}(\mu, \mathbb{I}), \mathcal{N}(\widehat{\mu}, \mathbb{I})) \leq \alpha$.*

The above "clip-and-noise" mechanism simply clips all the points in the dataset to within radius $\lambda = R + \sqrt{d + 2\sqrt{d\log(n/\beta)} + 2\log(n/\beta)}$ around the origin, and outputs the noisy empirical mean of the clipped points via the Gaussian mechanism.

Claim 2.1, combined with the result above, yields a 1-public-sample, private mean estimation algorithm via the following three-step process: (1) form shifted private dataset $Y_1, \ldots, Y_n$ by taking each $Y_j := X_j - \widetilde{X}$; (2) run "clip-and-noise" on $Y_1, \ldots, Y_n$, setting $R = \sqrt{d + 2\sqrt{d\log(2/\beta)} + 2\log(2/\beta)} = O(\sqrt{d\log(1/\beta)})$ and target failure probability $\beta/2$, to obtain $\widehat{\mu}_Y$; and (3) output $\widehat{\mu} := \widehat{\mu}_Y + \widetilde{X}$.

By a union bound, with probability $\geq 1-\beta$, the distribution of $Y_1, \ldots, Y_n$ satisfies tight boundedness independent of the range parameters, and furthermore, succeeds in outputting a good estimate $\widehat{\mu}_Y$ for $\mu_Y := \mathbb{E}Y_1$. The last step undoes the shift to yield a good estimate $\widehat{\mu}$ for the original μ. Privacy is preserved with respect to private data, since we are doing entry-wise pre-processing and post-processing with public data. For full version of this argument (which includes the ε-DP case), see Theorem B.2.

This yields a private sample complexity that no longer depends on the range bound R, allowing us to use a fixed sample size to estimate the family of all identity covariance Gaussians over $\mathbb{R}^d$.

Theorem 2.3 (zCDP Gaussian Mean Estimation with Public Data)**.** *For all $\alpha, \beta, \rho > 0$, there exists a ρ-zCDP algorithm that is private with respect to the private samples, takes 1 public sample and $n = \widetilde{O}\left(\frac{d+\log(1/\beta)}{\alpha^2} + \frac{d\log^{1.5}(1/\beta)}{\alpha\sqrt{\rho}}\right)$ private samples from $\mathcal{N}(\mu, \mathbb{I})$ over $\mathbb{R}^d$, and outputs an estimate $\widehat{\mu} \in \mathbb{R}^d$, such that with probability at least $1-\beta$, $\mathrm{d_{TV}}(\mathcal{N}(\mu, \mathbb{I}), \mathcal{N}(\widehat{\mu}, \mathbb{I})) \leq \alpha$.*

2.2 General Gaussians with $d+1$ Public Samples

In this section, we discuss our main result for privately estimating Gaussians with public data.

Theorem 2.4 (Private Gaussian Estimation with Public Data)**.** *For all $\alpha, \beta, \varepsilon, \rho > 0$, there exist ρ-zCDP and ε-DP algorithms that take $d+1$ public samples $\widetilde{X}_1, \ldots, \widetilde{X}_{d+1}$ and n private samples $X_1, \ldots, X_n$ from unknown, d-dimensional Gaussians $\mathcal{N}(\widetilde{\mu}, \widetilde{\Sigma})$ and $\mathcal{N}(\mu, \Sigma)$, respectively, with $0 \leq \mathrm{d_{TV}}(\mathcal{N}(\mu, \Sigma), \mathcal{N}(\widetilde{\mu}, \widetilde{\Sigma})) \leq \gamma < 1$, are private with respect to the private samples, and return $\widehat{\mu}_X \in \mathbb{R}^d$ and $\widehat{\Sigma}_X \in \mathbb{R}^{d\times d}$, such that $\mathrm{d_{TV}}(\mathcal{N}(\widehat{\mu}_X, \widehat{\Sigma}_X), \mathcal{N}(\mu, \Sigma)) \leq \alpha$ with probability at least $1-\beta$, as long as the following bounds on n hold.*

1. *For ρ-zCDP (with computational efficiency) via [KLSU19] and $d+1$ public samples,*

$$n = O\left(\frac{d^2 + \log\left(\frac{1}{\beta}\right)}{\alpha^2} + \frac{d^2 \cdot \operatorname{polylog}\left(\frac{d}{\alpha\beta\rho}\right) + d\log\left(\frac{d}{\alpha\beta\rho(1-\gamma)}\right)}{\alpha\sqrt{\rho}} + \frac{d^{1.5} \cdot \operatorname{polylog}\left(\frac{d}{\beta\rho(1-\gamma)}\right)}{\sqrt{\rho}}\right).$$

2. *For ε-DP (without computational efficiency) via [BKSW19] and $d+1$ public samples,*

$$n = O\left(\frac{d^2 + \log\left(\frac{1}{\beta}\right)}{\alpha^2} + \frac{d^2 \log\left(\frac{d}{\alpha\beta(1-\gamma)}\right)}{\alpha\varepsilon}\right).$$

For simplicity, the rest of this section examines the $\gamma = 0$ case, that is, when the public data and the private data distributions are the same. Later, we address what modifications are needed for the $\gamma > 0$ case.

Public Data Preconditioning. We extend the same high-level idea from the identity covariance case. We use public data to do coarse estimation, and then use that coarse estimate to transform the private data, reducing to the bounded case that can be solved using existing private algorithms (Lemma A.22). Specifically, we use the $d+1$ public samples for "public data preconditioning".

Algorithm 1: Public Data Preconditioner $\mathrm{PubPreconditioner}_\beta(\widetilde{X})$

Input: Public samples $\widetilde{X} = (\widetilde{X}_1, \ldots, \widetilde{X}_{d+1})$. Failure probability $\beta > 0$.
Output: $\widehat{\mu} \in \mathbb{R}^d$, $\widehat{\Sigma} \in \mathbb{R}^{d\times d}$, $L \in \mathbb{R}$, $U \in \mathbb{R}$.

`// Compute the empirical mean and covariance of` $\widetilde{X}$.

$$\widehat{\mu} \leftarrow \frac{1}{d+1}\sum_{i=1}^{d+1} \widetilde{X}_i \quad \text{and} \quad \widehat{\Sigma} \leftarrow \frac{1}{d}\sum_{i=1}^{d+1} (\widetilde{X}_i - \widehat{\mu})(\widetilde{X}_i - \widehat{\mu})^T$$

`// Compute` L `and` U.

$$L \leftarrow \frac{d}{4d + 4\sqrt{2d\log\frac{3}{\beta}} + 2\log\frac{3}{\beta}} \quad \text{and} \quad U \leftarrow \frac{9d^2}{\beta^2}$$

Return $(\widehat{\mu}, \widehat{\Sigma}, L, U)$.

Public-Private Gaussian Estimator. We obtain a public-private Gaussian estimator via the following process. (1) We use the preconditioning parameters output by Algorithm 1 to recenter, then rescale our private data $X_1, \ldots, X_n$. Specifically, the transformed private data, denoted by $Y_1, \ldots, Y_n$, are given by $Y_j := \frac{1}{\sqrt{L}}\widetilde{\Sigma}^{-1/2}(X_j - \widehat{\mu})$. (2) The $Y_1, \ldots, Y_n$ are then fed as input to an existing $\mathrm{DPGaussianEstimator}$ (from Lemma A.22) for bounded Gaussians. It outputs estimates $\widehat{\mu}_Y$ and $\widehat{\Sigma}_Y$. (3) For our final estimates, we output $\widehat{\Sigma}_X := L\widehat{\Sigma}^{1/2}\Sigma_Y\widehat{\Sigma}^{1/2}$ and $\widehat{\mu}_X := \sqrt{L}\widehat{\Sigma}^{1/2}\widehat{\mu}_Y + \widehat{\mu}$. See Algorithm 2 in Appendix B for an exact description.

To establish correctness, we must ensure that, with high probability over the sampling of public data, the parameters of the true distribution underlying $Y_1, \ldots, Y_n$ indeed satisfy tight range bounds required by $\mathrm{DPGaussianEstimator}$ to provide the desired success guarantee with parameter-free private sample complexity. In other words, the mean of the transformed Gaussian must lie in a known $\mathrm{poly}(d, 1/\beta)$ ball, and the condition number of its covariance must be $\mathrm{poly}(d, 1/\beta)$.

Lemma 2.5 (Public Data Preconditioning)**.** *For all $\beta > 0$, there exists an algorithm that takes $d+1$ independent samples $\widetilde{X}_1, \ldots, \widetilde{X}_{d+1}$ from a Gaussian $\mathcal{N}(\mu, \Sigma)$ over $\mathbb{R}^d$, and outputs $\widehat{\mu} \in \mathbb{R}^d$, $\widehat{\Sigma} \in \mathbb{R}^{d\times d}$, $L \in \mathbb{R}$, and $U \in \mathbb{R}$, such that if $\Sigma_Y = \frac{1}{L}\widehat{\Sigma}^{-1/2}\Sigma\widehat{\Sigma}^{-1/2}$ and $\mu_Y = \frac{1}{\sqrt{L}}\widehat{\Sigma}^{-1/2}(\mu - \widehat{\mu})$, then with probability at least $1-\beta$ over the sampling of the data, (1) $\mathbb{I} \preceq \Sigma_Y \preceq \frac{U}{L}\mathbb{I}$, and (2) $\|\mu_Y\| \le \sqrt{\frac{U}{L}} \cdot \sqrt{5\log(3/\beta)}$, where $\frac{U}{L} = O\left(\frac{d^2\log(1/\beta)}{\beta^2}\right)$.*

Proof. We prove the theorem by proving the utility of Algorithm 1. Note that the quantities L, U, as defined in Algorithm 1, satisfy $U/L = O(d^2 \log(1/\beta)/\beta^2)$. To prove part (1), where we wish to bound the eigenvalues of Σ_Y, by the the properties of Loewner ordering, and since $\widehat{\Sigma}, \Sigma$ are symmetric, we have

$$\mathbb{I} \preceq \frac{1}{L}\widehat{\Sigma}^{-1/2}\Sigma\widehat{\Sigma}^{-1/2} \preceq \frac{U}{L}\mathbb{I} \iff L\widehat{\Sigma} \preceq \Sigma \preceq U\widehat{\Sigma} \iff L\Sigma^{-1/2}\widehat{\Sigma}\Sigma^{-1/2} \preceq \mathbb{I} \preceq U\Sigma^{-1/2}\widehat{\Sigma}\Sigma^{-1/2}.$$

Therefore, it is sufficient to prove the final inequality. Let $\widehat{\Sigma}_Z := \Sigma^{-1/2}\widehat{\Sigma}\Sigma^{-1/2}$. Then

$$\widehat{\Sigma}_Z = \frac{1}{d}\sum_{i=1}^{d+1}(\Sigma^{-1/2}(\widetilde{X}_i - \widehat{\mu}))(\Sigma^{-1/2}(\widetilde{X}_i - \widehat{\mu}))^T = \frac{1}{d}\sum_{i=1}^{d+1}(Z_i - \widehat{\mu}_Z)(Z_i - \widehat{\mu}_Z)^T$$

where for each $i \in [d+1]$, $Z_i := \Sigma^{-1/2}(X_i - \mu) \sim N(0, \mathbb{I})$ independently and $\widehat{\mu}_Z := \Sigma^{-1/2}(\widehat{\mu} - \mu) = \frac{1}{d+1}\sum_{i=1}^{d+1} Z_i$. Therefore, it suffices to show $L\widehat{\Sigma}_Z \preceq \mathbb{I}$ and $\mathbb{I} \preceq U\widehat{\Sigma}_Z$. For $i \in [d]$, let $W_i \sim \mathcal{N}(0, \mathbb{I})$ independently. Then Cochran's theorem (see Fact B.4) says that $\widehat{\Sigma}_Z$ is *identically distributed* to $\frac{1}{d}\sum_{i=1}^{d} W_i W_i^T$. From here, we can apply the bounds from Fact B.3 by noting that $\sigma_d(\widehat{\Sigma}_Z) \sim \frac{1}{d}\sigma_d\left([W_1, \ldots, W_d]\right)^2$ and $\sigma_1(\widehat{\Sigma}_Z) \sim \frac{1}{d}\sigma_1\left([W_1, \ldots, W_d]\right)^2$.

With probability $\geq 1 - \beta/3$, $\frac{1}{d}\sigma_d\left([W_1, \ldots, W_d]\right)^2 > \frac{(\beta/3)^2}{d^2} \implies U\widehat{\Sigma}_Z \succeq \mathbb{I}$.

With probability $\geq 1 - \beta/3$, $\frac{1}{d}\sigma_1\left([W_1, \ldots, W_d]\right)^2 < \frac{\left(2\sqrt{d} + \sqrt{2\log(3/\beta)}\right)^2}{d} \implies L\widehat{\Sigma}_Z \preceq \mathbb{I}$.

Taking the union bound, (1) holds with probability at least $1 - \frac{2\beta}{3}$. Next, we prove the bound on μ_Y in (2). We have

$$\mu_Y = \frac{1}{\sqrt{L}}\widehat{\Sigma}^{-1/2}(\mu - \widehat{\mu}) = \frac{1}{\sqrt{L}}\widehat{\Sigma}^{-1/2}\left(\mu - \frac{1}{d+1}\sum_{i=1}^{d+1}(\Sigma^{1/2}Z_i + \mu)\right) = -\frac{1}{\sqrt{L}}\widehat{\Sigma}^{-1/2}\Sigma^{1/2}\widehat{\mu}_Z.$$

Since $\widehat{\mu}_Z$ is identically distributed to $\frac{1}{\sqrt{d+1}}Z_1$, by concentration of the norm for multivariate Gaussians (Lemma A.3), we have that with probability $\geq 1 - \beta/3$,

$$\|\widehat{\mu}_Z\|_2 \leq \sqrt{\frac{d + 2\sqrt{d\log(3/\beta)} + 2\log(3/\beta)}{d+1}}$$

From (1), we know that $\left\|\frac{1}{L}\widehat{\Sigma}^{-1/2}\Sigma\widehat{\Sigma}^{-1/2}\right\|_2 \leq \frac{U}{L}$. Hence, $\left\|-\frac{1}{\sqrt{L}}\widehat{\Sigma}^{-1/2}\Sigma^{1/2}\right\|_2 \leq \sqrt{U/L}$. This implies that

$$\|\mu_Y\|_2 \leq \left\|-\frac{1}{\sqrt{L}}\widehat{\Sigma}^{-1/2}\Sigma^{1/2}\right\|_2 \cdot \|\widehat{\mu}_Z\|_2 \leq \sqrt{\frac{U}{L}} \cdot \sqrt{\frac{d + 2\sqrt{d\log(3/\beta)} + 2\log(3/\beta)}{d+1}}$$

which implies (2). Applying the union bound again, we have the claim. □

Hence the process described above is indeed a $(d+1)$-public-sample private algorithm satisfying privacy with respect to private data $X_1, \ldots, X_n$ and the required accuracy guarantees. This follows from the guarantees of public data preconditioning (as stated in the aforementioned Lemma 2.5) combined with the guarantees of existing DPGaussianEstimator (stated in Lemma A.24). See Theorem B.5 for full details. Again, we note that our sample complexity no longer depends on the *a priori* bounds on the mean and the covariance of the unknown private data distribution.

Different Public and Private Distributions. Thus far, we have only considered the case where the public and the private data come from the same distribution. In the more general case where the public data $\widetilde{X}_1, \ldots, \widetilde{X}_{d+1}$ comes from a Gaussian $\widetilde{D}$ within TV distance $\gamma < 1$ of the private data distribution D, it can be shown that Algorithm 1 (public data preconditioning) with a slight modification will work similarly to map our unbounded problem onto a bounded one. In particular, outputting $U_\gamma = 4U/(1-\gamma)^4$ and $L_\gamma = (1-\gamma)^4 L/4$ instead of U and L leads to covariance and norm bounds $K = O(d^2\log(1/\beta)/\beta^2(1-\gamma)^8)$ and $R = O(d\log(1/\beta)/\beta(1-\gamma)^{4.5})$ on the

underlying parameters of the transformed private data. This result is the consequence of a technical Lemma that translates the TV bound γ into a bound on Gaussian parameters. Full details are in Appendix B.2.

With modified Algorithm 1, we can perform the same process of: (1) transforming private data with our public data preconditioner; (2) running an existing private algorithm requiring bounded range parameters; and (3) undoing the transformation. The guarantees of such a process are stated in Theorem 2.4. We pick up an extra $\mathrm{polylog}(1/(1-\gamma))$ factor in our sample complexity.

3 Estimating Gaussian Mixtures

The general process of estimating the parameters of Gaussian mixtures involves separating the mixture components by using techniques like PCA, followed by their isolation, and finally by individual component estimation. In other words, the components are first projected onto a low-dimensional subspace of $\mathbb{R}^d$, which separates the large mixture components from the rest, given that the separation in the original space is enough. The subspace is computed with respect to the available data, that is, it is chosen to be the top k subspace of the data matrix, which preserves the original distances between the individual components, but shrinks down each component so that the intra-component distances are small but inter-component distances are large still. This allows us to separate a group of components from the rest. Repeating this process on each group, we can narrow down to a single Gaussian in each group, and estimate its parameters.

This has been the high-level idea in a lot of prior work, both non-private [VW02; AM05] and private [KSSU19]. In our setting, we utilise the public data available in different settings at various points to establish these tasks, while having a low cost in the sample complexity of the private data. Given a lower bound on the mixing weights ($w_{\min}$), we consider two settings in terms of the availability of the public data: (1) when there is very little available – $\widetilde{O}(1/w_{\min})$; and (2) when there is a lot more available – $\widetilde{O}(d/w_{\min})$. The key challenge, as indicated, is accurate clustering of data. Due to space constraints, the complete technical details for this section are deferred to Appendix C.

3.1 Public Data Sample Complexity: $\widetilde{O}(1/w_{\min})$

Clustering. This is the more challenging setting of the two because there is not enough public data for techniques such as PCA to give accurate results. We therefore have to rely on the private data for these tasks, and use their private counterparts to get the desired results. The entire procedure is described in detail in Algorithm 6, and its corresponding result in Theorem C.21.

1. Superclustering to reduce sensitivity for Private PCA: The main goal, as in most private estimation tasks, is to reduce the amount of noise added for privacy in order to get better accuracy. This boils down to limiting the sensitivity of the target empirical function of the data. For this, we would like to make sure that the range of the data (to which, we scale the noise) is very tight – in our context, it means that we would like to have a ball that contains a group of mixture components, which are very close to one another (a "supercluster"), having close to the smallest possible radius. We can limit the data to within this ball, and the sensitivity for the next step would be reduced. To get such a ball, we use the public data, and come up with a "superclustering" algorithm (Algorithm 3, Theorem C.14), which does exactly that. This algorithm essentially relies on the concentration properties of high-dimensional Gaussians – it finds the radius of the largest Gaussian and a point (as the centre) within that Gaussian, and additively grows the ball until it stops finding more points. This gives a ball whose radius is an $O(k)$ approximation to the radius of the largest Gaussian. We use this ball to isolate the same components in the private data, and perform private PCA to separate the components within. Getting this tight ball is crucial to reduce the cost of private PCA in terms of the sample complexity compared to the prior work of [KSSU19].
2. Private PCA: We use the well-known private PCA algorithm (Algorithm 4, Corollary C.18) that has been previously analysed in [DTTZ14; KSSU19] for this.
3. Partitioning in low dimensions: The next task is to partition the components in the low-dimensional subspace in a way that there is no overlap. For this, we use an algorithm (Algorithm 5, Theorem C.20) that is similar to the "terrific ball" algorithm of [KSSU19],

and relies upon the accuracy guarantees of the private PCA step. It uses both public and private data, and looks for a terrific ball within them. If there is one, indicating that there is more than one component in that subset of the datasets, it partitions the datasets (and the components), and queues the partitions for further work. Otherwise, it just adds that subset of the private dataset to the set of discovered clusters.

4. The whole process is repeated on all the remaining partitions.

Here, we provide some intuition about the clustering process. To explain the implication of step (1), we state an informal version of Theorem C.14 only for the first application of Algorithm 3 when it is being run initially on the whole public dataset (of size m).

Theorem 3.1. *There exists an algorithm, which if given $m \geq O\left(\frac{\ln(k/\beta)}{w_{\min}}\right)$ samples from a Gaussian mixture $\mathcal{D}$, it outputs $c \in \mathbb{R}^d$ and $R \in \mathbb{R}$, such that with probability $\geq 1-\beta$: (1) $B_R(c)$ is non-empty and contains either $\geq 1-\beta/k$ probability mass of a Gaussian component or $< \beta/k$; and (2) R is an $O(k)$ multiplicative approximation of the radius of the largest Gaussian within $B_R(c)$.*

The above result essentially says that a supercluster is captured by this algorithm in a tight ball. Because with high probability, the points from high-dimensional Gaussians are highly concentrated, $B_R(c)$ would capture all points from both public and private datasets only corresponding to the captured components. Note that this process is performed entirely using the public dataset.

In step (2), where we perform private PCA, the radius of the ball we're limiting the points to ($B_R(c)$) is small, that is, R is an $O(k)$ approximation of the radius of the largest mixture component contained within it. Therefore, while computing the noisy empirical covariance, we add a relatively small amount of noise that preserves the signal of the points from that Gaussian. This ensures that projection matrix we get preserves the inter-Gaussian distances appropriately, but significantly reduces the intra-Gaussian distances. In other words, it preserves the distance between the mean of that largest Gaussian from the means of the rest of Gaussians within $B_R(c)$, while shrinking down all the Gaussians. This helps to isolate at least one component in step (3).

Step (3) relies on the following observation about Gaussians, along with the result from step (2). In the case when there are at least two components within $B_R(c)$, step (2) ensures that in the low-dimensional subspace, there is at least one Gaussian that is far from all the others. Thus, there must be a ball that contains a large number of points, such that expanding it by a constant factor would not give us any new points (because of the available empty space), and the complement of that expanded ball would contain a large number of points. On the other hand, if there is only one component within $B_R(c)$, then projecting it to a low-dimensional subspace would never give us such a ball. Using this observation, Algorithm 5 uses both public and private datasets to determine whether there are one or more components in $B_R(c)$, and in the first case, returns a ball that partitions the datasets in a non-trivial way that no component has points in both subsets (a "clean partition"), or in the second case, returns $\perp$. The number of public samples is enough, such that in the first case, querying the projected public dataset for the existence of such a ball would give us one, but not enough to correctly reject in the second case, which is why we use the much larger private dataset to confirm whether the positive answer we get by querying the public dataset is correct or not.

In the end, the idea of the clustering algorithm is that in each iteration, it would either isolate both public and private points from a component or partition the datasets further in a clean way. In either case, we move one step closer towards obtaining k clean subsets of the two datasets.

Component Estimation. The final task is to estimate the parameters of the k isolated components (Algorithm 7, Theorem C.22). In the (ε, δ)-DP setting (Theorem C.23), we simply apply the private Gaussian learner of [AL21] to each component, which enables us to learn with no dependence in the sample complexity on the range parameters of the Gaussian itself (among other available choices, were the (ε, δ)-DP learners from [KMSSU21; KMV21; TCKMS21], but this one had the joint best sample complexity with [TCKMS21]). In the ρ-zCDP setting (Theorem C.24), we use the Gaussian learner from [KLSU19], instead. Note that in this setting, the private sample complexity does depend on the range parameters of the Gaussians, themselves.

3.2 Public Data Sample Complexity: $\widetilde{O}(d/w_{\min})$

Clustering. In the second setting, clustering is easier in terms of the tasks that need to be done privately – it can be done entirely by using the public dataset itself (Algorithm 9, Theorem C.26). This time, the number of public samples is enough to be able to accurately do PCA on the public data, hence, perform clustering using just the public data itself. This improves the private data sample complexity by removing the term due to private PCA altogether.

1. No supercustering involved: The first change in the clustering algorithm is eliminating the superclustering step.
2. PCA: We essentially get the projection matrix for the top k subspace of the data using non-private PCA, and work within that subspace using the public data to further partition the two datasets (and the components).
3. Partitioning in low dimensions: The algorithm for partitioning in the low-dimensional subspace (Algorithm 8, Theorem C.25) is similar to the private partitioner used in the previous case, except that there is no need for privacy in this case.
4. The whole process is repeated on all the remaining partitions.

Component Estimation. Once the components are separated, as before, we individually estimate them using the partitions of the private dataset privately (Algorithm 10, Theorem C.27). In the (ε, δ)-DP setting (Theorem C.28), we again use the learner from [AL21], which ensures parameter-free private data sample complexity in this process. In the ρ-zCDP setting though (Theorem C.29), we use our own zCDP learner from this text (Theorem 1.1), which uses the public data, as well, to ensure parameter-free private data sample complexity for this process.

4 Conclusion

We investigate the task of private estimation with access to a small amount of public data. Availability of some public data is a reasonable assumption in practice, therefore, we demonstrate ways to take advantage of it. We show that significant improvements can be made in terms of the private sample complexity for the tasks of estimating multivariate Gaussians and Gaussian mixtures, given a small amount of public data. For the first problem, we show that even in the case where the public data distribution differs from the private data distribution significantly, we can still estimate the private data distribution using a small number of public samples, and improve the private sample complexity by removing any dependence on the range parameters of the private data distribution, which is not possible without any public data. For the problem of estimating Gaussian mixtures under privacy constraints, our new algorithms take advantage of public data to obtain sample complexity bounds, which are significantly better than those of the best known private algorithms for this problem.

Limitations and Open Problems. One question in the context of Gaussian estimation is whether we could obtain computationally efficient, pure DP algorithms that use public data, and match the best known private sample complexity for the problem. Another question is whether we could polynomially improve the private sample complexity, given a much larger, but $o(d^2)$, number of public samples. A natural question to ask is whether our results for Gaussian estimation could be extended to mean and covariance estimation of heavy-tailed distributions. For estimation of Gaussian mixtures, our work focuses on the case where the public data and the private data distributions are the same, so studying this problem for the case where they are different would be an interesting challenge. Another related question is understanding other instances of clustering problems, where the results of the existing private algorithms could be improved using some additional public data.

Acknowledgments and Disclosure of Funding

AB, GK, and VS are supported by an NSERC Discovery Grant, a University of Waterloo Startup Grant, an unrestricted gift from Google, and an unrestricted gift from Apple. AB is supported by a Vector Scholarship in Artificial Intelligence.

References

[AAK21] 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. JMLR, Inc., 2021, pp. 185–216 (cit. on p. 4).

[ABM19] Noga Alon, Raef Bassily, and Shay Moran. "Limits of Private Learning with Access to Public Data". In: *Advances in Neural Information Processing Systems 32*. NeurIPS '19. Curran Associates, Inc., 2019, pp. 10342–10352 (cit. on p. 4).

[ACGMMTZ16] Martin 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 Conference on Computer and Communications Security*. CCS '16. New York, NY, USA: ACM, 2016, pp. 308–318 (cit. on p. 4).

[ADK19] Brendan Avent, Yatharth Dubey, and Aleksandra Korolova. "The Power of the Hybrid Model for Mean Estimation". In: *Proceedings on Privacy Enhancing Technologies* 2020.4 (2019), pp. 48–68 (cit. on p. 4).

[AGMRSSSTT21] Ehsan Amid, Arun Ganesh, Rajiv Mathews, Swaroop Ramaswamy, Shuang Song, Thomas Steinke, Vinith M Suriyakumar, Om Thakkar, and Abhradeep Thakurta. "Public Data-Assisted Mirror Descent for Private Model Training". In: *arXiv preprint arXiv:2112.00193* (2021) (cit. on p. 4).

[AK01] Sanjeev Arora and Ravi Kannan. "Learning mixtures of arbitrary Gaussians". In: *Proceedings of the 33rd Annual ACM Symposium on the Theory of Computing*. STOC '01. New York, NY, USA: ACM, 2001, pp. 247–257 (cit. on p. 16).

[AKZHL17] Brendan Avent, Aleksandra Korolova, David Zeber, Torgeir Hovden, and Benjamin Livshits. "BLENDER: Enabling Local Search with a Hybrid Differential Privacy Model". In: *26th USENIX Security Symposium*. USENIX Security '17. USENIX Association, 2017, pp. 747–764 (cit. on p. 4).

[AL21] Hassan Ashtiani and Christopher Liaw. "Private and polynomial time algorithms for learning Gaussians and beyond". In: *arXiv preprint arXiv:2111.11320* (2021) (cit. on pp. 3, 4, 9, 10, 19).

[AM05] Dimitris Achlioptas and Frank McSherry. "On Spectral Learning of Mixtures of Distributions". In: *Proceedings of the 18th Annual Conference on Learning Theory*. COLT '05. Springer, 2005, pp. 458–469 (cit. on pp. 8, 27).

[BCMNUW20] Raef Bassily, Albert Cheu, Shay Moran, Aleksandar Nikolov, Jonathan Ullman, and Steven Wu. "Private Query Release Assisted by Public Data". In: *Proceedings of the 37th International Conference on Machine Learning*. ICML '20. JMLR, Inc., 2020, pp. 695–703 (cit. on pp. 2, 4).

[BDKU20] 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. Curran Associates, Inc., 2020, pp. 14475–14485 (cit. on pp. 48, 49).

[BKSW19] Mark Bun, Gautam Kamath, Thomas Steinke, and Zhiwei Steven Wu. "Private Hypothesis Selection". In: *Advances in Neural Information Processing Systems 32*. NeurIPS '19. Curran Associates, Inc., 2019, pp. 156–167 (cit. on pp. 6, 19, 20, 22, 27).

[BMN20] Raef Bassily, Shay Moran, and Anupama Nandi. "Learning from Mixtures of Private and Public Populations". In: *Advances in Neural Information Processing Systems 33*. NeurIPS '20. Curran Associates, Inc., 2020, pp. 2947–2957 (cit. on p. 4).

[BNS16a] Amos Beimel, Kobbi Nissim, and Uri Stemmer. "Private Learning and Sanitization: Pure vs. Approximate Differential Privacy". In: *Theory of Computing* 12.1 (2016), pp. 1–61 (cit. on pp. 2, 4).

[BNS16b] 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. New York, NY, USA: ACM, 2016, pp. 369–380 (cit. on p. 4).

[BS16] 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. Berlin, Heidelberg: Springer, 2016, pp. 635–658 (cit. on pp. 17, 18).

[BTGT18] Raef Bassily, Om Thakkar, and Abhradeep Guha Thakurta. "Model-Agnostic Private Learning". In: *Advances in Neural Information Processing Systems 31*. NeurIPS '18. Curran Associates, Inc., 2018, pp. 7102–7112 (cit. on p. 4).

[DKKLMS16] 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. Washington, DC, USA: IEEE Computer Society, 2016, pp. 655–664 (cit. on p. 15).

[DMNS06] 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. Berlin, Heidelberg: Springer, 2006, pp. 265–284 (cit. on pp. 1, 17, 18).

[DRV10] 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. Washington, DC, USA: IEEE Computer Society, 2010, pp. 51–60 (cit. on p. 18).

[DTTZ14] 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. New York, NY, USA: ACM, 2014, pp. 11–20 (cit. on pp. 8, 33).

[HKM22] Samuel B Hopkins, Gautam Kamath, and Mahbod Majid. "Efficient Mean Estimation with Pure Differential Privacy via a Sum-of-Squares Exponential Mechanism". In: STOC '22 (2022) (cit. on p. 20).

[HW71] David Lee Hanson and Farroll Tim Wright. "A Bound on Tail Probabilities for Quadratic Forms in Independent Random Variables". In: *The Annals of Mathematical Statistics* 42.3 (1971), pp. 1079–1083 (cit. on p. 15).

[JE13] Zhanglong Ji and Charles Elkan. "Differential Privacy Based on Importance Weighting". In: *Machine Learning* 93.1 (2013), pp. 163–183 (cit. on p. 4).

[KKMN09] 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. New York, NY, USA: ACM, 2009, pp. 171–180 (cit. on p. 4).

[KLSU19] 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. 2019, pp. 1853–1902 (cit. on pp. 4, 5, 9, 19, 22, 27).

[KMSSU21] Gautam Kamath, Argyris Mouzakis, Vikrant Singhal, Thomas Steinke, and Jonathan Ullman. *A Private and Computationally-Efficient Estimator for Unbounded Gaussians*. 2021. URL: https://arxiv.org/abs/2111.04609 (cit. on pp. 4, 9).

[KMV21] Pravesh K Kothari, Pasin Manurangsi, and Ameya Velingker. "Private Robust Estimation by Stabilizing Convex Relaxations". In: *arXiv preprint arXiv:2112.03548* (2021) (cit. on pp. 4, 9).

[KRRT21] Peter Kairouz, Mónica Ribero, Keith Rush, and Abhradeep Thakurta. "(Nearly) Dimension Independent Private ERM with AdaGrad Rates via Publicly Estimated Subspaces". In: *Proceedings of the 34th Annual Conference on Learning Theory*. COLT '21. 2021, pp. 2717–2746 (cit. on p. 4).

[KSSU19] 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. Curran Associates, Inc., 2019, pp. 168–180 (cit. on pp. 3, 8, 17, 27, 34).

[KV18] 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. Dagstuhl, Germany: Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2018, 44:1–44:9 (cit. on p. 4).

[LKO21] Xiyang Liu, Weihao Kong, and Sewoong Oh. "Differential privacy and robust statistics in high dimensions". In: *arXiv preprint arXiv:2111.06578* (2021) (cit. on p. 4).

[LTLH22] Xuechen Li, Florian Tramèr, Percy Liang, and Tatsunori Hashimoto. "Large Language Models Can Be Strong Differentially Private Learners". In: *Proceedings of the 10th International Conference on Learning Representations*. ICLR '22. 2022 (cit. on p. 4).

[LVSUW21] Terrance Liu, Giuseppe Vietri, Thomas Steinke, Jonathan Ullman, and Steven Wu. "Leveraging Public Data for Practical Private Query Release". In: *Proceedings of the 38th International Conference on Machine Learning*. ICML '21. JMLR, Inc., 2021, pp. 6968–6977 (cit. on p. 4).

[LWAFF21] Zelun Luo, Daniel J Wu, Ehsan Adeli, and Li Fei-Fei. "Scalable Differential Privacy with Sparse Network Finetuning". In: *Proceedings of the 2021 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. CVPR '21. Washington, DC, USA: IEEE Computer Society, 2021, pp. 5059–5068 (cit. on p. 4).

[NB20] Anupama Nandi and Raef Bassily. "Privately Answering Classification Queries in the Agnostic PAC Model". In: *Proceedings of the 31st International Conference on Algorithmic Learning Theory*. ALT '20. JMLR, Inc., 2020, pp. 687–703 (cit. on p. 4).

[PAEGT17] Nicolas Papernot, Martín Abadi, Ulfar Erlingsson, Ian Goodfellow, and Kunal Talwar. "Semi-supervised knowledge transfer for deep learning from private training data". In: *Proceedings of the 5th International Conference on Learning Representations*. ICLR '17. 2017 (cit. on p. 4).

[PCSTE19] Nicolas Papernot, Steve Chien, Shuang Song, Abhradeep Thakurta, and Ulfar Erlingsson. *Making the Shoe Fit: Architectures, Initializations, and Tuning for Learning with Privacy*. https://openreview.net/forum?id=rJg851rYwH. 2019 (cit. on p. 4).

[PSMRTE18] Nicolas Papernot, Shuang Song, Ilya Mironov, Ananth Raghunathan, Kunal Talwar, and Úlfar Erlingsson. "Scalable Private Learning with PATE". In: *Proceedings of the 6th International Conference on Learning Representations*. ICLR '18. 2018 (cit. on p. 4).

[SST06] Arvind Sankar, Daniel A. Spielman, and Shang-Hua Teng. "Smoothed Analysis of the Condition Numbers and Growth Factors of Matrices". In: *SIAM J. Matrix Anal. Appl.* 28.2 (2006), 446–476 (cit. on p. 22).

[TB21] Florian Tramèr and Dan Boneh. "Differentially Private Learning Needs Better Features (or Much More Data)". In: *Proceedings of the 9th International Conference on Learning Representations*. ICLR '21. 2021 (cit. on p. 4).

[TCKMS21] Eliad Tsfadia, Edith Cohen, Haim Kaplan, Yishay Mansour, and Uri Stemmer. "FriendlyCore: Practical Differentially Private Aggregation". In: *arXiv preprint arXiv:2110.10132* (2021) (cit. on pp. 4, 9).

[VW02] Santosh Vempala and Grant Wang. "A Spectral Algorithm for Learning Mixtures of Distributions". In: *Proceedings of the 43rd Annual IEEE Symposium on Foundations of Computer Science*. FOCS '02. Washington, DC, USA: IEEE Computer Society, 2002, pp. 113–123 (cit. on p. 8).

[Yu+22] Da Yu, Saurabh Naik, Arturs Backurs, Sivakanth Gopi, Huseyin A Inan, Gautam Kamath, Janardhan Kulkarni, Yin Tat Lee, Andre Manoel, Lukas Wutschitz, Sergey Yekhanin, and Huishuai Zhang. "Differentially Private Fine-tuning of Language Models". In: *Proceedings of the 10th International Conference on Learning Representations*. ICLR '22. 2022 (cit. on p. 4).

[YZCL21] Da Yu, Huishuai Zhang, Wei Chen, and Tie-Yan Liu. "Do Not Let Privacy Overbill Utility: Gradient Embedding Perturbation for Private Learning". In: *Proceedings of the 9th International Conference on Learning Representations*. ICLR '21. 2021 (cit. on p. 4).

[ZWB21] Yingxue Zhou, Zhiwei Steven Wu, and Arindam Banerjee. "Bypassing the Ambient Dimension: Private SGD with Gradient Subspace Identification". In: *Proceedings of the 9th International Conference on Learning Representations*. ICLR '21. 2021 (cit. on p. 4).

[ZZ18] Anru R. Zhang and Yuchen Zhou. *On the Non-asymptotic and Sharp Lower Tail Bounds of Random Variables*. 2018. URL: https://arxiv.org/abs/1810.09006 (cit. on p. 15).

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] See Section 4.
   (c) Did you discuss any potential negative societal impacts of your work? [N/A]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes]
   (b) Did you include complete proofs of all theoretical results? [Yes] See Section 2 and Appendices B and C.
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes]
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes]
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes]
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [N/A] No license in the repository (https://github.com/twistedcubic/coin-press), however we received permission from the authors to use their code.
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

A Preliminaries

A.1 Notations

We define a few notations here to be used throughout the paper.

- Private data are denoted by $X = (X_1, \ldots, X_n)$, with each $X_j \in \mathbb{R}^d$. Public data is denoted by $\widetilde{X} = (\widetilde{X}_1, \ldots, \widetilde{X}_m)$, with each $\widetilde{X}_i \in \mathbb{R}^d$.
- We denote a ball of radius $r > 0$ centred around a point $c \in \mathbb{R}^d$ by $B_r(c)$.
- For a set S, we denote its power set by $\mathcal{P}(S)$.

A.2 Useful Concentration Inequalities

We first state a multiplicative Chernoff bound.

Lemma A.1 (Multiplicative Chernoff)**.** *Let $X_1, \ldots, X_m$ be independent Bernoulli random variables, and let X be their sum. If $p = \mathbb{E}(X_i)$, then for $0 \leq \delta_1 \leq 1$ and $\delta_2 \geq 0$,*

$$\mathbb{P}(X \leq (1-\delta_1)pm) \leq e^{-\frac{\delta_1^2 pm}{2}}$$

and

$$\mathbb{P}(X \leq (1+\delta_2)pm) \leq e^{-\frac{\delta_2^2 pm}{2+\delta_2}}.$$

Next, we state Bernstein's inequality.

Lemma A.2 (Bernstein's Inequality)**.** *Let $X_1, \ldots, X_m$ be independent Bernoulli random variables. Let $p = \mathbb{E}(X_i)$. Then for $m \geq \frac{5p}{2\varepsilon^2}\ln(2/\beta)$ and $\varepsilon \leq p/4$,*

$$\mathbb{P}\left(\left|\frac{1}{m}\sum X_i - p\right| \geq \varepsilon\right) \leq 2e^{-\varepsilon^2 m/2(p+\varepsilon)} \leq \beta.$$

Now, we state the Hanson-Wright inequality about quadratic forms.

Lemma A.3 (Hanson-Wright inequality [HW71])**.** *Let $X \sim \mathcal{N}(0, \mathbb{I}_{d\times d})$ and let A be a $d \times d$ matrix. Then for all $t > 0$, the following two bounds hold:*

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

$$\mathbb{P}\left(X^T AX - \mathrm{tr}(A) \leq -2\|A\|_F\sqrt{t}\right) \leq \exp(-t).$$

We mention an inequality that bounds the tails of a one-dimensional Gaussian $\mathcal{N}(\mu, \sigma^2)$.

Lemma A.4 (1-D Gaussian Concentration)**.** *Let $Z \sim \mathcal{N}(\mu, \sigma^2)$. Then,*

$$\mathbb{P}(|Z - \mu| \leq t\sigma) \leq 2e^{-\frac{t^2}{2}}.$$

Next, we state a concentration inequality for 0-mean Laplace random variables.

Lemma A.5 (Laplace Concentration)**.** *Let $Z \sim \mathrm{Lap}(t)$. Then $\mathbb{P}(|Z| > t \cdot \ln(1/\beta)) \leq \beta$.*

We now mention an anti-concentration inequality for weighted χ^2 distributions from [ZZ18]. We adjust the constants appropriately in the following theorem as per the specifications in the aforementioned article.

Lemma A.6 (Theorem 6 from [ZZ18])**.** *Let $Z \sim \mathcal{N}(0, 1)$, $a > 0$, and $Y = aZ$. Then for $\tau \geq a$, we have the following.*

$$\mathbb{P}(Y \geq a + \tau) \geq 0.06e^{-\frac{3\tau}{2a}}$$

The following are standard concentration results for the empirical mean and covariance of a set of Gaussian vectors (see, e.g., [DKKLMS16]).

Lemma A.7. *Let $X_1, \ldots, X_n$ be i.i.d. samples from $\mathcal{N}(0, \mathbb{I}_{d\times d})$. Then we have that*

$$\mathbb{P}\left(\left\|\frac{1}{n}\sum_{i\in[n]} X_i\right\|_2 \geq t\right) \leq 4\exp(c_1 d - c_2 n t^2);$$

$$\mathbb{P}\left(\left\|\frac{1}{n}\sum_{i\in[n]} X_i X_i^T - I\right\|_2 \geq t\right) \leq 4\exp(c_3 d - c_4 n \min(t, t^2)),$$

where $c_1, c_2, c_3, c_4 > 0$ are some absolute constants.

A.3 Gaussian Mixtures

Here, we provide preliminaries for the problem of parameter estimation of mixtures of Gaussians. Let Sym_d^+ denote set of all $d \times d$, symmetric, and positive semidefinite matrices. Let $\mathcal{G}(d) = \{\mathcal{N}(\mu, \Sigma) : \mu \in \mathbb{R}^d, \Sigma \in \mathsf{Sym}_d^+\}$ be the family of d-dimensional Gaussians. We can now define the class $\mathcal{G}(d, k)$ of mixtures of Gaussians as follows.

Definition A.8 (Gaussian Mixtures)**.** The class of *Gaussian k-mixtures in $\mathbb{R}^d$* is

$$\mathcal{G}(d, k) \coloneqq \left\{\sum_{i=1}^k w_i G_i : G_1, \ldots, G_k \in \mathcal{G}(d), w_1, \ldots, w_k > 0, \sum_{i=1}^k w_i = 1\right\}.$$

We can specify a Gaussian mixture by a set of k tuples as: $\{(\mu_1, \Sigma_1, w_1), \ldots, (\mu_k, \Sigma_k, w_k)\}$, where each tuple represents the mean, covariance matrix, and mixing weight of one of its components. Additionally, for each i, we refer to $\sigma_i^2 = \|\Sigma_i\|_2$ as the maximum directional variance of component i.

We assume that the mixing weight of each component is lower bounded by $w_{\min}$. We impose a separation condition for mixtures of Gaussians to be able to learn them.

Definition A.9 (Separated Mixtures)**.** For $s > 0$, a Gaussian mixtures $\mathcal{D} \in \mathcal{G}(d, k)$ is *s-separated* if

$$\forall 1 \leq i < j \leq k, \quad \|\mu_i - \mu_j\|_2 \geq \left(s + \frac{10}{\sqrt{w_i}} + \frac{10}{\sqrt{w_j}}\right) \cdot \max\{\sigma_i, \sigma_j\}.$$

We denote the family of separated Gaussian mixtures by $\mathcal{G}(d, k, s)$.

Now, we define what it means to "learn" a Gaussian mixture in our setting.

Definition A.10 ((α, β)-Learning)**.** Let $\mathcal{D} \in \mathcal{G}(d, k)$ be parameterized by $\{(\mu_1, \Sigma_1, w_1), \ldots, (\mu_k, \Sigma_k, w_k)\}$. We say that an algorithm *(α, β)-learns* $\mathcal{D}$, if on being given sample-access to $\mathcal{D}$, it outputs with probablity at least $1 - \beta$ a distribution $\widehat{\mathcal{D}} \in \mathcal{G}(d, k)$ parameterised by $\{(\widehat{\mu}_1, \widehat{\Sigma}_1, \widehat{w}_1), \ldots, (\widehat{\mu}_k, \widehat{\Sigma}_k, \widehat{w}_k)\}$, such that there exists a permutation $\pi : [k] \to [k]$, for which the following conditions hold.

1. For all $1 \leq i \leq k$, $\mathrm{d_{TV}}(\mathcal{N}(\mu_i, \Sigma_i), \mathcal{N}(\widehat{\mu}_{\pi(i)}, \widehat{\Sigma}_{\pi(i)})) \leq O(\alpha)$.
2. For all $1 \leq i \leq k$, $\left|w_i - \widehat{w}_{\pi(i)}\right| \leq O\left(\frac{\alpha}{k}\right)$.

Note that the above two conditions together imply that $\mathrm{d_{TV}}(\mathcal{D}, \widehat{\mathcal{D}}) \leq \alpha$.

Finally, we define the "median radius" of a Gaussian.

Definition A.11. Let $G \coloneqq \mathcal{N}(\mu, \Sigma)$ be a d-dimensional Gaussian, and $R > 0$. Then R is called the median radius of G, if the G-measure of $B_R(\mu)$ is exactly $1/2$.

The following is an anti-concentration lemma from [AK01] about Gaussians. It lower bounds the distance of a sample from a Gaussian from any arbitrary point in space.

Lemma A.12 (Lemma 6 from [AK01])**.** *Let $G \coloneqq \mathcal{N}(\mu, \Sigma)$ be a Gaussian in $\mathbb{R}^d$ with median radius R, and let $\sigma^2 = \|\Sigma\|$. Suppose $z \in \mathbb{R}^d$ is an arbitrary point, and $x \sim G$. Then for all $t \geq 1$, with probability at least $1 - 2e^{-t}$,*

$$\|x - z\|^2 \geq (\max\{R - t\sigma, 0\})^2 + \|z - \mu\|^2 - 2\sqrt{2t}\sigma\|z - \mu\|.$$

A.4 Robustness of PCA to Noise

Principal Component Analysis (PCA) is one of the main tools in learning mixtures of Gaussians. It is common to project the entire data onto the top-k principal directions (a subspace, which would approximately contain all the means of the k components). Ideally, PCA should eliminate directions that are not so useful, whilst maintaining the distances among the Gaussian components. This would allow us to cluster the data in low dimensions easily based on the most useful directions, while not having to worry about the rest of the directions that do not give us useful information for this process. For the purpose of doing PCA under differential privacy, we would like to have results for PCA when there is noise involved in the process of obtaining DP guarantees.

Let $X \in \mathbb{R}^{n\times d}$ be the dataset obtained from the mixture of Gaussians in question. Suppose $A \in \mathbb{R}^{n\times d}$, such that for each i, A_i is the true mean of the Gaussian from which X_i has been sampled. Let n_j denote the number of points in X belonging to component j. The following lemma gives guarantees for PCA when we have a noisy approximation to the top-k subspace of the dataset.

Lemma A.13 (Lemma 3.1 from [KSSU19])**.** *Let $X \in \mathbb{R}^{n\times d}$ be a collection of n datapoints from k clusters each centered at $\mu_1, \mu_2, ..., \mu_k$. Let $A \in \mathbb{R}^{n\times d}$ be the corresponding matrix of (unknown) centers (for each j we place the center $\mu_{c(j)}$ with $c(j)$ denoting the clustering point X_j belongs to). Let $\Pi_{V_k} \in \mathbb{R}^{d\times d}$ denote the k-PCA projection of X's rows. Let $\Pi_U \in \mathbb{R}^{d\times d}$ be a projection such that for some bound $B \geq 0$ it holds that $\|X^TX - (X\Pi_U)^T(X\Pi_U)\|_2 \leq \|X^TX - (X\Pi_{V_k})^T(X\Pi_{V_k})\|_2 + B$. Denote $\bar{\mu_i}$ as the empirical mean of all points in cluster i and denote $\hat{\mu_i}$ as the projection of the empirical mean $\hat{\mu_i} = \Pi_U\bar{\mu_i}$. Then*

$$\|\bar{\mu_i} - \hat{\mu_i}\|_2 \leq \frac{1}{\sqrt{n_i}}\|X - A\|_2 + \sqrt{\frac{B}{n_i}}$$

The next lemma bounds the singular values of a matrix X that is sampled from a mixture of Gaussians, but centred around A.

Lemma A.14 (Lemma 3.2 from [KSSU19])**.** *Let $X \in \mathbb{R}^{n\times d}$ be a sample from $\mathcal{D} \in \mathcal{G}(d,k)$, and let $A \in \mathbb{R}^{n\times d}$ be the matrix where each row i is the (unknown) mean of the Gaussian from which X_i was sampled. For each i, let σ_i^2 denote the maximum directional variance of component i, and w_i denote its mixing weight. Define $\sigma^2 = \max_i\{\sigma_i^2\}$ and $w_{\min} = \min_i\{w_i\}$. If*

$$n \geq \frac{1}{w_{\min}}\left(\xi_1 d + \xi_2 \log\left(\frac{2k}{\beta}\right)\right),$$

where ξ_1, ξ_2 are universal constants, then with probability at least $1-\beta$,

$$\frac{\sqrt{nw_{\min}}\sigma}{4} \leq \|X - A\|_2 \leq 4\sqrt{n\sum_{i=1}^{k} w_i\sigma_i^2}.$$

A.5 Privacy Preliminaries

We start with different definitions of differential privacy.

Definition A.15 (Differential Privacy (DP) [DMNS06])**.** A randomized algorithm $M : \mathcal{X}^n \to \mathcal{Y}$ satisfies (ε, δ)-differential privacy ((ε, δ)-DP) if for every pair of neighboring datasets $X, X' \in \mathcal{X}^n$ (i.e., datasets that differ in exactly one entry),

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

When $\delta = 0$, we say that M satisfies ε-differential privacy or pure differential privacy.

Definition A.16 (Concentrated Differential Privacy (zCDP) [BS16])**.** A randomized algorithm $M : \mathcal{X}^n \to \mathcal{Y}$ satisfies *ρ-zCDP* if for every pair of neighboring datasets $X, X' \in \mathcal{X}^n$,

$$\forall \alpha \in (1, \infty) \quad D_\alpha\left(M(X)||M(X')\right) \leq \rho\alpha,$$

where $D_\alpha\left(M(X)||M(X')\right)$ is the α-Rényi divergence between $M(X)$ and $M(X')$.[3]

[3] Given two probability distributions P, Q over Ω, $D_\alpha(P\|Q) = \frac{1}{\alpha-1}\log\left(\sum_x P(x)^\alpha Q(x)^{1-\alpha}\right)$.

Note that $(\varepsilon, 0)$-DP implies $\frac{\varepsilon^2}{2}$-zCDP, which implies $(\varepsilon\sqrt{\log(1/\delta)}, \delta)$-DP for every $\delta > 0$ [BS16]. Now, we define the notion of "public-private" algorithms that take public data samples and private data samples as input, and guarantee differential privacy with respect to the private data.

Definition A.17 (Public-Private Algorithms)**.** Let $\widetilde{\mathcal{X}}$ be the domain of public data and $\mathcal{X}$ be the domain of private data. A randomized algorithm $M : \mathcal{X}^n \times \widetilde{\mathcal{X}}^m \to \mathcal{Y}$ taking in public and private data satisfies (ε, δ)-DP (or ρ-zCDP) *with respect to the private data* if for any public dataset $\widetilde{X} \in \widetilde{\mathcal{X}}^m$, the resulting randomized algorithm $M(\widetilde{X}, \cdot) : \mathcal{X}^n \to \mathcal{Y}$ is (ε, δ)-DP (or ρ-zCDP, respectively).

These definitions of DP are closed under post-processing, and can be composed with graceful degradation of the privacy parameters.

Lemma A.18 (Post-Processing [DMNS06; BS16])**.** *If $M : \mathcal{X}^n \to \mathcal{Y}$ is (ε, δ)-DP, and $P : \mathcal{Y} \to \mathcal{Z}$ is any randomized function, then the algorithm $P \circ M$ is (ε, δ)-DP. Similarly if M is ρ-zCDP then the algorithm $P \circ M$ is ρ-zCDP.*

Lemma A.19 (Composition of DP [DMNS06; DRV10; BS16])**.** *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 (ε, δ)-DP for*
$$\varepsilon = \sum_t \varepsilon_t \quad \text{and} \quad \delta = \sum_t \delta_t.$$
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 (ε, δ)-DP for*
$$\varepsilon = \varepsilon_0\sqrt{6T\log(1/\delta_0)} \quad \text{and} \quad \delta = \delta_0 + \sum_t \delta_t$$
3. *If $M_1, \ldots, M_T$ are $\rho_1, \ldots, \rho_T$-zCDP then M is ρ-zCDP for $\rho = \sum_t \rho_t$.*

A.5.1 Known Differentially Private Mechanisms

We state standard results on achieving differential privacy via noise addition proportional to the sensitivity [DMNS06].

Definition A.20 (Sensitivity)**.** Let $f : \mathcal{X}^n \to \mathbb{R}^d$ be a function, its ℓ_1-*sensitivity* and ℓ_2-*sensitivity* are
$$\Delta_{f,1} = \max_{X \sim X' \in \mathcal{X}^n} \|f(X) - f(X')\|_1 \quad \text{and} \quad \Delta_{f,2} = \max_{X \sim X' \in \mathcal{X}^n} \|f(X) - f(X')\|_2,$$
respectively. Here, $X \sim X'$ denotes that X and X' are neighboring datasets (i.e., those that differ in exactly one entry).

For functions with bounded ℓ_1-sensitivity, we can achieve ε-DP by adding noise from a Laplace distribution proportional to ℓ_1-sensitivity. For functions taking values in $\mathbb{R}^d$ for large d it is more useful to add noise from a Gaussian distribution proportional to the ℓ_2-sensitivity, to get (ε, δ)-DP and ρ-zCDP.

Lemma A.21 (Laplace Mechanism)**.** *Let $f : \mathcal{X}^n \to \mathbb{R}^d$ be a function with ℓ_1-sensitivity $\Delta_{f,1}$. Then the Laplace mechanism*
$$M(X) = f(X) + \mathrm{Lap}\left(\frac{\Delta_{f,1}}{\varepsilon}\right)^{\otimes d}$$
satisfies ε-DP.

Lemma A.22 (Gaussian Mechanism)**.** *Let $f : \mathcal{X}^n \to \mathbb{R}^d$ be a function with ℓ_2-sensitivity $\Delta_{f,2}$. Then the Gaussian mechanism*
$$M(X) = f(X) + \mathcal{N}\left(0, \left(\frac{\Delta_{f,2}\sqrt{2\ln(2/\delta)}}{\varepsilon}\right)^2 \cdot \mathbb{I}_{d\times d}\right)$$
satisfies (ε, δ)-DP. Similarly, the Gaussian mechanism
$$M_f(X) = f(X) + \mathcal{N}\left(0, \left(\frac{\Delta_{f,2}}{\sqrt{2\rho}}\right)^2 \cdot \mathbb{I}_{d\times d}\right)$$
satisfies ρ-zCDP.

Next, we mention a very basic pure DP algorithm that computes the cardinality of a dataset as a simple application of Lemma A.21.

Lemma A.23 (PCount)**.** *Let $X = (X_1, \ldots, X_n)$ be a set of points from some data universe χ. Then for all $\varepsilon > 0$ and $0 < \beta < 1$, there exists an ε-DP mechanism ($\mathrm{PCount} : \chi^* \to \mathbb{R}$) that on input X outputs n', such that with probability at least $1 - \beta$, $|n - n'| \leq \frac{\ln(1/\beta)}{\varepsilon}$.*

Proof. The algorithm is just the following. Step 1: sample $\gamma \sim \mathrm{Lap}\left(\frac{1}{\varepsilon}\right)$. Step 2: output $|X| + \gamma$.

Since the sensitivity of $|X|$ is 1, by Lemma A.21, PCount is ε-DP.

Next, by the guarantees of Lemma A.5, $|\gamma| \leq \frac{\ln(1/\beta)}{\varepsilon}$ with probability at least $1 - \beta$. □

Now, we mention three results from prior work for learning high-dimensional Gaussians under approximate DP, zCDP, and pure DP constraints, respectively.

Lemma A.24 (DPGaussianEstimator)**.** *Let $\Sigma \in \mathbb{R}^{d \times d}$ be symmetric and positive-definite, and $\mu \in \mathbb{R}^d$. For all $0 < \alpha, \beta < 1$, if given n_{GE} independent samples from $\mathcal{N}(\mu, \Sigma)$, then the following algorithms output a symmetric and positive-definite $\widehat{\Sigma} \in \mathbb{R}^{d \times d}$, and a vector $\widehat{\mu} \in \mathbb{R}^d$, such that with probability at least $1 - \beta$, $\mathrm{d_{TV}}(\mathcal{N}(\mu, \Sigma), \mathcal{N}(\widehat{\mu}, \widehat{\Sigma})) \leq \alpha$.*

1. *For the (ε, δ)-DP Gaussian learner from [AL21],*

$$n_{GE} = O\left(\frac{d^2 + \log\left(\frac{1}{\beta}\right)}{\alpha^2} + \frac{\left(d^2\sqrt{\log\left(\frac{1}{\delta}\right)} + d\log\left(\frac{1}{\delta}\right)\right) \cdot \mathrm{polylog}\left(d, \frac{1}{\alpha}, \frac{1}{\beta}, \frac{1}{\varepsilon}, \log\left(\frac{1}{\delta}\right)\right)}{\alpha\varepsilon}\right).$$

2. *For the ρ-zCDP Gaussian learner from [KLSU19], for $\mathbb{I} \preceq \Sigma \preceq K\mathbb{I}$ and $\|\mu\| \leq R$ (where $K \geq 1$ and $R > 0$),*

$$n_{GE} = O\left(\frac{d^2 + \log\left(\frac{1}{\beta}\right)}{\alpha^2} + \frac{d^2 \cdot \mathrm{polylog}\left(\frac{d}{\alpha\beta\rho}\right) + d\log\left(\frac{d\log(R)}{\alpha\beta\rho}\right)}{\alpha\sqrt{\rho}} + \frac{d^{1.5}\sqrt{\log(K)} \cdot \mathrm{polylog}\left(\frac{d\log(K)}{\rho\beta}\right) + \sqrt{d\log\left(\frac{Rd}{\beta}\right)}}{\sqrt{\rho}}\right).$$

3. *For the (computationally inefficient) ε-DP Gaussian learner from [BKSW19], for $\mathbb{I} \preceq \Sigma \preceq K\mathbb{I}$ and $\|\mu\| \leq R$ (where $K \geq 1$ and $R > 0$),*

$$n_{GE} = O\left(\frac{d^2 + \log\left(\frac{1}{\beta}\right)}{\alpha^2} + \frac{d\log\left(\frac{dR}{\alpha}\right) + d^2\log\left(\frac{dK}{\alpha}\right) + \log\left(\frac{1}{\beta}\right)}{\alpha\varepsilon}\right).$$

B Estimating Gaussians

Here we present missing details from Section 2.

B.1 Same Public and Private Distributions

B.1.1 Unknown Mean, Identity Covariance with 1 Public Sample

In this setting, we are given a single public sample $\widetilde{X}$, along with private samples $X_1, \ldots, X_n$, where the $\widetilde{X}$ and the X_j are drawn from a d-dimensional Gaussian $\mathcal{N}(\mu, \mathbb{I})$ independently. Recall our observation from Section 2: a single public sample is sufficient to get a "good enough" coarse estimate of μ, allowing us to apply existing private algorithms for a finer estimate. This was captured in Claim 2.1.

We restate existing pure and concentrated DP algorithms for Gaussian mean estimation below.

Lemma B.1 (Known Private Mean Estimators for Identity Covariance Gaussians)**.** *For all $\alpha, \beta, \varepsilon, \rho > 0$, there exist ε-DP and ρ-zCDP algorithms that take $n > 0$ samples from $\mathcal{N}(\mu, \mathbb{I})$ over $\mathbb{R}^d$, where $\|\mu\|_2 \leq R$, and output estimate $\widehat{\mu} \in \mathbb{R}^d$, such that with probability at least $1 - \beta$, $\mathrm{d_{TV}}(\mathcal{N}(\mu, \mathbb{I}), \mathcal{N}(\widehat{\mu}, \mathbb{I})) \leq \alpha$, as long as the following bounds on n hold.*

1. *For the ρ-zCDP "clip-and-noise" algorithm via the Gaussian Mechanism,*

$$n = \widetilde{O}\left(\frac{d + \log(1/\beta)}{\alpha^2} + \frac{\sqrt{d}(R + \sqrt{d})\log(1/\beta)}{\alpha\sqrt{\rho}}\right).$$

2. *For the (inefficient) ε-DP algorithm from [BKSW19],*

$$n = O\left(\frac{d + \log(1/\beta)}{\alpha^2} + \frac{d\log(dR/\alpha) + \log(1/\beta)}{\alpha\varepsilon}\right).$$

3. *For the ε-DP algorithm from [HKM22],*

$$n = \widetilde{O}\left(\frac{d + \log(1/\beta)}{\alpha^2} + \frac{d + \log(1/\beta)}{\alpha^2\varepsilon} + \frac{d\log R + \min\{d, \log R\} \cdot \log(1/\beta)}{\varepsilon}\right).$$

In the above, the "clip-and-noise" mechanism simply clips all the points in the dataset to within radius $\lambda = R + \sqrt{d + 2\sqrt{d\log(n/\beta)} + 2\log(n/\beta)}$ around the origin, and outputs the noisy empirical mean of the clipped points via the Gaussian mechanism. Note that the ℓ_2 sensitivity of the empirical mean is now $\frac{2\lambda}{n}$, therefore, the magnitude of the noise vector added to the empirical mean is at most $\widetilde{O}\left(\frac{\sqrt{d}\lambda}{\sqrt{\rho}n}\right)$ with high probability, which is less than α when $n \geq \widetilde{O}\left(\frac{\sqrt{d}\lambda}{\sqrt{\rho}\alpha}\right)$. This is ρ-zCDP from Lemma A.22.

Also, in part (3) of the above lemma, the third term of the sample complexity arises when the algorithm tries to find a coarse estimate of the mean, that is, when it attempts to get an estimate that lies within $O(\sqrt{d})$ of the true mean (Theorem 5.1 of [HKM22]). If we already have such a coarse estimate with us (which we will), then that term would disappear, and we would simply have to apply Theorem 6.1 from [HKM22] to get the first term.

Claim 2.1 combined with the results above, yields 1-public-sample, private mean estimation algorithms.

Theorem B.2 (Public-Private Gaussian Mean Estimation)**.** *For all $\alpha, \beta, \varepsilon, \rho > 0$, there exist ε-DP and ρ-zCDP algorithms that take $n > 0$ private samples $X = (X_1, \ldots, X_n)$ and 1 public sample $\widetilde{X}$ from $\mathcal{N}(\mu, \mathbb{I})$ over $\mathbb{R}^d$, are private with respect to the private samples X, and output estimate $\widehat{\mu} \in \mathbb{R}^d$, such that with probability at least $1 - \beta$, $\mathrm{d_{TV}}(\mathcal{N}(\mu, \mathbb{I}), \mathcal{N}(\widehat{\mu}, \mathbb{I})) \leq \alpha$, as long as the following bounds on n hold.*

1. *For the ρ-zCDP "clip-and-noise" algorithm via the Gaussian Mechanism and 1 public sample,*

$$n = \widetilde{O}\left(\frac{d + \log\left(\frac{1}{\beta}\right)}{\alpha^2} + \frac{d\log^{1.5}\left(\frac{1}{\beta}\right)}{\alpha\sqrt{\rho}}\right).$$

2. *For the (inefficient) ε-DP algorithm via [BKSW19] and 1 public sample,*

$$n = O\left(\frac{d + \log\left(\frac{1}{\beta}\right)}{\alpha^2} + \frac{d\log\left(\frac{d\log(1/\beta)}{\alpha}\right) + \log\left(\frac{1}{\beta}\right)}{\alpha\varepsilon}\right).$$

3. *For the (efficient) ε-DP algorithm via [HKM22] and 1 public sample,*

$$n = \widetilde{O}\left(\frac{d + \log\left(\frac{1}{\beta}\right)}{\alpha^2} + \frac{d + \log\left(\frac{1}{\beta}\right)}{\alpha^2\varepsilon}\right).$$

Proof. For concreteness, we look at (1), however an analogous argument suffices for (2) and (3). Say we draw our one public sample $\widetilde{X}$, and form a shifted private dataset $Y_1, \ldots, Y_n$, each $Y_j = X_j - \widetilde{X}$. Define $\lambda = \sqrt{d + 2\sqrt{d\log(2/\beta)} + 2\log(2/\beta)}$ and $\mu_Y := \mathbb{E}(Y_1) = \mu - \widetilde{X}$. By Claim 2.1, we have that with probability $\geq 1 - \beta/2$ over the sampling of $\widetilde{X}$, $\|\mu_Y\|_2 \leq \lambda$. Hence, we set $R = \lambda$ and target failure probability $\beta/2$, and run the zCDP mean estimation algorithm from Lemma B.1 on $Y_1, \ldots, Y_n$. Suppose the output of the algorithm is $\widehat{\mu}_Y$. Then we return $\widehat{\mu} = \widehat{\mu}_Y + \widetilde{X}$.

By union bound, with probability $\geq 1 - \beta$, we have that our private data distribution satisfies the range requirements for our private algorithm's guarantees to hold *and* that our private algorithm succeeds. In this case, we have $\|\mu - \widehat{\mu}\|_2 = \|(\mu - \widetilde{X}) - (\widehat{\mu} - \widetilde{X})\|_2 = \|\mu_Y - \widehat{\mu}_Y\|_2$, so $\mathrm{d_{TV}}(\mathcal{N}(\widehat{\mu}, \mathbb{I}), \mathcal{N}(\mu, \mathbb{I})) = \mathrm{d_{TV}}(\mathcal{N}(\widehat{\mu}_Y, \mathbb{I}), \mathcal{N}(\mu_Y, \mathbb{I})) \leq \alpha$. Plugging in $R = \lambda$ and target failure probability $\beta/2$ into the algorithm's sample complexity in Lemma B.1 gives us the desired result. As mentioned earlier, if we are to use part (3) of Lemma B.1, then the third term of the sample complexity would disappear because the public sample already gives us an $O(\sqrt{d})$ estimate of μ.

Notice that $\widehat{\mu}_Y$ is ρ-zCDP with respect to $Y_1, \ldots, Y_n$, which implies that $\widehat{\mu}$ is ρ-zCDP *with respect to private data* $X_1, \ldots, X_n$. To see why, note that for any fixed $\widetilde{X}$, the private algorithm from Lemma B.1 is robust to arbitrary change in Y_j, therefore, to any change in X_j because each X_j maps to exactly one Y_j. By the post-processing guarantee of zCDP (Lemma A.18), $\widehat{\mu}$ is also private with respect to X. □

We remark that the above 1-public-sample private mean estimators can also be used to estimate the mean of a Gaussian with arbitrary known covariance, by reducing to the identity covariance case via rescaling with the known covariance.

B.1.2 Unknown Mean and Covariance with $d + 1$ Public Samples

In this setting, we are given $d + 1$ public samples $\widetilde{X}_1, \ldots, \widetilde{X}_{d+1}$ and n private samples $X_1, \ldots, X_n$, where all the $\widetilde{X}_i$ and X_j are drawn from an unknown, d-dimensional Gaussian $\mathcal{N}(\mu, \Sigma)$ independently. Public data is used to transform private samples via "public data preconditioning" (Algorithm 1, as discussed in Section 2), which reduces the estimation problem to the bounded case which can be solved using the existing private algorithms.

Algorithm 2 gives the precise description of how we can use Algorithm 1 and existing private Gaussian estimators to obtain a public-private Gaussian estimator.

Algorithm 2: Public-Private Gaussian Estimator $\mathrm{PubDPGaussianEstimator}_{\alpha,\beta,\mathrm{PrivParams}}(\widetilde{X}, X)$

Input: Public samples $\widetilde{X} = (\widetilde{X}_1, \ldots, \widetilde{X}_{d+1})$. Private samples $X = (X_1, \ldots, X_n)$. Error tolerance $\alpha > 0$, failure probability $\beta > 0$. Privacy parameters $\mathrm{PrivParams} \subset \mathbb{R}$.

Output: $\widehat{\mu}_X \in \mathbb{R}^d$, $\widehat{\Sigma}_X \in \mathbb{R}^{d \times d}$

`// Precondition the private data using the public dataset.`

$(\widehat{\mu}, \widehat{\Sigma}, L, U) \leftarrow \mathrm{PubPreconditioner}_{\frac{\beta}{2}}(\widetilde{X})$.

For $j \in [n]$

Set $Y_j \leftarrow \frac{1}{\sqrt{L}} \widehat{\Sigma}^{-1/2}(X_j - \widehat{\mu})$.

`// Set range parameters for private applications.`

$$R = \sqrt{U/L} \cdot \sqrt{\log(6/\beta)} \quad \text{and} \quad K = U/L$$

`// Estimate Gaussian using private data and private algorithm.`

Set $Y \leftarrow (Y_1, \ldots, Y_n)$ and $(\widehat{\mu}_Y, \widehat{\Sigma}_Y) \leftarrow \mathrm{DPGaussianEstimator}_{\alpha, \frac{\beta}{2}, \mathrm{PrivParams}, R, K}(Y)$.

Set

$$\widehat{\mu}_X \leftarrow \sqrt{L}\widehat{\Sigma}^{1/2}\widehat{\mu}_Y + \widehat{\mu} \quad \text{and} \quad \widehat{\Sigma}_X \leftarrow L\widehat{\Sigma}^{1/2}\widehat{\Sigma}_Y\widehat{\Sigma}^{1/2}$$

Return $\widehat{\mu}_X, \widehat{\Sigma}_X$.

The proof of the correctness of Algorithm 2 relies on proof of correctness for public data preconditioning, which was given in Section 2. The following facts were used for the proof in that section.

Fact B.3 (Singular Values of Gaussian Matrices [SST06]). *Let $Z \in \mathbb{R}^{d\times d}$ be a matrix with each $Z_{ij} \sim N(0,1)$ independently. Denote by $\sigma_d(Z)$ the smallest singular value of Z, and by $\sigma_1(Z)$ its largest singular value. Then we have the following.*

1. $\mathbb{P}\left(|\sigma_d(Z)| \le \beta/\sqrt{d}\right) \le \beta$
2. $\mathbb{P}\left(|\sigma_1(Z)| \ge 2\sqrt{d} + \sqrt{2\log(1/\beta)}\right) \le \beta$

Fact B.4 (Properties of Wishart Distribution[4]). *Let $Z_1, \ldots, Z_{m+1}$, $W_1, \ldots, W_m$ be chosen i.i.d. from $\mathcal{N}(0, \mathbb{I})$ over $\mathbb{R}^d$. Suppose $\widehat{\mu} = \frac{1}{m+1}\sum_{i=1}^{m+1} Z_i$. Then*

$$\frac{1}{m}\sum_{i=1}^{m+1}(Z_i - \widehat{\mu})(Z_i - \widehat{\mu})^T \sim \frac{1}{m}\sum_{i=1}^{m} W_i W_i^T \sim \frac{1}{m}\mathcal{W}_d(m, \mathbb{I}).$$

In other words, the two quantities are identically distributed according to the scaled, d-dimensional Wishart distribution with m degrees of freedom.

Below is the same-distribution version of Theorem 2.4, which employs Lemma 2.5 from Section 2.

Theorem B.5 (Public-Private Gaussian Estimation). *For all $\alpha, \beta, \varepsilon, \rho > 0$, there exist ε-DP and ρ-zCDP algorithms that take $d+1$ public samples $\widetilde{X} = (\widetilde{X}_1, \ldots, \widetilde{X}_{d+1})$ and n private samples $X = (X_1, \ldots, X_n)$ from an unknown, d-dimensional Gaussian $\mathcal{N}(\mu, \Sigma)$, and are private with respect to the private samples, and return $\widehat{\mu}_X \in \mathbb{R}^d$ and $\widehat{\Sigma}_X \in \mathbb{R}^{d\times d}$, such that $\mathrm{d_{TV}}(\mathcal{N}(\widehat{\mu}_X, \widehat{\Sigma}_X), \mathcal{N}(\mu, \Sigma)) \le \alpha$ with probability at least $1-\beta$, as long as the following bounds on n hold.*

1. *For ρ-zCDP (with computational efficiency) via [KLSU19] and $d+1$ public samples,*

$$n = O\left(\frac{d^2 + \log\left(\frac{1}{\beta}\right)}{\alpha^2} + \frac{d^2 \cdot \mathrm{polylog}\left(\frac{d}{\alpha\beta\rho}\right)}{\alpha\sqrt{\rho}}\right).$$

2. *For ε-DP (without computational efficiency) via [BKSW19] and $d+1$ public samples,*

$$n = O\left(\frac{d^2 + \log\left(\frac{1}{\beta}\right)}{\alpha^2} + \frac{d^2 \log\left(\frac{d}{\alpha\beta}\right)}{\alpha\varepsilon}\right).$$

Proof. We prove the privacy and the utility guarantees for Algorithm 2. In this proof, we will work in the ρ-zCDP regime (where, $\mathrm{PrivParams} = \{\rho\}$) using the Gaussian estimator of [KLSU19] (the second algorithm in Lemma A.24), and the analogous guarantees in the ε-DP case (where $\mathrm{PrivParams} = \{\varepsilon\}$) would follow by the same argument (but by using the third algorithm from Lemma A.24, instead).

We start by proving the utility guarantees first. Applying our public data preconditioner (Algorithm 1 with public data $\widetilde{X}$ as input) on X with target failure probability $\beta/2$ yields $Y_1, \ldots, Y_n$, such that for each $j \in [n]$, $Y_j \sim \mathcal{N}(\mu_Y, \Sigma_Y)$, where μ_Y and Σ_Y are quantites as defined in Lemma 2.5. Then with probability $\ge 1 - \beta/2$ over the sampling of public data, we have $\mathbb{I} \preceq \Sigma_Y \preceq \frac{U}{L}\mathbb{I}$ and $\|\mu_Y\|_2 \le \sqrt{U/L} \cdot \sqrt{5\log(6/\beta)}$ (Lemma 2.5). Hence, we can set $K = U/L = O(d^2\log(1/\beta)/\beta^2)$, and $R = \sqrt{U/L} \cdot \sqrt{5\log(6/\beta)} = O(d\log(1/\beta)/\beta)$, and run the ρ-zCDP Gaussian estimator on $Y_1, \ldots, Y_n$ with target failure probability $\beta/2$. We obtain our private sample complexity, which is now independent of the range parameters of the underlying distribution, by plugging in these values into the private Gaussian estimator's sample complexity.

[4] See Theorem 6 from https://www.stat.pitt.edu/sungkyu/course/2221Fall13/lec2.pdf.

Under these parameter settings and sample complexity, by Lemma A.24 and the union bound, we have that with probability $\geq 1-\beta$, the algorithm succeeds in outputting $\widehat{\mu}_Y$ and $\widehat{\Sigma}_Y$, such that $\mathrm{d_{TV}}(\mathcal{N}(\widehat{\mu}_Y, \widehat{\Sigma}_Y), \mathcal{N}(\mu_Y, \Sigma_Y)) \leq \alpha$. We output the estimates $\widehat{\Sigma}_X := L\widehat{\Sigma}^{1/2}\widehat{\Sigma}_Y\widehat{\Sigma}^{1/2}$ and $\widehat{\mu}_X := \sqrt{L}\widehat{\Sigma}^{1/2}\widehat{\mu}_Y + \widehat{\mu}$. Denoting $A := \frac{1}{L}\widehat{\Sigma}^{-1}$, by the properties of the Mahalanobis norm, $\|\cdot\|_\Sigma$:

$$\begin{aligned}
\|\widehat{\Sigma}_X - \Sigma\|_\Sigma &= \|A^{1/2}\widehat{\Sigma}_X A^{1/2} - A^{1/2}\Sigma A^{1/2}\|_{A^{1/2}\Sigma A^{1/2}} = \|\widehat{\Sigma}_Y - \Sigma_Y\|_{\Sigma_Y} \\
\|\widehat{\mu}_X - \mu\|_\Sigma &= \|A^{1/2}\widehat{\mu}_X - A^{1/2}\mu\|_{A^{1/2}\Sigma A^{1/2}} \\
&= \|(\widehat{\mu}_Y + A^{1/2}\widehat{\mu}) - (\mu_Y + A^{1/2}\widehat{\mu})\|_{\Sigma_Y} \\
&= \|\widehat{\mu}_Y - \mu_Y\|_{\Sigma_Y}
\end{aligned}$$

which implies that $\mathrm{d_{TV}}(\mathcal{N}(\widehat{\mu}_X, \widehat{\Sigma}_X), \mathcal{N}(\mu, \Sigma)) = \mathrm{d_{TV}}(\mathcal{N}(\widehat{\mu}_Y, \widehat{\Sigma}_Y), \mathcal{N}(\mu_Y, \Sigma_Y)) \leq \alpha$.

To argue about privacy, note that releasing $\widehat{\mu}_X$ and $\widehat{\Sigma}_X$ is ρ-zCDP with respect to Y, since it is post-processing (Lemma A.18) of the output $(\widehat{\mu}_Y, \widehat{\Sigma}_Y)$ of a ρ-zCDP algorithm using public information. To argue about the ρ-zCDP of our algorithm *with respect to the private dataset* X, note that for any fixed public dataset $\widetilde{X}$, the application of $\mathrm{DPGaussianEstimator}$ from Lemma A.24 is robust to changing one sample Y_j arbitrarily. Since each X_j maps to exactly one Y_j, $\mathrm{DPGaussianEstimator}$ is robust against changing any X_j arbitrarily, as well. This gives us the final privacy guarantee with respect to X. □

B.2 Different Public Data and Private Data Distributions

In this section, we give results for the scenario, where the public data does not come from the same distribution as that of the private data. Specifically, we show that in the case, where our public data comes from another Gaussian within TV distance γ of our private data distribution, a slight modification of Algorithm 1 (public data preconditioning) will also work to reduce our unbounded private estimation problem to a bounded one.

B.2.1 Technical Lemmata

We start by stating results about different distance metrics for distributions. Let $\mathrm{d_H}(\cdot,\cdot)$ denote the Hellinger distance between two distributions, and let $\mathrm{d_{TV}}(\cdot,\cdot)$ denote their total variation (TV) distance. We start with a known fact about the relation between $\mathrm{d_H}(\cdot,\cdot)^2$ and $\mathrm{d_{TV}}(\cdot,\cdot)$.

Lemma B.6 (Hellinger Distance vs. TV Distance)**.** *Let P, Q be distributions over R^d. Then* $\mathrm{d_H}(P,Q)^2 \leq \mathrm{d_{TV}}(P,Q)$.

Next, we state the expression for Hellinger distance between two univariate Gaussians.

Lemma B.7 (Hellinger Distance for Gaussians)**.** *Let $G_1 \equiv \mathcal{N}(\mu_1, \sigma_1^2), G_2 \equiv \mathcal{N}(\mu_2, \sigma_2^2)$ be Gaussians over $\mathbb{R}$. Then*

$$\mathrm{d_H}(G_1, G_2)^2 = 1 - \sqrt{\frac{2\sigma_1\sigma_2}{\sigma_1^2 + \sigma_2^2}} \cdot e^{-\frac{(\mu_1-\mu_2)^2}{4(\sigma_1^2+\sigma_2^2)}}.$$

Now, we lower-bound the Hellinger distance between two univariate Gaussians.

Lemma B.8 (Hellinger Distance Lower Bounds)**.** *Let $G_1 \equiv \mathcal{N}(\mu_1, \sigma_1^2), G_2 \equiv \mathcal{N}(\mu_2, \sigma_2^2)$ be Gaussians over $\mathbb{R}$. Suppose $\sigma_{\max} = \max\{\sigma_1, \sigma_2\}$. Then $\mathrm{d_H}(G_1, G_2)^2 \geq \mathrm{d_H}(\mathcal{N}(\mu_1, \sigma_{\max}^2), \mathcal{N}(\mu_2, \sigma_{\max}^2))^2$ and $\mathrm{d_H}(G_1, G_2)^2 \geq \mathrm{d_H}(\mathcal{N}(0, \sigma_1^2), \mathcal{N}(0, \sigma_2^2))^2$.*

Proof. Using Lemma B.7, we have the following.

$$\begin{aligned}
\mathrm{d_H}(G_1, G_2)^2 &= 1 - \sqrt{\frac{2\sigma_1\sigma_2}{\sigma_1^2 + \sigma_2^2}} \cdot e^{-\frac{(\mu_1-\mu_2)^2}{4(\sigma_1^2+\sigma_2^2)}} \\
&\geq 1 - e^{-\frac{(\mu_1-\mu_2)^2}{4(\sigma_1^2+\sigma_2^2)}} \\
&\geq 1 - e^{-\frac{(\mu_1-\mu_2)^2}{8\sigma_{\max}^2}}
\end{aligned}$$

$$= \mathrm{d_H}(\mathcal{N}(\mu_1, \sigma_{\max}^2), \mathcal{N}(\mu_2, \sigma_{\max}^2))^2$$

In the above, the second line follows from AM-GM inequality, that is, $\frac{\sigma_1^2+\sigma_2^2}{2} \geq \sigma_1\sigma_2$. This proves the first part. Now, we prove the second part.

$$\begin{aligned}
\mathrm{d_H}(G_1, G_2)^2 &= 1 - \sqrt{\frac{2\sigma_1\sigma_2}{\sigma_1^2+\sigma_2^2}} \cdot e^{-\frac{(\mu_1-\mu_2)^2}{4(\sigma_1^2+\sigma_2^2)}} \\
&\geq 1 - \sqrt{\frac{2\sigma_1\sigma_2}{\sigma_1^2+\sigma_2^2}} \\
&= \mathrm{d_H}(\mathcal{N}(0, \sigma_1^2), \mathcal{N}(0, \sigma_2^2))^2
\end{aligned}$$

In the above, the second line follows from the fact that $e^{-x} \leq 1$ for $x \geq 0$. This completes our proof. □

The next lemma describes the relation between the parameters of two univariate Gaussians when their TV distance is upper-bounded.

Lemma B.9 (TV Distance and Gaussian Parameters)**.** *Let $G_1 \equiv \mathcal{N}(\mu_1, \sigma_1^2), G_2 \equiv \mathcal{N}(\mu_2, \sigma_2^2)$ be Gaussians over $\mathbb{R}$. Suppose $\sigma_{\max} = \max\{\sigma_1, \sigma_2\}$ and $\sigma_{\min} = \min\{\sigma_1, \sigma_2\}$. If $\mathrm{d_{TV}}(G_1, G_2) \leq \alpha$, then*

$$\frac{(\mu_2-\mu_1)^2}{\sigma_{\max}^2} \leq \frac{8\alpha}{1-\alpha} \quad \textit{and} \quad \frac{\sigma_{\max}}{\sigma_{\min}} \leq \frac{2}{(1-\alpha)^2}.$$

Proof. We have the following from Lemmata B.6 and B.8.

$$\begin{aligned}
\alpha &\geq \mathrm{d_{TV}}(G_1, G_2) \\
&\geq \mathrm{d_H}(\mathcal{N}(\mu_1, \sigma_{\max}^2), \mathcal{N}(\mu_2, \sigma_{\max}^2))^2 \\
&= 1 - e^{-\frac{(\mu_1-\mu_2)^2}{8\sigma_{\max}^2}} \\
\iff e^{-\frac{(\mu_1-\mu_2)^2}{8\sigma_{\max}^2}} &\geq 1-\alpha \\
\iff \frac{(\mu_2-\mu_1)^2}{\sigma_{\max}^2} &\leq 8\ln\left(\frac{1}{1-\alpha}\right) \\
&\leq \frac{8\alpha}{1-\alpha}
\end{aligned}$$

The last line holds because $\frac{1}{1-\alpha} = 1 + \frac{\alpha}{1-\alpha}$, and $x \geq \ln(1+x)$ for all $x \in \mathbb{R}$.

For the second part, we apply Lemmata B.6 and B.8 again to get the following.

$$\begin{aligned}
\alpha &\geq \mathrm{d_{TV}}(G_1, G_2) \\
&\geq \mathrm{d_H}(\mathcal{N}(0, \sigma_1^2), \mathcal{N}(0, \sigma_2^2))^2 \\
&= 1 - \sqrt{\frac{2\sigma_1\sigma_2}{\sigma_1^2+\sigma_2^2}} \\
\iff \frac{1}{(1-\alpha)^2} &\geq \frac{\sigma_1^2+\sigma_2^2}{2\sigma_1\sigma_2} \\
\iff \frac{2}{(1-\alpha)^2} &\geq \frac{\sigma_1}{\sigma_2} + \frac{\sigma_2}{\sigma_1} \\
&= \frac{\sigma_{\max}}{\sigma_{\min}} + \frac{\sigma_{\min}}{\sigma_{\max}} \\
\implies \frac{2}{(1-\alpha)^2} &\geq \frac{\sigma_{\max}}{\sigma_{\min}}
\end{aligned}$$

□

Finally, we state a multivariate analogue of Lemma B.9.

Lemma B.10. *Suppose* $\mathrm{d_{TV}}(\mathcal{N}(\mu,\Sigma),\mathcal{N}(\widetilde{\mu},\widetilde{\Sigma})) \le \gamma$ *for some* $\gamma < 1$. *Then*

1. $\frac{(1-\gamma)^4}{4}\widetilde{\Sigma} \preceq \Sigma \preceq \frac{4}{(1-\gamma)^4}\widetilde{\Sigma}$
2. $(\mu-\widetilde{\mu})(\mu-\widetilde{\mu})^T \preceq \frac{8\gamma}{1-\gamma}(\Sigma+\widetilde{\Sigma})$

Proof. Let $G_1 \equiv \mathcal{N}(\mu,\Sigma)$ and $G_2 \equiv \mathcal{N}(\widetilde{\mu},\widetilde{\Sigma})$. For any unit vector $v \in \mathbb{R}^d$, denote by $v^T G_i$ the distribution over $\mathbb{R}$ obtained by sampling $x \sim G_i$ and outputting $v^T x$. A data-processing inequality for the TV distance gives us that, for any unit vector $v \in \mathbb{R}^d$,

$$\mathrm{d_{TV}}(\mathcal{N}(v^T\mu, v^T\Sigma v), \mathcal{N}(v^T\widetilde{\mu}, v^T\widetilde{\Sigma}v)) = \mathrm{d_{TV}}(v^T G_1, v^T G_2) \le \mathrm{d_{TV}}(G_1, G_2)$$

where the first equality comes from the fact that the projection of a Gaussian is also Gaussian, with the above parameters.

By the second univariate bound in Lemma B.9, we have that for every unit vector $v \in \mathbb{R}^d$,

$$\frac{(1-\gamma)^4}{4} \le \frac{v^T\Sigma v}{v^T\widetilde{\Sigma}v} \le \frac{4}{(1-\gamma)^4}.$$

Rearranging the above gives us (1).

For (2), we have that for every unit vector $v \in \mathbb{R}^d$,

$$\frac{v^T(\widetilde{\mu}-\mu)(\widetilde{\mu}-\mu)^T v}{v^T(\Sigma+\widetilde{\Sigma})v} = \frac{(v^T\widetilde{\mu}-v^T\mu)^2}{v^T\Sigma v + v^T\widetilde{\Sigma}v} \le \frac{(v^T\widetilde{\mu}-v^T\mu)^2}{\max\{v^T\Sigma v, v^T\widetilde{\Sigma}v\}} \le \frac{8\gamma}{1-\gamma}$$

where the last inequality comes from applying the first univariate bound in Lemma B.9. Rearranging the above gives us (2). □

B.2.2 Unknown Mean and Covariance with $d+1$ Public Samples

Let L, U be quantities, as defined in Algorithm 1, and let $L_\gamma = \frac{(1-\gamma)^4}{4}\cdot L$ and $U_\gamma = \frac{4}{(1-\gamma)^4}\cdot U$ for $0 \le \gamma < 1$. The following is an analogue of Lemma 2.5 for the case, where we apply Algorithm 1 (public data preconditioning) with public data that comes from a Gaussian that is at most γ-far in TV distance from the private data distribution.

Lemma B.11 (γ-Far Public Data Preconditioning)**.** *For all* $\beta > 0$, *there exists an algorithm that takes* $d+1$ *independent samples* $\widetilde{X} = (\widetilde{X}_1, \ldots, \widetilde{X}_{d+1})$ *from a Gaussian* $\mathcal{N}(\widetilde{\mu},\widetilde{\Sigma})$ *over* $\mathbb{R}^d$, *and outputs* $\widehat{\mu} \in \mathbb{R}^d$, $\widehat{\Sigma} \in \mathbb{R}^{d\times d}$, $L_\gamma \in \mathbb{R}$, *and* $U_\gamma \in \mathbb{R}$, *such that for a Gaussian* $\mathcal{N}(\mu,\Sigma)$ *over* $\mathbb{R}^d$ *with* $0 \le \mathrm{d_{TV}}(\mathcal{N}(\mu,\Sigma),\mathcal{N}(\widetilde{\mu},\widetilde{\Sigma})) \le \gamma < 1$, *if* $\Sigma_Y = \frac{1}{L_\gamma}\widehat{\Sigma}^{-1/2}\Sigma\widehat{\Sigma}^{-1/2}$ *and* $\mu_Y = \frac{1}{\sqrt{L_\gamma}}\widehat{\Sigma}^{-1/2}(\mu-\widehat{\mu})$, *then with probability at least* $1-\beta$ *over the sampling of the data,*

1. $\mathbb{I} \preceq \Sigma_Y \preceq \frac{U_\gamma}{L_\gamma}\mathbb{I}$
2. $\|\mu_Y\|_2 \le \sqrt{\frac{U_\gamma}{L_\gamma}}\cdot\left(\sqrt{\frac{10\gamma}{1-\gamma}} + \sqrt{5\log(3/\beta)}\right)$

where $\frac{U_\gamma}{L_\gamma} = O\left(\frac{d^2\log(1/\beta)}{\beta^2(1-\gamma)^8}\right)$.

Proof. We prove the lemma by proving the utility of a modified version of Algorithm 1, which returns L_γ and U_γ, instead of L and U, respectively. The result follows from tracing through the proof of Lemma 2.5, and applying Lemma B.10 as necessary. We highlight the differences.

We start with (1). By the same chain of equivalences in the proof of Lemma 2.5, it suffices to show that $L_\gamma\Sigma^{-1/2}\widehat{\Sigma}\Sigma^{-1/2} \preceq \mathbb{I} \preceq U_\gamma\Sigma^{-1/2}\widehat{\Sigma}\Sigma^{-1/2}$. We have the following.

$$\Sigma^{-1/2}\widehat{\Sigma}\Sigma^{-1/2} = \Sigma^{-1/2}\left(\frac{1}{d}\sum_{i=1}^{d+1}((\widetilde{\Sigma}^{1/2}Z_i+\widetilde{\mu})-\widehat{\mu})((\widetilde{\Sigma}^{1/2}Z_i+\widetilde{\mu})-\widehat{\mu})^T\right)\Sigma^{-1/2}$$

$$= \Sigma^{-1/2}\left(\frac{1}{d}\sum_{i=1}^{d+1}(\widetilde{\Sigma}^{1/2}(Z_i-\widehat{\mu}_Z))(\widetilde{\Sigma}^{1/2}(Z_i-\widehat{\mu}_Z))^T\right)\Sigma^{-1/2}$$
$$= \Sigma^{-1/2}\widetilde{\Sigma}^{1/2}\widehat{\Sigma}_Z\widetilde{\Sigma}^{1/2}\Sigma^{-1/2}$$

In the above, Z_i, $\widehat{\mu}_Z$, and $\widehat{\Sigma}_Z$ are quantities, as defined in the proof of Lemma 2.5. From the same proof, we know that with probability $\geq 1-\beta/3$, we have $U\widehat{\Sigma}_Z \succeq \mathbb{I}$, which implies that

$$U\Sigma^{-1/2}\widetilde{\Sigma}^{1/2}\widehat{\Sigma}_Z\widetilde{\Sigma}^{1/2}\Sigma^{-1/2} \succeq \Sigma^{-1/2}\widetilde{\Sigma}\Sigma^{-1/2} \succeq \frac{(1-\gamma)^4}{4}\cdot\mathbb{I}$$

where the last inequality follows from (1) in Lemma B.10. Recalling that we set $U_\gamma = \frac{4}{(1-\gamma)^4}\cdot U$, and rearranging gives us that $U_\gamma\Sigma^{-1/2}\widehat{\Sigma}\Sigma^{-1/2} \succeq \mathbb{I}$, as desired. Similarly, with probability $\geq 1-\beta/3$, $L\widehat{\Sigma}_Z \preceq \mathbb{I}$, which implies that

$$L\Sigma^{-1/2}\widetilde{\Sigma}^{1/2}\widehat{\Sigma}_Z\widetilde{\Sigma}^{1/2}\Sigma^{-1/2} \preceq \Sigma^{-1/2}\widetilde{\Sigma}\Sigma^{-1/2} \preceq \frac{4}{(1-\gamma)^4}\cdot\mathbb{I}$$

which again, after rearranging, allows us to conclude $L_\gamma\Sigma^{-1/2}\widehat{\Sigma}\Sigma^{-1/2} \preceq \mathbb{I}$.

It remains to verify that (2) holds. Write

$$\mu_Y = \frac{1}{\sqrt{L_\gamma}}\widehat{\Sigma}^{-1/2}(\mu-\widehat{\mu}) = \frac{1}{\sqrt{L_\gamma}}\widehat{\Sigma}^{-1/2}(\mu-\widetilde{\mu}-\widetilde{\Sigma}^{1/2}\widehat{\mu}_Z) = \frac{1}{\sqrt{L_\gamma}}\widehat{\Sigma}^{-1/2}(\mu-\widetilde{\mu})-\frac{1}{\sqrt{L_\gamma}}\widehat{\Sigma}^{-1/2}\widetilde{\Sigma}^{1/2}\widehat{\mu}_Z.$$

We bound the two terms separately. Note that the second term appears in the proof of Lemma 2.5. By the same argument as in that proof, we claim that with probability $\geq 1-\frac{\beta}{3}$, $\|\widehat{\mu}_Z\|_2 \leq \sqrt{\frac{d+2\sqrt{d\log(3/\beta)}+2\log(3/\beta)}{d+1}}$. We already know that $\left\|\widehat{\Sigma}^{-1/2}\widetilde{\Sigma}^{1/2}\right\|_2 \leq \sqrt{U}$. Taking the union bound, with probability $\geq 1-\beta$,

$$\left\|-\frac{1}{\sqrt{L_\gamma}}\widehat{\Sigma}^{-1/2}\widetilde{\Sigma}^{1/2}\widehat{\mu}_Z\right\|_2 \leq \sqrt{\frac{U}{L_\gamma}}\cdot\sqrt{\frac{d+2\sqrt{d\log(3/\beta)}+2\log(3/\beta)}{d+1}}.$$

Now, we argue that the first term is also bounded. First, we apply Lemma B.10 to get

$$(\mu-\tilde{\mu})(\mu-\tilde{\mu})^T \preceq \frac{8\gamma}{1-\gamma}(\Sigma+\widetilde{\Sigma}) \preceq \frac{8\gamma}{1-\gamma}\left(\frac{4}{(1-\gamma)^4}\widetilde{\Sigma}+\widetilde{\Sigma}\right) \preceq \frac{40\gamma}{(1-\gamma)^5}\widetilde{\Sigma}.$$

Note that $U\widehat{\Sigma}_Z \succeq \mathbb{I} \implies \widetilde{\Sigma} \preceq U\widehat{\Sigma}$. Plugging this in above, and rearranging gives

$$\widehat{\Sigma}^{-1/2}(\mu-\tilde{\mu})(\mu-\tilde{\mu})^T\widehat{\Sigma}^{-1/2} \preceq \frac{4}{(1-\gamma)^4}\cdot U\cdot\frac{10\gamma}{1-\gamma}\cdot\mathbb{I} = U_\gamma\cdot\frac{10\gamma}{1-\gamma}\cdot\mathbb{I}.$$

Thus, we have

$$\left\|\frac{1}{\sqrt{L_\gamma}}\widehat{\Sigma}^{-1/2}(\mu-\tilde{\mu})\right\|_2 \leq \frac{1}{\sqrt{L_\gamma}}\cdot\sqrt{\sigma_1(\widehat{\Sigma}^{-1/2}(\mu-\tilde{\mu})(\mu-\tilde{\mu})^T\widehat{\Sigma}^{-1/2})} \leq \sqrt{\frac{U_\gamma}{L_\gamma}}\cdot\sqrt{\frac{10\gamma}{1-\gamma}}.$$

Combining the two terms gives us

$$\|\mu_Y\|_2 \leq \sqrt{\frac{U_\gamma}{L_\gamma}}\cdot\left(\sqrt{\frac{10\gamma}{1-\gamma}}+\sqrt{\frac{d+2\sqrt{d\log(3/\beta)}+2\log(3/\beta)}{d+1}}\right)$$

which completes the proof. □

Lemma B.11, combined with guarantees of the Gaussian estimators from Lemma A.24, allows us to conclude the following analogue to Theorem B.5 (this stated in the main body as Theorem 2.4).

Theorem B.12 (γ-Far Public-Private Gaussian Estimation)**.** *For all $\alpha, \beta, \varepsilon, \rho > 0$, there exist ε-DP and ρ-zCDP algorithms that take $d+1$ public samples $\widetilde{X} = (\widetilde{X}_1, \ldots, \widetilde{X}_{d+1})$ and n private samples $X = (X_1, \ldots, X_n)$ from unknown, d-dimensional Gaussians $\mathcal{N}(\widetilde{\mu}, \widetilde{\Sigma})$ and $\mathcal{N}(\mu, \Sigma)$, respectively, such that $0 \leq \mathrm{d_{TV}}(\mathcal{N}(\mu, \Sigma), \mathcal{N}(\widetilde{\mu}, \widetilde{\Sigma})) \leq \gamma < 1$, and are private with respect to the private samples, and return $\widehat{\mu}_X \in \mathbb{R}^d$ and $\widehat{\Sigma}_X \in \mathbb{R}^{d \times d}$, such that $\mathrm{d_{TV}}(\mathcal{N}(\widehat{\mu}_X, \widehat{\Sigma}_X), \mathcal{N}(\mu, \Sigma)) \leq \alpha$ with probability at least $1 - \beta$, as long as the following bounds on n hold.*

1. *For ρ-zCDP (with computational efficiency) via [KLSU19] and $d+1$ public samples,*

$$n = O\left(\frac{d^2 + \log\left(\frac{1}{\beta}\right)}{\alpha^2} + \frac{d^2 \cdot \mathrm{polylog}\left(\frac{d}{\alpha\beta\rho}\right) + d\log\left(\frac{d}{\alpha\beta\rho(1-\gamma)}\right)}{\alpha\sqrt{\rho}} + \frac{d^{1.5} \cdot \mathrm{polylog}\left(\frac{d}{\beta\rho(1-\gamma)}\right)}{\sqrt{\rho}}\right).$$

2. *For ε-DP (without computational efficiency) via [BKSW19] and $d+1$ public samples,*

$$n = O\left(\frac{d^2 + \log\left(\frac{1}{\beta}\right)}{\alpha^2} + \frac{d^2 \log\left(\frac{d}{\alpha\beta(1-\gamma)}\right)}{\alpha\varepsilon}\right).$$

Proof. The theorem follows from the privacy and the utility guarantees of a modified version of Algorithm 2, which uses the modified version of Algorithm 1 as outlined in Lemma B.11, and parameters L_γ and U_γ, instead. The proof remains the same as that of Theorem B.5. □

C Estimating Gaussian Mixtures

In this section, we state our algorithms to learn mixtures of well-separated Gaussians under differential privacy, when we have trace amounts of public data available to us. We then provide theoretical guarantees for privacy of our algorithms with respect to the private data samples, along with its utility. We analyse two cases here: when we have $\widetilde{O}(1/w_{\min})$ public samples available to us, and when we have $\widetilde{O}(d/w_{\min})$ public samples available.

The general scheme of our algorithms is similar to that of the non-private algorithm for learning Mixtures of Gaussians by [AM05], and the private algorithm (with no public data available) by [KSSU19]. Many algorithms for learning mixtures of Gaussians follow a specific outline – use PCA to project the data onto a low-dimensional subspace, which would separate the largest Gaussian from the rest; partition the dataset again, if there is more than one Gaussian present, otherwise isolate the lone Gaussian; estimate the parameters of that Gaussian; repeat the process on the remaining points. Our algorithms are also spectral algorithms that rely upon techniques, like PCA, but use their private counterparts at various stages.

The difference in the aforementioned cases, where we have $O(1/w_{\min})$ and $O(d/w_{\min})$ public samples available, is that in the former, we have very few public samples, but they are enough to be able to isolate a group of nearby clusters (called, a "supercluster"), which we could then use for an application of private PCA. In the latter case, we wouldn't need to apply private PCA at all because the number of public samples is enough to be able to do non-private PCA accurately.

Assumption. We make an assumption about the shape of the covariances of all Gaussians, which essentially says that the Gaussians are not too flat or degenerate. Let N be the total number of points sampled from $D \in \mathcal{G}(d, k, s)$. We formalise this as follows.

$$\forall i \in [k], \quad \|\Sigma_i\|_F \sqrt{\log(Nk/\beta)} \leq \frac{1}{8}\mathrm{tr}(\Sigma_i) \quad \text{and} \quad \|\Sigma_i\|_2 \log^2(Nk/\beta) \leq \frac{1}{8}\mathrm{tr}(\Sigma_i) \tag{2}$$

Note that this implies that $d \geq 8\log^2(Nk/\beta)$ because $\mathrm{tr}(\Sigma_i) \leq d\|\Sigma_i\|_2$. We also assume that $\beta < 1/2$.

Notations. We also define a few notations before moving on to the technical sections. We say that a Gaussian $\mathcal{N}(\mu, \Sigma)$ in high dimensions satisfying Condition 2 is contained within $S \subset \mathbb{R}^d$, if $B_{\sqrt{\frac{3}{2}\mathrm{tr}(\Sigma)}}(\mu) \subseteq S$. For a low-dimensional Gaussian $\mathcal{N}(\mu, \Sigma')$ in $\ell < d$ dimensions, we say that a set $S \subset \mathbb{R}^d$ contains the Gaussian for fixed $0 < \beta < 1$ and $N \geq 1$, if $B_{\sqrt{\|\Sigma'\|_2}\sqrt{2\ell \ln(2N\ell/\beta)}}(\mu') \subset S$. Also, given $D \in \mathcal{G}(d, k, s)$, we say that $B \subset \mathbb{R}^d$ is "pure", if for each Gaussian component i, B either contains the Gaussian $\mathcal{N}(\mu_i, \Sigma_i)$, or $B_{\sqrt{\frac{3}{2}\mathrm{tr}(\Sigma_i)}}(\mu_i) \cap B = \emptyset$. Similarly, given a set of points T from D, we say that $S \subseteq T$ is "clean", if for every $i \in [k]$, S has all points from component i that lie in T, or it has none of them. We sometimes say that for a clean $S_1 \subseteq T$, S_2 is a clean subset of S_1 if $S_2 \subseteq S_1$ and S_2 is clean with respect to T.

C.1 Deterministic Regularity Conditions

Here, we state a few results that would be useful in solving the problems in the next two subsections. The following condition bounds the number of points from each component of a Gaussian mixture.

Condition C.1 (Sample Frequency)**.** Suppose we have n samples from a mixture of Gaussians $D = \{(\mu_1, \Sigma_1, w_1), \ldots, (\mu_k, \Sigma_k, w_k)\} \in \mathcal{G}(d, k, s)$ satisfying Condition 2. Then for each $i \in [k]$, the number of samples from component i lies in $\left[\frac{nw_i}{2}, \frac{3nw_i}{2}\right]$. Furthermore, if $w_i \geq \frac{4\alpha}{9k}$, then the number of points from component i lies in $\left[n(w_i - \frac{\alpha}{9k}), n(w_i + \frac{\alpha}{9k})\right]$.

The next condition bounds the distance of any point sampled from a Gaussian mixture from the mean of its respective component.

Condition C.2 (Intra-Gaussian Distances From Mean)**.** Suppose we have n samples from a mixture of Gaussians $D = \{(\mu_1, \Sigma_1, w_1), \ldots, (\mu_k, \Sigma_k, w_k)\} \in \mathcal{G}(d, k, s)$ satisfying Condition 2. Then for each $i \in [k]$, if x is one of the samples from component i, then $\frac{3}{4}\mathrm{tr}(\Sigma_i) \leq \|x - \mu_i\|^2 \leq \frac{3}{2}\mathrm{tr}(\Sigma_i)$.

This condition bounds the distance between any two points sampled from the same component of a Gaussian mixture.

Condition C.3 (Intra-Gaussian Distances Between Points)**.** Suppose we have n samples from $D = \{(\mu_1, \Sigma_1, w_1), \ldots, (\mu_k, \Sigma_k, w_k)\} \in \mathcal{G}(d, k, s)$ satisfying Condition 2. Then for each $i \in [k]$, if x, y are two of the samples from component i, then $\frac{3}{2}\mathrm{tr}(\Sigma_i) \leq \|x - y\|^2 \leq 3\mathrm{tr}(\Sigma_i)$.

This final condition quantifies the minimum distance between any two points from different components of a Gaussian mixture.

Condition C.4 (Inter-Gaussian Distances)**.** Suppose we have n samples from a mixture of Gaussians $D = \{(\mu_1, \Sigma_1, w_1), \ldots, (\mu_k, \Sigma_k, w_k)\} \in \mathcal{G}(d, k, s)$ satisfying Condition 2, where $s \geq \Omega(\sqrt{\ln(n/\beta)})$. Then for any $i, j \in [k]$ with $i \neq j$, if x is one of the samples from component i, and y is one of the samples from component j, then $\|x - y\| \geq \frac{\sqrt{\max\{\mathrm{tr}(\Sigma_i), \mathrm{tr}(\Sigma_j)\}}}{4}$.

We state another condition that says that in low dimensions, too, the distance between the mean of a Gaussian from all its points is bounded, and it is true for all the components of the mixture.

Condition C.5 (Intra-Gaussian Distances from Mean in Low Dimensions)**.** Suppose we have n samples from $D = \{(\mu_1, \Sigma_1, w_1), \ldots, (\mu_k, \Sigma_k, w_k)\} \in \mathcal{G}(\ell, k)$. For each $i \in [k]$, let $\sigma_i^2 = \|\Sigma_i\|_2$. Then for any $i \in [k]$ and a fixed $0 < \beta < 1$, if x is a datapoint sampled from $\mathcal{N}(\mu_i, \Sigma_i)$, then $\|\mu_i - x\| \leq \sigma_i\sqrt{2\ell \ln(2n\ell/\beta)}$.

The final condition about low-dimensional Gaussian mixtures bounds the distance between two points from the same component.

Condition C.6 (Intra-Gaussian Distances Between Points in Low Dimensions)**.** Suppose we have n samples from $D = \{(\mu_1, \Sigma_1, w_1), \ldots, (\mu_k, \Sigma_k, w_k)\} \in \mathcal{G}(\ell, k)$. For each $i \in [k]$, let $\sigma_i^2 = \|\Sigma_i\|_2$. Then for any $i \in [k]$ and a fixed $0 < \beta < 1$, if x, y are a datapoints sampled from $\mathcal{N}(\mu_i, \Sigma_i)$, then $\|x - y\| \leq 2\sigma_i\sqrt{2\ell \ln(2n\ell/\beta)}$.

Now, we prove that the above conditions hold with high probability.

Lemma C.7. *Suppose we have n samples from $D \in \mathcal{G}(d, k, s)$ satisfying Condition 2. If*

$$n \geq \max\left\{\frac{12}{w_{\min}}\ln(2k/\beta), \frac{405k^2}{2\alpha^2}\ln(2k/\beta)\right\},$$

then Condition C.1 holds with probability at least $1-\beta$.

Proof. We simply use Lemma A.1 k times with $p \geq w_{\min}$, and Lemma A.2 k times, and take the union bound. □

Lemma C.8. *Suppose we have n samples from $D \in \mathcal{G}(d,k,s)$ satisfying Condition 2. Then Condition C.2 holds with probability at least $1-\beta$.*

Proof. From the Hanson-Wright inequality (Lemma A.3), $\forall i, \forall x \sim \mathcal{N}(\mu_i, \Sigma_i)$, we have that

$$\mathrm{tr}(\Sigma_i) - 2\|\Sigma_i\|_F\sqrt{\log(n/\beta)} \leq \|x-\mu_i\|_2^2 \leq \mathrm{tr}(\Sigma_i) + 2\|\Sigma_i\|_F\sqrt{\log(n/\beta)} + 2\|\Sigma_i\|_2\log(n/\beta).$$

Condition 2, combined with the above, gives us the required result. □

Lemma C.9. *Suppose we have n samples from $D \in \mathcal{G}(d,k,s)$ satisfying Condition 2. Then Condition C.3 holds with probability at least $1-\beta$.*

Proof. For every i and for any $x, y \sim \mathcal{N}(\mu_i, \Sigma_i)$, we have that $x-y \sim \mathcal{N}(0, 2\Sigma_i)$. Using Condition 2 and Lemma A.3 again, we have the result. □

The following is a quantification of the median radius of a Gaussian in terms of the trace of its covariance.

Lemma C.10. *Suppose we have n samples from $D = \{(\mu_1,\Sigma_1,w_1),\ldots,(\mu_k,\Sigma_k,w_k)\} \in \mathcal{G}(d,k,s)$ satisfying Condition 2. Then for each $i \in [k]$, the median radius R_i of component i lies in $\left(\sqrt{\frac{3}{4}\mathrm{tr}(\Sigma_i)}, \sqrt{\frac{3}{2}\mathrm{tr}(\Sigma_i)}\right)$.*

Proof. For each $i \in [k]$, let $a_i = \sqrt{\frac{3}{4}\mathrm{tr}(\Sigma_i)}$ and $b_i = \sqrt{\frac{3}{2}\mathrm{tr}(\Sigma_i)}$. We know from the proof of Lemma C.8 that for a given $i \in [k]$, $1-\frac{\beta}{k}$ of the probability mass of $G_i := \mathcal{N}(\mu_i, \Sigma_i)$ lies in $B_{b_i}(\mu_i) \setminus B_{a_i}(\mu_i)$. Let $\beta_1 \leq \beta/k$ be the probability mass of G_i in $B_{a_i}(\mu_i)$, and let $\beta_2 = \beta/k - \beta_1$. We know that $\beta_1 < 1/2$, therefore, $R_i > a_i$. Since the mass of G_i outside $B_{b_i}(\mu_i)$ is $\beta_2 < 1/2$, it must be the case that $R_i < b_i$. Hence, the claim. □

Lemma C.11. *Suppose we have n samples from $D \in \mathcal{G}(d,k,s)$ satisfying Condition 2. If $s \geq \Omega(\sqrt{\ln(n/\beta)})$, then Condition C.4 holds with probability at least $1-2\beta$.*

Proof. For this, we will use Lemmata A.12 and C.10. We know from Lemma A.12 that with probability at least $1-\beta/m$, for any given $i \in [k]$, and x sampled from $G_i := \mathcal{N}(\mu_i, \Sigma_i)$ in the dataset sampled from D,

$$\|x-z\|^2 \geq (\max\{R_i - \ln(\beta/2m)\sigma_i, 0\})^2 + \|z-\mu_i\|^2 - 2\sqrt{2\ln(\beta/2m)}\sigma_i\|z-\mu_i\|.$$

For any $j \in [k]$ with $j \neq i$, let $z = \mu_j$. Suppose R_i and R_j are the median radii of G_i and G_j, respectively. WLOG, let's assume that $R_i \geq R_j$. Then we have the following.

$$\begin{aligned}\|x-\mu_j\|^2 &\geq (\max\{R_i - \ln(2m/\beta)\sigma_i, 0\})^2 + \|\mu_j-\mu_i\|^2 - 2\sqrt{2\ln(2m/\beta)}\sigma_i\|\mu_j-\mu_i\| \\ &\geq (\max\{R_i - \ln(2m/\beta)\sigma_i, 0\})^2 + \frac{\|\mu_j-\mu_i\|^2}{2} \qquad \text{(Separation condition and Condition 2.)}\end{aligned}$$

Now, let $y \sim G_j$. Due to our separation condition, we know that,

$$2\sqrt{2\ln(2m/\beta)}\sigma_j \leq \frac{\|\mu_j-\mu_i\|}{2\sqrt{2}} \leq \frac{\|x-\mu_j\|}{2}.$$

We apply Lemma A.12 again with $z = x$.

$$\begin{aligned}\|x-y\|^2 &\geq (\max\{R_j - \ln(2m/\beta)\sigma_j, 0\})^2 + \|x-\mu_j\|^2 - 2\sqrt{2\ln(2m/\beta)}\sigma_j\|x-\mu_j\| \\ &\geq (\max\{R_j - \ln(2m/\beta)\sigma_j, 0\})^2 + \frac{\|x-\mu_j\|^2}{2} \qquad \text{(Separation condition.)}\end{aligned}$$

$$
\begin{aligned}
&\geq (\max\{R_j - \ln(2m/\beta)\sigma_j, 0\})^2 + \frac{(\max\{R_i - \ln(2m/\beta)\sigma_i, 0\})^2}{4} + \frac{\|\mu_j - \mu_i\|^2}{8} \\
&\geq \frac{\mathrm{tr}(\Sigma_j)}{4} + \frac{\mathrm{tr}(\Sigma_i)}{16} + \frac{\|\mu_j - \mu_i\|^2}{8} && \text{(Condition 2.)} \\
&\geq \frac{\mathrm{tr}(\Sigma_i)}{16}
\end{aligned}
$$

This proves the lemma. □

Lemma C.12. *Suppose we have n samples from $D = \{(\mu_1, \Sigma_1, w_1), \ldots, (\mu_k, \Sigma_k, w_k)\} \in \mathcal{G}(\ell, k)$. Then for a fixed $0 < \beta < 1$, Condition C.5 holds with probability at least $1 - \beta$.*

Proof. Fix $i \in [k]$, x be an arbitrary sample from component i, and an eigenvector of Σ_i v. In that direction, using Lemma A.4, we know that with probability at least $1 - \frac{\beta}{n\ell}$, $\|(x - \mu_i)vv^T\| \leq \sigma_i\sqrt{2\ln(2n\ell/\beta)}$. Applying the union bound in all ℓ directions of Σ_i, we have that with probability at least $1 - \frac{\beta}{n}$, $\|x - \mu_i\| \leq \sigma_i\sqrt{2\ell\ln(2n\ell/\beta)}$. Applying the union bound again over all n points, we get the required result. □

Lemma C.13. *Suppose we have n samples from $D = \{(\mu_1, \Sigma_1, w_1), \ldots, (\mu_k, \Sigma_k, w_k)\} \in \mathcal{G}(\ell, k)$. Then for a fixed $0 < \beta < 1$, Condition C.6 holds with probability at least $1 - \beta$.*

Proof. The proof just involves application of triangle inequality for any pairs of points from the same component, along with Lemma C.12. □

C.2 $\widetilde{O}(1/w_{\min})$ Public Samples

In this subsection, we assume that we have $\widetilde{O}(1/w_{\min})$ public samples available. Here, the number of public samples is not enough to be able to do PCA accurately, so we, instead, try to use them to isolate the superclusters in the data, and perform private actions, like private PCA and private estimation algorithms, on the private data based on that information.

We start with a superclustering algorithm for public data. It first computes a constant-factor approximation to the diamater ($\approx r$) of the largest Gaussian in the public dataset by finding the largest minimum pairwise distance among all points. It then tries to find a supercluster around the point (x) that has the largest minimum distance to all other points. The idea is that $B_r(x)$ will have a lot of points, but if $B_{r+r}(x)$ doesn't have any more points, and $B_{r+2r}(x)$ doesn't have any more points either, then it must be the case that with high probability, no other Gaussian would have a point in $B_{2r}(x)$. This follows from the concentration properties of high-dimensional Gaussians, and the fact that r approximates the diameter of the largest Gaussian. If we do encounter more points in either $B_{2r}(x)$ or $B_{3r}(x)$, we expand the ball by $O(r)$ and check for these conditions again. We show that the ball returned by the algorithm is of size at most $O(kr)$, and that it contains the Gaussian corresponding to the point x.

We now state the main result about Algorithm 3.

Theorem C.14. *There exists an algorithm, which if given a clean subset of*

$$
m \geq O\left(\frac{\ln(k/\beta)}{w_{\min}}\right)
$$

points from $D \in \mathcal{G}(d, k)$, where D satisfies Condition 2, it outputs $c \in \mathbb{R}^d$ and $R \in \mathbb{R}$, such that the following holds, given Conditions C.1 to C.4 hold.

1. *$B_R(c)$ is pure.*
2. *Let Σ' be the covariance of the largest Gaussian (covariance with maximum trace) contained in $B_R(c)$. Then $R \in O(k\sqrt{\mathrm{tr}(\Sigma')})$.*

Proof. Let $\mathcal{N}(\mu, \Sigma)$ be the largest Gaussian that has points in $\widetilde{X}$. First, we claim that $r \in \Theta(\sqrt{\mathrm{tr}(\Sigma)})$. We know from Lemma C.11, that the minimum distance between $x \sim \mathcal{N}(\mu_i, \Sigma_i)$ and $y \sim \mathcal{N}(\mu_j, \Sigma_j)$

Algorithm 3: Superclustering on Public Data $\mathrm{SC}_{k,m}(\widetilde{X})$

Input: Samples $\widetilde{X}_1, \ldots, \widetilde{X}_{m'} \in \mathbb{R}^d$. Parameters $\beta, k, m > 0$.
Output: Centre $c \in \mathbb{R}^d$ and radius $R \in \mathbb{R}$.

Let $r \leftarrow 16 \cdot \max_{x \in \widetilde{X}} \min_{y \in \widetilde{X}} \|x - y\|$.
Let $c \leftarrow \arg\max_{x \in \widetilde{X}} \min_{y \in \widetilde{X}} \|x - y\|$.

Set $pure \leftarrow$ FALSE.
Let $R \leftarrow r$.
For $i \leftarrow 1, \ldots, k$
  Let $m_i \leftarrow \left| B_R(c) \cap \widetilde{X} \right|$.
  If $\left| B_{R+r}(c) \cap \widetilde{X} \right| = m_i$
    If $pure =$ TRUE
      Return $B_R(c)$
    elif $\left| B_{R+2r}(c) \cap \widetilde{X} \right| = m_i$
      Return $(c, R + r)$
    Else
      $pure =$ FALSE
      $R \leftarrow R + 3r$
  Else
    If $\left| B_{R+2r}(c) \cap \widetilde{X} \right| = \left| B_{R+r}(c) \cap \widetilde{X} \right|$
      $pure =$ TRUE
    Else
      $pure =$ FALSE
    $R \leftarrow R + 2r$
Return (c, R).

for $i \neq j$ is lower bounded by $\frac{\sqrt{\max\{\mathrm{tr}(\Sigma_i), \mathrm{tr}(\Sigma_j)\}}}{4}$. We also know from Lemma C.9 that for any $i \in [k]$, and any $x, z \sim \mathcal{N}(\mu_i, \Sigma_i)$, $\|x - z\| \geq \sqrt{\frac{3}{2}\mathrm{tr}(\Sigma_i)}$. Let $y \in \widetilde{X}$, such that y is sampled from component $j \neq i$ and is closest to x, that is, y is the closest point to x that has not been sampled from component i to which x belongs. This means that distance between x and its nearest point is at least $\min\left\{\sqrt{\frac{3}{2}\mathrm{tr}(\Sigma_i)}, \frac{\sqrt{\max\{\mathrm{tr}(\Sigma_i), \mathrm{tr}(\Sigma_j)\}}}{4}\right\} \geq \min\left\{\sqrt{\frac{3}{2}\mathrm{tr}(\Sigma_i)}, \frac{\sqrt{\mathrm{tr}(\Sigma_i)}}{4}\right\}$. We also know from Lemma C.9 that for any $x, z \sim \mathcal{N}(\mu_i, \Sigma_i)$, $\|x - z\| \leq \sqrt{3\mathrm{tr}(\Sigma_i)}$. This means that the point in $\widetilde{X}$ that is closest to x is at most $\sqrt{3\mathrm{tr}(\Sigma_i)}$ far from x. This shows that the distance between x and the point in $\widetilde{X}$ that is closest to x lies in $\left[\frac{\sqrt{\mathrm{tr}(\Sigma_i)}}{4}, \sqrt{3\mathrm{tr}(\Sigma_i)}\right]$. This is true for any point in $\widetilde{X}$. Therefore, the largest minimum distance has to lie in the interval $\left[\frac{\sqrt{\mathrm{tr}(\Sigma)}}{4}, \sqrt{3\mathrm{tr}(\Sigma)}\right]$. This shows that $r \in \Theta(\sqrt{\mathrm{tr}(\Sigma)})$.

We use this to show that $B_R(c)$ is pure. We prove the following claims for that.

Claim C.15. *If at the end of any iteration, the algorithm doesn't exit the loop, but calls any of the return steps within the loop instead, then the returned ball is pure.*

Proof. Suppose it happens in iteration i. Consider the case where the return statement was called in the **elif** block. If $B_R(c)$ were intersecting with a component, then $B_{R+r}(c)$ would contain it (Lemmata C.8 and C.9) because r is large enough. Now, we need to show that $B_{R+r}(c)$ doesn't intersect with any new component. By construction, it must mean that $\left| B_R(c) \cap \widetilde{X} \right| = m_i$, but

$\left|B_{R+r}(c) \cap \widetilde{X}\right| = m_i$ and $\left|B_{R+2r}(c) \cap \widetilde{X}\right| = m_i$. If $B_{R+r}(c)$ intersects with another component that it doesn't fully contain, then it must be the case that the component would have at least one more point in $B_{R+2r}(c) \setminus B_{R+r}(c)$ and would be contained in that region (by Lemma C.8) because r is large enough. Since that doesn't happen, there cannot be any component that is partially contained in $B_{R+r}(c)$. Therefore, $B_{R+r}(c)$ is pure.

If return were called because $pure$ were true, then it must have been the case that in iteration $i-1$, before the final update to R, $pure$ must have been set to TRUE because $B_{R+r}(c)$ found new points, but $B_{R+2r}(c)$ couldn't. In all other case, $pure$ is set to FALSE. Since in iteration i, $B_{R+r}(c)$ couldn't find new points either, by the same reasoning as above, $B_R(c)$ must be pure. □

Claim C.16. *In iteration i, either the algorithm exits with a pure ball containing at least $i-1$ components, or at the end of iteration i, $B_R(c)$ contains at least i components.*

Proof. We show this by induction on i, as defined in Algorithm 3.
Base Case: At the beginning of the first iteration, $R = r$. But we know from Lemmata C.9 and C.8 that r is big enough that the mean of the Gaussian from which c has been sampled, along with all other points from the same Gaussian would be contained in $B_r(c)$ because the mean of the Gaussian is at most $\sqrt{\frac{3}{2}\mathrm{tr}(\Sigma)}$ away from c and all other points from the same component are at most $\sqrt{3\mathrm{tr}(\Sigma)}$ away from c. Now, if the algorithm does not exit after the end of the iteration, our base case holds because we have at least one component in the ball at end of the iteration. If the algorithm exited, it must be the case that $B_{2r}(c)$ is pure (from the above claim) and has at least one component.
Inductive Step: Suppose for $i \geq 1$, assume that at the end of iteration i, $B_R(c)$ contains at least i Gaussian components, or that it exits with at least $i-1$ components. If the algorithm exits in iteration i, we are done. So, we assume that it doesn't. In iteration $i+1$, if the algorithm exits and returns a ball, then it is pure (by the claim above and the inductive hypothesis), and has at least i components. So, we're done. Otherwise, either $B_{R+r}(c)$ or $B_{R+2r}(c)$ finds new points. If $B_{R+r}(c)$ finds new points, it is fully containing a component not previously contained within $B_R(c)$, or it is intersecting with new components. In the former case, a new component is added by the end of the iteration, and our claim holds. This holds in the latter case, too, by Lemmata C.8 and C.9 because $B_{R+2r}(c)$ would contain that new component. So, we have contained at least $i+1$ components at the end of iteration $i+1$. If $B_{R+r}(c)$ has no new points, but $B_{R+2r}(c)$ does, then these must be points from a new component. Therefore, $B_{R+3r}(c)$ would completely contain the new component by Lemmata C.8 and C.9. So, we have $i+1$ components again by the end of the iteration. □

From the above argument, we have that either the algorithm exited in the loop, and returned a pure ball, or it had k components at the end of iteration k, in which case, the returned ball is again pure. This gives us the first part of the theorem.

Now, we prove the second part. Let $c \in \widetilde{X}$ be as defined in Algorithm 3. Let Σ be the covariance of the largest Gaussian G in $\widetilde{X}$. Suppose c was sampled from component $G' := \mathcal{N}(\mu', \Sigma')$. If $\mathrm{tr}(\Sigma) = \mathrm{tr}(\Sigma')$, then by the proof above, $r \in \Theta(\sqrt{\mathrm{tr}(\Sigma)})$. Suppose $\mathrm{tr}(\Sigma') < \mathrm{tr}(\Sigma)$. Then in the worst case, the nearest point to c is $\sqrt{3\mathrm{tr}(\Sigma')}$ away from c. It is possible in this case that all points from the largest Gaussian had nearest points $\frac{\sqrt{\mathrm{tr}(\Sigma)}}{4}$ away from them, but $\sqrt{3\mathrm{tr}(\Sigma')} \geq \frac{\sqrt{\mathrm{tr}(\Sigma)}}{4}$. In this case r would still be $16\sqrt{3}$-factor approximation of $\sqrt{\mathrm{tr}(\Sigma')}$. We know that the returned ball $B_R(c)$ contains the component of c. Also, the loop of the algorithm runs at most k times, where in each iteration, R can only increase additively by $3r$. Therefore, the final radius can be at most $(3k+1)r \in O(k\sqrt{\mathrm{tr}(\Sigma')})$. This proves the second part of the theorem. □

Now, we restate a folklore private algorithm for obtaining a projection matrix of the top-k subspace of a dataset via PCA, along with its main result. Here, we just talk about its utility, but focus on the privacy later when we instantiate it for different forms of differential privacy. Given a dataset, it simply truncates all points to within a given range, computes the empirical covariance matrix, then adds random noise to each entry of the matrix that is sampled from a Gaussian distribution scaled according to the appropriate privacy parameters depending on the notion of privacy. In the algorithm, $f_{PCA}(\mathrm{PrivParams})$ denotes an appropriate function of the privacy parameters, which we would set according to the type of differential privacy guarantee we require.

Algorithm 4: DP PCA $\mathrm{PrivPCA}_{\mathrm{PrivParams},\ell}(X, r)$

Input: Private samples $X = (X_1, \ldots, X_{n'}) \in \mathbb{R}^{n' \times d}$. Radius $r \in \mathbb{R}$. Parameters $\mathrm{PrivParams} \subset \mathbb{R}, \ell > 0$.
Output: Projection matrix $\Pi \in \mathbb{R}^{d \times d}$.

Let f_{PCA} be an appropriate function of the elements of $\mathrm{PrivParams}$ to ensure DP.
Set $\sigma_P \leftarrow 2r^2 f_{PCA}(\mathrm{PrivParams})$.
Truncate all points of X to within $B_r(0)$ to get dataset Y.
Let $E \in \mathbb{R}^{d \times d}$, such that for all $i, j \in [d]$ with $i \leq j$, $E_{i,j} \sim \mathcal{N}(0, \sigma_P^2)$ and $E_{j,i} \leftarrow E_{i,j}$.
Let Π be the projection matrix of the top-ℓ subspace of $Z \leftarrow Y^T Y + E$.

Return Π.

Theorem C.17 (Theorem 9 from [DTTZ14])**.** *Let $X, Y, Z, \ell, c, r, \sigma_P$ be quantities as defined in Algorithm 4, C_ℓ be the best rank-ℓ approximation to $Y^T Y$, and $\widehat{C}_\ell$ be the rank-ℓ approximation to Z. Then with probability at least $1 - \beta$,*

$$\|Y^T Y - \widehat{C}_\ell\|_2 \leq \|Y^T Y - C_\ell\|_2 + O\left(\sigma_P \sqrt{d} + \sigma_P \sqrt{\ln(1/\beta)}\right).$$

We get the following corollary using Theorem C.17 and Lemmata A.13 and A.14.

Corollary C.18. *Let $X, Y, Z, \ell, c, r, \sigma_P, \Pi$ be quantities as defined in Algorithm 4, such that X is a clean subset of n samples from a mixture of k Gaussians D satisfying Condition 2, and X contains points from k' components. Suppose $w_{\min}$ is the minimum mixing weight of a components with respect to D. Let $\sigma_{\max}^2$ be the largest directional variance of any component that has points in X, and Σ be the covariance of the Gaussian with largest trace that has points in X. If $r \in O(k\sqrt{\mathrm{tr}(\Sigma)})$, all points in X lie within $B_r(0)$, and*

$$n \geq O\left(\frac{d}{w_{\min}} + d^{1.5} k^2 f_{PCA}(\mathrm{PrivParams}) + \frac{\ln(k/\beta)}{w_{\min}}\right),$$

then with probability at least $1 - O(\beta)$, for each component (μ', Σ', w') that has points in X,

$$\|\mu' - \mu'\Pi\| \leq O\left(\frac{\sigma_{\max}}{\sqrt{w'}}\right).$$

Proof. By construction of our algorithm and our assumption on r, we know that $O\left(\sigma_P \sqrt{d} + \sigma_P \sqrt{\ln(1/\beta)}\right) \in f_{PCA}(\mathrm{PrivParams}) \cdot O\left(d^{1.5} k^2 + dk^2 \sqrt{\ln(1/\beta)}\right) \in f_{PCA}(\mathrm{PrivParams}) \cdot O\left(d^{1.5} k^2\right)$ (since D satisfies Condition 2). Using Lemma A.7 and our bound on n, we know that for each component (μ', Σ', w'), the empirical mean of all the points from that Gaussian in X would be at most $\frac{\sqrt{\|\Sigma'\|_2}}{\sqrt{w'}}$ away from μ'. By the same reasoning, the empirical mean of the same projected points would be at most $\frac{\sqrt{\|\Sigma'\|_2}}{\sqrt{w'}}$ away from $\mu'\Pi$ because the projection of a Gaussian is still a Gaussian. Lemmata A.13 and A.14, and Theorem C.17 together guarantee that the distance between the empirical mean of those points and that of the projected points of that component would be at most $O\left(\frac{\sigma_{\max}}{\sqrt{w'}}\right)$. Applying the triangle inequality gives us the result. □

Now, we mention a result about the sensitivity of $Y^T Y$ in Algorithm 4, which would be used in later subsections.

Lemma C.19. *In Algorithm 4, the ℓ_2 sensitivity of $Y^T Y$ is $2r^2$.*

Proof. Since all points in X are truncated to within $B_r(0)$, for all i, $\|X_i\| \leq r$. Therefore, for a neighbouring dataset X', and corresponding truncated dataset Y', $\|Y^T Y - Y'^T Y'\|_F \leq 2r^2$. □

Next, we describe an algorithm for partitioning the data after the PCA step has been performed. For this, we assume that the data is in low-dimensions (ℓ-dimensional), where Condition 2 may not hold, but there is at least one cluster that is well-separated from the other points, or there is just one cluster in the whole dataset. We provide a private partitioner for private and public data for this regime, which in the first case, returns a partition of the datasets that contains their clean subsets, and in the second case, it returns the same cluster itself. For this, we define the following queries (akin to those in [KSSU19]) that would be used for the public and the private data (respectively) in the algorithm.

$$\begin{aligned}\mathrm{Q}_{\mathrm{Pub}}(X, c, r, t) \coloneqq (|X \cap B_r(c)| \geq t) \wedge \\ (|X \cap (B_{11r}(c) \setminus B_r(c))| = 0) \wedge \\ \left(\left|X \cap (\mathbb{R}^d \setminus B_{11r}(c))\right| \geq t\right)\end{aligned} \tag{3}$$

$$\begin{aligned}\mathrm{Q}_{\mathrm{Priv}}(X, c, r, t, \mathrm{PrivParams}) \coloneqq (\mathrm{PCount}_{\mathrm{PrivParams}}(X \cap B_r(c)) \geq t) \wedge \\ \left(\mathrm{PCount}_{\mathrm{PrivParams}}(X \cap (B_{5r}(c) \setminus B_r(c))) < \frac{t}{320}\right) \wedge \\ \left(\mathrm{PCount}_{\mathrm{PrivParams}}(X \cap (\mathbb{R}^d \setminus B_{5r}(c))) \geq t\right)\end{aligned} \tag{4}$$

Just like in [KSSU19], we define the notion of a *terrific ball*. We say a ball $B_r(c)$ is (γ, t)-*terrific* with respect to some dataset X for $\gamma > 1, t > 0$, if (1) $B_r(c)$ contains at least t points from X, (2) $B_{\gamma r}(c) \setminus B_r(c)$ contains at most $\frac{t}{80}$ points from X, and (3) $\mathbb{R}^d \setminus B_{\gamma r}(c)$ contains at least t points from X. We sometimes omit the parameter t when the context is clear.

Algorithm 5: DP Partitioner $\mathrm{DPLowDimPartitioner}_{n,m,w_{\min},\mathrm{PrivParams}}(\widetilde{Y}, \widetilde{Z}, r_{\max}, r_{\min})$

Input: Private Samples $\widetilde{Z}_1, \ldots, \widetilde{Z}_{n'} \in \mathbb{R}^d$. Public Samples $\widetilde{Y}_1, \ldots, \widetilde{Y}_{m'} \in \mathbb{R}^d$. Parameters $n \geq n', m \geq m', w_{\min} > 0$. Privacy Parameters $\mathrm{PrivParams}$.
Output: A tuple of centre $c \in \mathbb{R}^d$, radius $R \in \mathbb{R}$, or $\perp$.

For $i \leftarrow 0, \ldots, \log(r_{\max}/r_{\min})$
  $r_i \leftarrow \frac{r_{\max}}{2^i}$
  For $j \leftarrow 1, \ldots, m'$
    $c_j \leftarrow \widetilde{Y}_j$
    If $\mathrm{Q}_{\mathrm{Pub}}(\widetilde{Y}, c_j, r_i, \frac{m w_{\min}}{2}) = \textsc{True}$
      If $\mathrm{Q}_{\mathrm{Priv}}(\widetilde{Z}, c_j, 2r_i, \frac{n w_{\min}}{4}, \mathrm{PrivParams}) = \textsc{True}$
        Return $(c = c_j, R = 2r_i)$.
      Else
        Return $\perp$.
Return $\perp$.

Now, we prove the main theorem about the utility of Algorithm 5. In the following, $f_{\mathrm{PCount}}(\mathrm{PrivParams})$ denotes an appropriate function of the privacy parameters based on the accuracy guarantees of PCount (Lemma A.23) depending on the notion of DP being used. We prove the privacy guarantees in later sections.

Theorem C.20. *Let Y be a clean subset of a set of public samples from $D \in \mathcal{G}(d, k)$, Z be a clean subset of private samples from D, Π be a projection matrix to ℓ dimensions, $\widetilde{Y} = Y\Pi$, $\widetilde{Z} = Z\Pi$ be the input to Algorithm 5, $r_{\max}, r_{\min} > 0$, such that for some $j \in [m']$, $\widetilde{Y} \cap B_{r_{\max}}\left(\widetilde{Y}_i\right) = \widetilde{Y}$ and $\widetilde{Z} \cap B_{r_{\max}}\left(\widetilde{Y}_i\right) = \widetilde{Z}$ and $B_{r_{\max}}\left(\widetilde{Y}_i\right)$ is pure with respect to the components of D projected on to the subspace of Π having points in $\widetilde{Y}$ (and $\widetilde{Z}$). Suppose $\sigma^2_{\max}$ is the largest directional variance among the Gaussians that have points in Y. Let that Gaussian be $\mathcal{N}(\mu, \Sigma)$. Suppose,*

$$n \geq O\left(\frac{d \ln(k/\beta)}{w_{\min}} + \frac{\ln(1/\beta)}{w_{\min} \cdot f_{\mathrm{PCount}}(\mathrm{PrivParams})}\right)$$

and

$$m \geq O\left(\frac{\ln(k/\beta)}{w_{\min}}\right).$$

Then we have the following with probability at least $1 - 4\beta$.

1. *Suppose there are at least two Gaussians that have points in Y, and the same Gaussians have points in Z. For any other Gaussian $\mathcal{N}(\mu', \Sigma')$ that has points in Y and Z, suppose for $N = m+n$, $\|\mu\Pi - \mu'\Pi\| \geq \Omega(\sigma_{\max}\sqrt{\ell \ln(N\ell/\beta)})$. Let $r_{\min} \leq \sigma_{\max}\sqrt{2\ell \ln(2N\ell/\beta)}$. Then the ball returned by Algorithm 5 is pure with respect to the components of D projected by Π, which have points in $\widetilde{Y}$, such that $\mathbb{R}^d \setminus B_R(c)$ only contains components from D projected by Π that have points in $\widetilde{Y}$ (alternatively, $\widetilde{Z}$), and so does $B_R(c)$.*

2. *Suppose there is only one Gaussian that has points in Y and Z. Then the algorithm returns $\perp$.*

Proof. We first prove the first part of the theorem. In this case, we know that there exists a ball of some radius $r > 0$ and some centre $c \in \widetilde{Y}$, such that $B_r(c)$ has the required properties. We prove the existence of such a ball first. Let $c \in \widetilde{Y}$ be a point, such that the corresponding Gaussian to c in d dimensions is $\mathcal{N}(\mu, \Sigma)$ (with $\|\Sigma\| = \sigma_{\max}^2$). For any Gaussian $\mathcal{N}(\mu', \Sigma')$ that has points in Y, in the subspace of Π, $\|\mu\Pi - \mu'\Pi\| \geq 103\sigma_{\max}\sqrt{\ell \ln(N\ell/\beta)}$. Now, we know that for any Gaussian $\mathcal{N}(\mu'', \Sigma'')$, in this subspace, for any $y \sim \mathcal{N}(\mu'', \Sigma'')$ that lies in Y and any $z \sim \mathcal{N}(\mu'', \Sigma'')$ that lies in Z, $\|y\Pi - \mu''\Pi\|, \|z\Pi - \mu''\Pi\| \leq \sqrt{\|\Pi\Sigma''\Pi\|_2}\sqrt{2\ell \ln(2N\ell/\beta)}$ from Lemma C.12. The triangle inequality implies that for any $y, z \sim \mathcal{N}(\mu, \Sigma)$ with $y \in Y$ and $z \in Z$, and $a, b \sim \mathcal{N}(\mu', \Sigma')$ with $a \in Y$ and $b \in Z$, $\|y\Pi - a\Pi\|, \|z\Pi - b\Pi\| \geq 100\sigma_{\max}^2\sqrt{\ell \ln(2N\ell/\beta)}$. Given that $B_{2\sigma_{\max}\sqrt{2\ln(2N\ell/\beta)}}(c)$ contains all points in $Y\Pi$ and $Z\Pi$, whose Gaussian in high dimensions is $\mathcal{N}(\mu, \Sigma)$, and contains the Gaussian in low dimensions, this ball satisfies the required properties.

Now, let's say the algorithm returns a ball $B = B_{2r_i}(c_j)$. We know that in this case, because the radius r_i decreases when i increases, $r_i \geq 2\sigma_{\max}\sqrt{2\ell \ln(2N\ell/\beta)}$. Because r_i is bigger than the largest possible diameter of any Gaussian component in the ℓ-dimensional subspace, $B_{r_i}(c_j)$ must be, at least, containing the component of c_j in that subspace. We know that the shell $S = B_{11r_i}(c_j) \setminus B_{r_i}(c_j)$ has no points from any component in $\widetilde{Y}$ (by construction). For any component inside $B_{r_i}(c_j)$ (either partially or completely), the distance between the mean of any such component from the mean of any other component that is completely outside $B_{r_i}(c_j)$ has to be at least $10r_i - 2 \times \frac{r_i}{2} = 9r_i$ (because r_i is at least twice the maximum distance from the mean of a Gaussian to any of its points in that subspace). If there a component, whose mean lies in the shell $T = B_{\frac{21r_i}{2}}(c_j) \setminus B_{\frac{3r_i}{2}}(c_j)$, it would be completely contained inside S, and we would see points of $\widetilde{Y}$ in S, which would be a contradiction. Therefore, the means of all components that lie completely outside $B_{r_i}(c_j)$ must be within $\mathbb{R}^d \setminus B_{\frac{21r_i}{2}}(c_j)$, which means that they would be entirely contained within $\mathbb{R}^d \setminus B_{10r_i}(c_j)$ (because of Lemma C.12). Therefore, B cannot have any component that lies completely outside $B_{r_i}(c_j)$. By similar reasoning, the mean of any component that partially intersects with $B_{r_i}(c_j)$ has to be within $B_{\frac{3r_i}{2}}(c_j)$. Therefore, by Lemma C.12 again, that component would be contained within $B_{2r_i}(c_j)$. Therefore, the ball B is pure. By the construction of the query $\mathrm{Q}_{\mathrm{Priv}}$, our sample complexity (of n), and the guarantees of PCount, with probability at least $1 - 3\beta$, we know that the noise in each call to PCount cannot be more than $\frac{nw_{\min}}{1280}$ (Lemma A.23). We also know that the number of points from each component is at least $\frac{nw_{\min}}{2}$ (from Lemma C.7). Since $B_{2r_i}(c_j)$ is pure, it must have at least $\frac{nw_{\min}}{2}$ points in it, so the noisy answer via PCount would be at least $\frac{nw_{\min}}{4}$. Similarly, since there is no Gaussian component even partially contained within $B_{10r_i}(c_j) \setminus B_{2r_i}(c_j)$, the noisy answer will be less than $\frac{nw_{\min}}{1280}$. Finally, since there is at least one component in $\mathbb{R}^d \setminus B_{10r_i}(c_j)$, the noisy answer of PCount would be over $\frac{nw_{\min}}{4}$. So, $\mathrm{Q}_{\mathrm{Priv}}$ would return TRUE. This proves the first part of the theorem.

Now, we prove the second part of the theorem. It is sufficient to show that there exists no 5-terrific ball in the private dataset $\widetilde{Z}$ when all the data is coming from a single Gaussian. First, we claim that if there is a 5-terrific ball in a dataset X, then there exists a 3-terrific ball, too. The argument is simple. Suppose $B_r(c)$ is a 5-terrific ball, then it is easy to see that $B_{r'}(c)$ for $r' = \frac{5r}{3}$ is a 3-terrific ball. This is because $B_{r'}(c)$ would contain more points that $B_r(c)$, hence, $B_{3r'}(c) \setminus B_{r'}(c)$ would contain fewer points than $B_{5r}(c) \setminus B_r(c)$. Since $\mathbb{R}^d \setminus B_{5r}(c) = \mathbb{R}^d \setminus B_{3r'}(c)$, the final constraint would hold, as well. Hence, the claim. So now, it is sufficient to show that there exists no 3-terrific ball with respect to $\widetilde{Z}$.

We prove this via contradiction. We know that due to our sample complexity (bound on n), the noise in each call to PCount is at most $\frac{nw_{\min}}{1280}$. Therefore, for a query $\mathrm{Q}_{\mathrm{Priv}}(\widetilde{Z}, c', r', \frac{nw_{\min}}{4}, \mathrm{PrivParams})$ to return a ball $B_{r'}(c')$, $B_{r'}(c')$ and $\mathbb{R}^d \setminus B_{5r'}(c')$ must contain at least $\frac{nw_{\min}}{8}$ points from $\widetilde{Z}$, and $B_{5r'}(c') \setminus B_{r'}(c')$ must contain at most $\frac{nw_{\min}}{640}$ points from $\widetilde{Z}$. This implies the existence of a 3-terrific ball of radius r with respect to $\widetilde{Z}$. This means that there exists a unit vector v such that on projecting all the data on to v, we would have an three adjacent intervals along the vector I_1, I_2, I_3 of length $2r$ each along the direction of v, such that I_1 and I_3 together contain at most $\frac{nw_{\min}}{640}$ points, I_2 contains at least $\frac{nw_{\min}}{8}$ points, and the rest of the line contains at least $\frac{nw_{\min}}{8}$ points. Suppose the mixing weight of the Gaussian in question is w_i. Lemma C.7 shows that the number of points in $\widetilde{Z}$ is Cnw_i, where $\frac{1}{2} \leq C \leq \frac{3}{2}$. A straightforward application of Chernoff bound (Lemma A.1) and the union bound over all $2^{O(d)}$ unit vectors in a cover of the unit sphere in $\mathbb{R}^d$, along with our sample complexity, implies that the probability mass contained in I_2 and the part of line excluding I_1, I_2, and I_3 (which we call, I_4) is at least $\frac{w_{\min}}{8Cw_i}$ each, and the mass contained in $I_1 \cup I_3$ is strictly less than $\frac{w_{\min}}{160Cw_i}$. We show that this is impossible when the underlying distribution along v is a (one-dimensional) Gaussian.

Suppose the variance along v of the said Gaussian is σ^2. WLOG, we will assume that its mean is 0. Note that for the said condition about the probability mass to be true, I_2 must contain the mean. This is because of the symmetry about the mean and unimodality of one-dimensional Gaussians – if I_2 does not contain the mean, then either I_1 or I_3 would be closer to the mean, and would contain more mass than I_2 because all three of these intervals are of the same length. Next, we show that the total mass contained within I_1 and I_3 is minimised when the mean is at the centre of I_2. After that, it would be sufficient to show that in this configuration, the probability mass contained within I_1 and I_3 together is at least $\frac{w_{\min}}{160Cw_i}$ when the mass contained in I_2 and I_4 each is at least $\frac{w_{\min}}{8Cw_i}$. Let $0 \leq r_1 \leq r$ and $r_2 = 2r - r_1$, such that one end point of I_2 is r_1 away from the mean, and the other end point is r_2 away from the mean. Then the probability mass contained within $I_1 \cup I_3$ is

$$\frac{1}{\sqrt{2\pi}\sigma} \int_{r_1}^{r_1+2r} e^{-\frac{t^2}{2\sigma^2}} dt + \frac{1}{\sqrt{2\pi}\sigma} \int_{r_2}^{r_2+2r} e^{-\frac{t^2}{2\sigma^2}} dt = \frac{1}{\sqrt{2\pi}} \int_{r_1/\sigma}^{(r_1+2r)/\sigma} e^{-\frac{t^2}{2}} dt + \frac{1}{\sqrt{2\pi}} \int_{(2r-r_1)/\sigma}^{(4r-r_1)/\sigma} e^{-\frac{t^2}{2}} dt.$$

Define $f(r_1) = \frac{1}{\sqrt{2\pi}} \int_{r_1/\sigma}^{(r_1+2r)/\sigma} e^{-\frac{t^2}{2}} dt + \frac{1}{\sqrt{2\pi}} \int_{(2r-r_1)/\sigma}^{(4r-r_1)/\sigma} e^{-\frac{t^2}{2}} dt$. Then we have the following.

$$\frac{df(r_1)}{dr_1} = \frac{1}{\sqrt{2\pi}} \left[\frac{1}{\sigma} \cdot e^{-\frac{(r_1+2r)^2}{2\sigma^2}} - \frac{1}{\sigma} \cdot e^{-\frac{r_1^2}{2\sigma^2}} - \frac{1}{\sigma} \cdot e^{-\frac{(4r-r_1)^2}{2\sigma^2}} + \frac{1}{\sigma} \cdot e^{-\frac{(2r-r_1)^2}{2\sigma^2}} \right]$$

Within the interval $[0, r]$, setting the above to 0, we have $r_1 = r$. It can be checked that for all $r_1 \in [0, r]$, $\frac{d^2 f(r_1)}{dr_1^2} > 0$. This means that within $[0, r]$, f is minimised at $r_1 = r$. Therefore, for the rest of the proof, we would concentrate on lower bounding the quantity $q(r) = \frac{2}{\sqrt{2\pi}} \int_{r/\sigma}^{3r/\sigma} e^{-\frac{t^2}{2}} dt$. We define $p(r) = \frac{2}{\sqrt{2\pi}} \int_{0}^{r/\sigma} e^{-\frac{t^2}{2}} dt$ to be the mass of I_2 in this regime, and $\lambda(r) = \frac{2}{\sqrt{2\pi}} \int_{3r/\sigma}^{\infty} e^{-\frac{t^2}{2}} dt$ to be the mass of I_4. We consider three cases.

Case 1: $r^2 = \sigma^2 + \tau, 0 \leq \tau \leq \sigma^2$. Noting that $e^{-x+0.25} \geq e^{-x^2}$ for $x \in [0, 1]$ and $\frac{1}{\sqrt{2}} \leq \frac{r}{\sqrt{2}\sigma} \leq 1$, we have the following.

$$\frac{w_{\min}}{8Cw_i} \leq p(r) = \frac{2}{\sqrt{\pi}} \int_{0}^{r/\sqrt{2}\sigma} e^{-s^2} ds \leq \frac{2e^{\frac{1}{4}}}{\sqrt{\pi}} \int_{0}^{r/\sqrt{2}\sigma} e^{-s} ds = \frac{2e^{\frac{1}{4}}}{\sqrt{\pi}} \cdot \left(1 - e^{-\frac{r}{\sqrt{2}\sigma}}\right)$$

Now, we compute $q(r)$.

$$q(r) = \frac{2}{\sqrt{\pi}} \int_{r/\sqrt{2}\sigma}^{3r/\sqrt{2}\sigma} e^{-s^2} ds$$

$$
\begin{aligned}
&= \frac{2}{\sqrt{\pi}} \left[\int_{r/\sqrt{2}\sigma}^{1} e^{-s^2} ds + \int_{1}^{3/\sqrt{2}} e^{-s^2} ds + \int_{3/\sqrt{2}}^{3r/\sqrt{2}\sigma} e^{-s^2} ds \right] \\
&\geq \frac{2}{\sqrt{\pi}} \left[\int_{r/\sqrt{2}\sigma}^{1} e^{-s} ds + \int_{1}^{3/\sqrt{2}} e^{-s^2} ds + \int_{3/\sqrt{2}}^{3r/\sqrt{2}\sigma} e^{-\frac{3rs}{\sqrt{2}\sigma}} ds \right] \\
&= \frac{2}{\sqrt{\pi}} \left[e^{-\frac{r}{\sqrt{2}\sigma}} - \frac{1}{e} + \int_{1}^{3/\sqrt{2}} e^{-s^2} ds + \frac{\sqrt{2}\sigma}{3r} \cdot e^{-\frac{9r}{2\sigma}} - \frac{\sqrt{2}\sigma}{3r} \cdot e^{-\frac{9r^2}{2\sigma^2}} \right] \\
&\geq \frac{2}{\sqrt{pi}} \left[\int_{1}^{3/\sqrt{2}} e^{-s^2} ds - \frac{1}{e} + e^{-\frac{r}{\sqrt{2}\sigma}} + \frac{1}{3} \cdot e^{-\frac{9r}{2\sigma}} - \frac{1}{3} \cdot e^{-\frac{9r^2}{2\sigma^2}} \right] \\
&> \frac{2e^{\frac{1}{4}}}{20\sqrt{\pi}} \cdot \left(1 - e^{-\frac{r}{\sqrt{2}\sigma}}\right) \\
&\geq \frac{w_{\min}}{160Cw_i}
\end{aligned}
$$

We have a contradiction. Thus, in this case, the probability mass in $I_1 \cup I_3$ has to be at least $\frac{w_{\min}}{160Cw_i}$.

Case 2: $r^2 = \sigma^2 + \tau, \tau \geq \sigma^2$. We have the following upper bound on $\lambda(r)$ using Lemma A.4.

$$
\lambda(r) = \frac{2}{\sqrt{2\pi}} \int_{3r/\sigma}^{\infty} e^{-\frac{t^2}{2}} dt \leq 2e^{-\frac{9r^2}{2\sigma^2}}
$$

We know that $\lambda(r) \geq \frac{w_{\min}}{8Cw_i}$. This implies that $\frac{r^2}{\sigma^2} \leq \frac{2}{9} \ln\left(\frac{16Cw_i}{w_{\min}}\right)$. This is also equivalent to $\frac{w_{\min}}{16Cw_i} \leq \frac{1}{e^9}$. Lemma A.6 gives us the following lower bound on $1 - p(r)$.

$$
1 - p(r) = \frac{2}{\sqrt{2\pi}} \int_{r/\sigma}^{\infty} e^{-\frac{t^2}{2}} dt \geq 0.06 e^{-\frac{3r^2}{2\sigma^2} + \frac{3}{2}} = (0.06) e^{\frac{3}{2}} e^{-\frac{3r^2}{2\sigma^2}}
$$

This gives us the following lower bound on $q(r)$.

$$
\begin{aligned}
q(r) &= \frac{2}{\sqrt{2\pi}} \int_{r/\sigma}^{\infty} e^{-\frac{t^2}{2}} dt - \frac{2}{\sqrt{2\pi}} \int_{3r/\sigma}^{\infty} e^{-\frac{t^2}{2}} dt \\
&\geq (0.06) e^{\frac{3}{2}} e^{-\frac{3r^2}{2}} - 2e^{-\frac{9r^2}{2\sigma^2}} \\
&\geq (0.03) e^{\frac{3}{2}} e^{-\frac{3r^2}{2}} && \text{(For } r^2 \geq 2\sigma^2\text{.)} \\
&\geq (0.03) e^{\frac{3}{2}} e^{-\frac{3}{2} \cdot \frac{2}{9} \ln\left(\frac{16Cw_i}{w_{\min}}\right)} \\
&= (0.03) e^{\frac{3}{2}} \cdot \left(\frac{w_{\min}}{16Cw_i}\right)^{\frac{1}{3}} \\
&\geq \frac{w_{\min}}{160Cw_i} && \text{(In the regime } \tfrac{w_{\min}}{16Cw_i} \leq \tfrac{1}{e^9}\text{.)}
\end{aligned}
$$

This gives us a contradiction for the second case, as well.

Case 3: $r^2 = \sigma^2 - \tau, 0 \leq \tau < \sigma^2$. This final case has two sub-cases to analyse: $9r^2 \leq 2\sigma^2$ and $9r^2 > 2\sigma^2$. In the first instance, we have the following bounds on $p(r)$.

$$
\frac{w_{\min}}{8Cw_i} \leq p(r) = \frac{2}{\sqrt{2\pi}} \int_{0}^{r/\sigma} e^{-\frac{t^2}{2}} dt \leq \frac{2r}{\sqrt{2\pi}\sigma}
$$

Next, we lower bound $p(3r)$.

$$p(3r) = \frac{2}{\sqrt{2\pi}} \int_0^{3r/\sigma} e^{-\frac{t^2}{2}} dt \geq \frac{2}{\sqrt{2\pi}} \cdot \frac{3r}{\sigma} \cdot e^{-\frac{9r^2}{2\sigma^2}} \geq \frac{6r}{e\sqrt{2\pi}\sigma}$$

This gives us the following lower bound on $q(r)$.

$$\begin{aligned} q(r) &= \frac{2}{\sqrt{2\pi}} \int_0^{3r/\sigma} e^{-\frac{t^2}{2}} dt - \frac{2}{\sqrt{2\pi}} \int_0^{r/\sigma} e^{-\frac{t^2}{2}} dt \\ &\geq \frac{6r}{e\sqrt{2\pi}\sigma} - \frac{2r}{\sqrt{2\pi}\sigma} \\ &> \frac{r}{5\sqrt{2\pi}\sigma} \\ &> \frac{w_{\min}}{160Cw_i} \end{aligned}$$

This shows that this situation is impossible. So, we move on to the final instance, that is, when $\frac{2\sigma^2}{9} < r^2 < \sigma^2$. Just as we did in Case 1, we bound $p(r)$. We get the following.

$$\frac{w_{\min}}{8Cw_i} \leq p(r) \leq \frac{2e^{\frac{1}{4}}}{\sqrt{\pi}} \cdot \left(1 - e^{-\frac{r}{\sqrt{2}\sigma}}\right)$$

We lower bound $p(3r)$ again.

$$\begin{aligned} p(3r) &= \frac{2}{\sqrt{\pi}} \int_0^1 e^{-s^2} ds + \frac{2}{\sqrt{\pi}} \int_1^{3r/\sqrt{2}\sigma} e^{-s^2} ds \\ &\geq \frac{2}{\sqrt{\pi}} \int_0^1 e^{-s^2} ds + \frac{2}{\sqrt{\pi}} \int_0^{3r/\sqrt{2}\sigma} e^{-\frac{3rs}{\sqrt{2}\sigma}} ds \\ &= \frac{2}{\sqrt{\pi}} \int_0^1 e^{-s^2} ds + \frac{2}{\sqrt{\pi}} \cdot \frac{\sqrt{2}\sigma}{3r} \left[e^{-\frac{3r}{\sqrt{2}\sigma}} - e^{-\frac{9r^2}{2\sigma^2}}\right] \\ &\geq \frac{2}{\sqrt{\pi}} \int_0^1 e^{-s^2} ds + \frac{2\sqrt{2}}{3\sqrt{\pi}} \left[e^{-\frac{3r}{\sqrt{2}\sigma}} - e^{-\frac{9r^2}{2\sigma^2}}\right] \end{aligned}$$

Now, we lower bound $q(r)$.

$$\begin{aligned} q(r) &= \frac{2}{\sqrt{2\pi}} \int_0^{3r/\sigma} e^{-\frac{t^2}{2}} dt - \frac{2}{\sqrt{2\pi}} \int_0^{r/\sigma} e^{-\frac{t^2}{2}} dt \\ &\geq \frac{2}{\sqrt{\pi}} \int_0^1 e^{-s^2} ds + \frac{2\sqrt{2}}{3\sqrt{\pi}} \left[e^{-\frac{3r}{\sqrt{2}\sigma}} - e^{-\frac{9r^2}{2\sigma^2}}\right] - \frac{2e^{\frac{1}{4}}}{\sqrt{\pi}} \cdot \left(1 - e^{-\frac{r}{\sqrt{2}\sigma}}\right) \\ &\geq \frac{1}{20} \cdot \frac{2e^{\frac{1}{4}}}{\sqrt{\pi}} \cdot \left(1 - e^{-\frac{r}{\sqrt{2}\sigma}}\right) && \text{(For } \tfrac{2\sigma^2}{9} \leq r^2 \leq \sigma^2\text{.)} \\ &\geq \frac{w_{\min}}{160Cw_i} \end{aligned}$$

This gives us the final contradiction, hence, completing the proof for Case 3.

Therefore, there cannot be such intervals on the line of v. This means that there cannot be any such direction, where such intervals exist, which means that there cannot be a 3-terrific ball with respect to $\widetilde{Z}$, implying that the algorithm could not have found a 5-terrific ball with respect to $\widetilde{Z}$. By the construction of the algorithm, the output would be $\perp$. This completes the proof of the theorem. $\square$

We now state the main private algorithm for clustering points from a Gaussian mixture (Algorithm 6) that utilises the public data available to it, along with a theorem describing its utility guarantees. In the theorem, f'_{PCA} and f'_{PCount} denote appropriate functions of the privacy parameters in terms of the previously mentioned functions f_{PCA} and f_{PCount}, respectively. Note that the output of the algorithm itself is not private, but the intermediate steps involved are. Since its output will not be released to public, and will just be utilised by our main private estimation algorithm, there will be no violation of privacy in the end. This algorithm is a general private framework, which could be instantiated for approximate DP and zCDP by choosing the right parameters and privacy primitives, for example, it would use approximate DP PCA algorithm for the former, and zCDP PCA algorithm for the latter. It first uses Algorithm 3 on the public data to find a supercluster that tightly bounds a group of clusters. Since the private data is sampled from the same distribution as the public data, it uses that supercluster to isolate the points in the private dataset corresponding to the components contained within the supercluster. Next, it uses private PCA to project the data on to the top-k subspace, and then works within that subspace to either determine if the data is coming from a single Gaussian, or to partition the dataset into clean subsets in case it is coming from at least two Gaussians. In the former case, it adds the set of private points in the original d-dimensional space to the output (which contains sets of points). In the latter case, it just proceeds to work on the partitions independently.

Theorem C.21. *There exists an algorithm (Algorithm 6) that takes n private samples X, and m public samples from $D \in \mathcal{G}(d, k, s)$ satisfying Condition 2, and outputs a partition C of X, such that given Conditions C.1 to C.6 hold, and*

$$s \geq \Omega(\sqrt{k \ln((n+m)k/\beta)}),$$

$$n \geq O\left(\frac{d \ln(k/\beta)}{w_{\min}} + d^{1.5}k^{2.5} f'_{PCA}(\text{PrivParams}) + \frac{\sqrt{k}\ln(k/\beta)}{w_{\min} \cdot f'_{\text{PCount}}(\text{PrivParams})}\right),$$

and

$$m \geq O\left(\frac{\ln(k/\beta)}{w_{\min}}\right),$$

then with probability at least $1 - O(\beta)$, $|C| = k$ and each $S \in C$ is clean and non-empty.

Proof. It is enough to show that $\forall i \geq 0$, at the end of the i-th iteration, (1) a new Gaussian component is isolated in C and *count* equals the number of isolated components in C, (2) or the private dataset X is further partitioned into non-empty, clean subsets, and the public dataset $\widetilde{X}$ is further partitioned into clean subsets, such that $|Q_{\text{Pub}}| = |Q_{\text{Priv}}|$ and for each $j \in [|Q_{\text{Pub}}|]$, $Q_{\text{Pub}}[j]$ and $Q_{\text{Priv}}[j]$ have points from the same Gaussian components. We can prove this via induction on i.

Base Case: The end of the 0-th iteration essentially means the actual start of the loop. In this case, we know that Q_{Pub} and Q_{Priv} are clean subsets of $\widetilde{X}$ and X, respectively. Therefore, the claim trivially holds in this case.

Inductive Step: We assume for all iterations, up to and including some $i \geq 0$, the claim holds. Then we show that it holds for iteration $i+1$, as well. By the inductive hypothesis, we know that Z and Y are clean subsets, and contain points from the same components. When the superclustering algorithm (Algorithm 3) is called, it finds a pure ball that will either contain the whole dataset Y (hence, Z because the ball is pure), or it partitions Y into clean subsets (Theorem C.14). Since the components from which the points in Y and Z come from are the same, $B_{R_i}(c_i)$ would contain points from the same components from Z as in Y (because the deterministic regularity conditions hold). So, adding $Z \setminus Z^i$ to Q_{Priv} and $Y \setminus Y^i$ to Q_{Pub} ensures that the partition of Z added to the back of Q_{Priv} is clean and contains the points from the same components as those that have points in $Y \setminus Y^i$ (which itself gets added to the back of Q_{Pub}), which preserves the ordering of the set of components in the two queues.

If there is just one component that has points in Y^i (hence, in Z^i), then after projecting the data points on to the subspace of Π_i, and feeding the projected datasets to Algorithm 5, we will get $\perp$ (by the guarantees from Theorem C.20) with probability at least $1 - \frac{\beta}{2k}$. Because Z^i is a clean subset, C now contains one more clean subset of X, and we have isolated a new component, and updated the value of *count* to reflect the change.

Algorithm 6: DP GMM Hard Clustering $\mathrm{DPHardClustering}_{\mathrm{PrivParams},\beta,k,w_{\min}}(\widetilde{X}, X)$

Input: Private samples $X = (X_1, \ldots, X_n) \in \mathbb{R}^{n\times d}$. Public samples $\widetilde{X} = (\widetilde{X}_1, \ldots, \widetilde{X}_m) \in \mathbb{R}^{m\times d}$. Parameters $\mathrm{PrivParams} \subset \mathbb{R}, \beta, k, w_{\min} > 0$.
Output: Set $C \subset \mathcal{P}(X)$.

Let $Y \leftarrow \widetilde{X}$ and $Z \leftarrow X$.
Let $C \leftarrow \emptyset$.
Let Q_{Priv} and Q_{Pub} be queues of sets of points.
Add set Z to Q_{Priv}, and set Y to Q_{Pub}.
$\mathrm{PrivParams}'$ be the set of privacy parameters modified as per composition based on $\mathrm{PrivParams}$.
Set $count \leftarrow 0$ and $i \leftarrow 1$.

While $count < k$ *and* $i \leq 2k$
  Pop Q_{Priv} to get Z, and Q_{Pub} to get Y.

  `// Run superclustering algorithm on the public dataset.`
  Set $(c_i, R_i) \leftarrow \mathrm{SC}_{k,m}(Y)$.

  `// Partition both datasets on the basis of the supercluster.`
  $Y^i \leftarrow Y \cap B_{R_i}(c_i)$ and $Z^i \leftarrow Z \cap B_{R_i}(c_i)$.
  Add $Z \setminus Z^i$ to Q_{Priv}, and $Y \setminus Y^i$ to Q_{Pub}.

  `// Private PCA: project points of both datasets onto returned subspace.`
  Let $M^i \in \mathbb{R}^{|Z^i|\times d}$, such that each row $M^i_j \leftarrow c_i$.
  $\Pi_i \leftarrow \mathrm{PrivPCA}_{\mathrm{PrivParams}',k}(Z^i - M^i, R_i)$.
  $Y' \leftarrow (Y^i - M^i)\Pi_i$ and $Z' \leftarrow (Z^i - M^i)\Pi_i$.

  `// If there is only one Gaussian, add it to` C`, otherwise further partition the datasets.`
  $B \leftarrow \mathrm{DPLowDimPartitioner}_{n,m,w_{\min},\mathrm{PrivParams}'}(Z', Y', R_i, \frac{R_i}{\sqrt{d}})$.
  If $B = \perp$
    $C \leftarrow C \cup \{Z^i\}$.
    $count \leftarrow count + 1$.
  Else
    B is an ordered pair (c'_i, r'_i).
    $S' \leftarrow B_{r'_i}(c') \cap Y'$ and $T' \leftarrow B_{r'_i}(c') \cap Z'$.
    Let S be points in Y^i corresponding to S', and T be points in Z^i corresponding to T'.
    Add S to Q_{Pub} and T to Q_{Priv}.
    Add $Y^i \setminus S$ to Q_{Pub} and $Z^i \setminus T$ to Q_{Priv}.

  $i \leftarrow i + 1$

Return C.

Suppose there are at least two components that have points in Y^i (hence, in Z^i). Then we know from Theorem C.14, that if Σ is the covariance having the largest trace among all components that have points in Y^i (hence, in Z^i), then $R_i \in O(k\sqrt{\mathrm{tr}(\Sigma)})$. Let the largest directional variance among the Gaussians that have points in Z^i (hence, in Y^i) be $\sigma^2_{\max}$, its mean be μ, and its mixing weight be w. From the guarantees in Corollary C.18, we know that $\|\mu - \mu\Pi_i\| \leq O\left(\frac{\sigma_{\max}}{\sqrt{w}}\right)$ with probability at least $1 - \frac{\beta}{2k}$. Because of the separation condition, applying the triangle inequality, we know that the distance between $\mu\Pi_i$ and the projected mean of any other Gaussian having points in $Z^i\Pi$ is at least $\Omega(\sigma_{\max}\sqrt{k\ln((n+m)k)/\beta})$. Then from the guarantees of Theorem C.20, we have that a ball is returned, and it is pure with respect to the components of the mixture projected by Π_i that have points in $Z^i\Pi$. Therefore, subsets S' and T' are clean, and so are S and T. This implies that $Y^i \setminus S$ and $Z^i \setminus T$ are clean, too. By adding them in order to Q_{Pub} and Q_{Priv} respectively, we partition

the datasets $\widetilde{X}$ and X further into clean subsets, and preserve the ordering of the components in the respective queues. This proves the claim.

Note that before the PCA step, we recentre the points in both Y^i and Z^i for the ease of analysis later when we provide guarantees for privacy. It does not affect correctness of any step of the algorithm because it is just recentering of data.

Now, we just have to show that the number of iterations we allow is enough with high probability. Because we have k components in the mixture, we can only partition into clean subsets at most $k-1$ times. For each component, the loop will run at most one time. So, $2k-1$ iterations are enough to capture all the components, and $|C| = k$. Taking the union bound over all possible $2k-1$ iterations, we have the required result. □

We finally state the private algorithm (Algorithm 7) for estimating the parameters of mixtures of Gaussians, along with a theorem highlighting its utility guarantees. Just like the DP clustering algorithm above, it is a private framework that could instantiated for approximate DP and zCDP settings by using the corresponding private algorithms. It calls Algorithm 6 to cluster the points according to their respective components, then calls private estimators to estimate the parameters of the respective components. It finally uses PCount to estimate their mixing weights based on the private data.

Algorithm 7: DP GMM Hard Case Estimator $\mathrm{DPHardEstimator}_{\mathrm{PrivParams},\alpha,\beta,k,w_{\min}}(\widetilde{X}, X)$

Input: Private samples $X = (X_1, \ldots, X_n) \in \mathbb{R}^{n\times d}$. Public samples $\widetilde{X} = (\widetilde{X}_1, \ldots, \widetilde{X}_m) \in \mathbb{R}^{m\times d}$. Parameters $\mathrm{PrivParams} \subset \mathbb{R}, \beta, k, w_{\min} > 0$.

Output: Set $\widehat{D} = \{(\widehat{\mu}_1, \widehat{\Sigma}_1, \widehat{w}_1), \ldots, (\widehat{\mu}_k, \widehat{\Sigma}_k, \widehat{w}_k)\}$.

`// Clustering.`

Set $C \leftarrow \mathrm{DPHardClustering}_{\mathrm{PrivParams},\beta,k,w_{\min}}(X, \widetilde{X})$.

`// Parameter estimation.`

$\widehat{D} \leftarrow \emptyset$.

For $i \leftarrow 1, \ldots, k$

  Set $(\widehat{\mu}_i, \widehat{\Sigma}_i) \leftarrow \mathrm{DPGaussianEstimator}_{\mathrm{PrivParams},\frac{\beta}{k}}(C_i)$.

  Set $\widehat{w}_i \leftarrow \max\left\{\frac{1}{n} \cdot \mathrm{PCount}_{\mathrm{PrivParams}}(|C_i|), \frac{\alpha}{2k}\right\}$.

  $\widehat{D} \leftarrow \widehat{D} \cup \{(\widehat{\mu}_i, \widehat{\Sigma}_i, \widehat{w}_i)\}$.

Return $\widehat{D}$.

Theorem C.22. *For all $\alpha, \beta > 0$ and sets of privacy parameters $\mathrm{PrivParams}$, there exists an algorithm (Algorithm 7) that takes n private samples and m public samples from $D \in \mathcal{G}(d, k, s)$, such that if $D = \{(\mu_1, \Sigma_1, w_1), \ldots, (\mu_k, \Sigma_k, w_k)\}$ satisfies Condition 2, and*

$$s \in \Omega(\sqrt{k \ln((n+m)k/\beta)})$$

$$n \in O\left(\frac{n_{GE}\ln(k/\beta)}{w_{\min}} + \frac{d\ln(k/\beta)}{w_{\min}} + d^{1.5}k^{2.5} f'_{PCA}(\mathrm{PrivParams}) + \frac{\sqrt{k}\ln(k/\beta)}{w_{\min} \cdot f'_{\mathrm{PCount}}(\mathrm{PrivParams})}\right)$$

$$m \in O\left(\frac{\ln(k/\beta)}{w_{\min}}\right),$$

where n_{GE} is the sample complexity of privately learning a Gaussian using Lemma A.24 according to the type of DP required, then it (α, β)-learns D.

Proof. By our sample complexity bound and Theorem C.21, with probability at least $1 - \frac{\beta}{3}$, for each $i \in [k]$, C_i is clean with respect to X. Therefore, each C_i can be used to learn an independent component.

Next, by the guarantees of Lemma A.24, for each $i \in [k]$, $\mathrm{d_{TV}}(\mathcal{N}(\widehat{\mu}_i, \widehat{\Sigma}_i), \mathcal{N}(\mu_i, \Sigma_i)) \leq \alpha$ with probability at least $1 - \frac{\beta}{3k}$.

Finally, using Lemmata C.7 and A.23, along with our sample complexity, we know that the error due to each call to PCount is at most $\frac{\alpha}{2k}$ with probability at least $1 - \frac{\beta}{3k}$, and that the error due to sampling is at most $\frac{\alpha}{2k}$. We apply the triangle inequality to get the final error of $\frac{\alpha}{k}$ for each mixing weight.

Applying the union bound over all failure events (including the failures of Conditions C.1 to C.6), we have the desired result. □

C.2.1 (ε, δ)-DP Algorithm

Here, we describe our results under approximate DP constraints, and instantiate Algorithms 4, 6, and 7 for this version of DP. We will not restate the entire algorithms, but just describe how PrivParams and the sample complexity in these different algorithms change. We now state the main theorem of our section.

Theorem C.23. *For all $\alpha, \beta, \varepsilon, \delta > 0$, there exists an $(O(\varepsilon), O(\delta))$-DP algorithm $\mathcal{M}$ that takes n private samples and m public samples from $D \in \mathcal{G}(d, k, s)$, and is private with respect to the private samples, such that if $D = \{(\mu_1, \Sigma_1, w_1), \ldots, (\mu_k, \Sigma_k, w_k)\}$ satisfies Condition 2, and*

$$s \in \Omega(\sqrt{k \ln((n+m)k/\beta)})$$

$$n \in O\left(\frac{d^2 \ln\left(\frac{k}{\beta}\right)}{w_{\min}\alpha^2} + \frac{(d^2 \log(k/\delta) + d \log^{1.5}(k/\delta)) \cdot \text{polylog}\left(d, \frac{k}{\beta}, \frac{1}{\alpha}, \frac{\sqrt{k}}{\varepsilon}, \ln\left(\frac{k}{\delta}\right)\right)}{w_{\min}\alpha\varepsilon} + \frac{d^{1.5}k^{2.5}\log\left(\frac{k}{\delta}\right)}{\varepsilon}\right)$$

$$m \in O\left(\frac{\log(k/\beta)}{w_{\min}}\right),$$

then $\mathcal{M}$ (α, β)-learns D.

Proof. We mainly focus on the privacy guarantees here because the accuracy would follow from Theorem C.22 after setting the parameters appropriately. Note that the privacy parameters in this case in the set PrivParams are ε, δ. As per the advanced composition guarantees of DP (Lemma A.19), we set the privacy parameters in PrivParams$'$ in Algorithm 6 to be $\varepsilon' = \frac{\varepsilon}{\sqrt{12k\log(1/\delta)}}$ and $\delta' = \frac{\delta}{4k}$. Define $\varepsilon_0 = \sqrt{k}\varepsilon'$.

Now, we show that the intermediate steps in the call to Algorithm 6 yield (ε, δ)-DP. We don't release the output of that step itself, but use it in subsequent steps. The clusters are formed on the basis of intersection of the privately formed balls with the private dataset, and are then used subsequently in other private algorithms. So, the final algorithm would be private, as well. Thus, it is enough to prove the privacy guarantees of those intermediate steps that yield the said balls.

In Algorithm 6, we start working on the private dataset X inside the loop, specifically, when we use private PCA. In each iteration i, we ensure that each row of the input to Algorithm 4 $(Z^i - M^i)$ has norm at most R_i by virtue of selecting points that lie in a ball of radius R_i, and by recentering the points appropriately. Then by Lemma C.19, the sensitivity of Y^TY in Algorithm 4 is at most $2R_i^2$. Therefore, adding Gaussian noise calibrated to ε', δ' to Y^TY, that is, setting $f_{PCA}(\varepsilon', \delta') = \frac{\sqrt{2\ln(2/\delta')}}{\varepsilon'}$, implying that $\sigma_P = \frac{2R_i^2\sqrt{2\ln(2/\delta')}}{\varepsilon'}$ is enough to ensure (ε', δ')-DP in this step (Lemmata A.22 and A.18). The next step in Algorithm 6 that works with the private data is at the call to Algorithm 5. In this algorithm, only one call to PCount is performed, and this is the only time it works with the private dataset. Therefore, by the guarantees of Lemma A.23, we know that this step is ε'-DP. The next steps are either partitioning the private data or adding the isolated private data to C. Since neither of those sets are being released to public, there is no privacy loss here. Therefore, each iteration is $(2\varepsilon', \delta')$-DP (by Lemma A.19). Applying composition over all $2k$ iterations, we have that all operations together in the entire run of the loop in the algorithm are $(2\varepsilon, \delta)$-DP.

In Theorem C.21, setting $f'_{PCA}(\text{PrivParams}) = f_{PCA}(\varepsilon_0, \delta_0) \in O\left(\frac{\log(k/\delta)}{\varepsilon}\right)$ give us the right sample complexity for accuracy for all calls to Algorithm 4 (by Lemma A.19), since we already

multiply the $\sqrt{k}$ factor in the numerator. Similarly, setting $f'_{\mathrm{PCount}}(\mathrm{PrivParams}) = f_{\mathrm{PCount}}(\varepsilon_0) = \frac{\varepsilon}{\sqrt{12\log(1/\delta)}}$ in Theorem C.21 gives us the right sample complexity for all calls to Algorithm 5.

Now, each call to the approximate-DP Gaussian learner (Lemma A.24) is on a disjoint part of the private dataset X. Therefore, all the k calls together are (ε, δ)-DP because changing one point in X can change one point in only one of the clusters. By the same reasoning, all calls to PCount together are (ε, δ)-DP.

As far as the accuracy goes, the first two terms in the sample complexity ensures that enough points go to each call to the Gaussian estimator from Lemma A.24 and to each call to PCount because of Lemma C.7. Therefore, our GMM estimator (α, β)-learns the mixture. □

C.2.2 ρ-zCDP Algorithm

Here, we state our results under zCDP, and instantiate Algorithms 4, 6, and 7 for this version of DP.

Theorem C.24. *For all $\alpha, \beta, \rho > 0$, there exists an $O(\rho)$-zCDP algorithm $\mathcal{M}$ that takes n private samples and m public samples from $D \in \mathcal{G}(d, k, s)$, and is private with respect to the private samples, such that if $D = \{(\mu_1, \Sigma_1, w_1), \ldots, (\mu_k, \Sigma_k, w_k)\}$ satisfies Condition 2, for each $i \in [k]$, $\|\mu_i\| \le R$ and $\mathbb{I} \preceq \Sigma_i \preceq K\mathbb{I}$, and*

$$
\begin{aligned}
s &\in \Omega(\sqrt{k\ln((n+m)k/\beta)}) \\
n &\in O\left(\frac{d^2 \ln\left(\frac{k}{\beta}\right)}{w_{\min}\alpha^2} + \frac{d^2 \cdot \mathrm{polylog}\left(\frac{dk}{\alpha\beta\rho}\right) + d\log\left(\frac{dk\log(R)}{\alpha\beta\rho}\right)}{w_{\min}\alpha\sqrt{\rho}}\right. \\
&\qquad \left. + \frac{d^{1.5}\sqrt{k\log(K)} \cdot \mathrm{polylog}\left(\frac{dk\log(K)}{\rho\beta}\right) + \sqrt{dk\log\left(\frac{Rdk}{\beta}\right)} + d^{1.5}k^{2.5}}{\sqrt{\rho}}\right) \\
m &\in O\left(\frac{\log(k/\beta)}{w_{\min}}\right),
\end{aligned}
$$

then $\mathcal{M}$ (α, β)-learns D.

Proof. Again, we focus on the privacy guarantees because the accuracy would follow from Theorem C.22 after setting the parameters correctly. Note that the privacy parameter in this case in the set $\mathrm{PrivParams}$ is ρ. As per the advanced composition guarantees of DP (Lemma A.19), we set the privacy parameter in $\mathrm{PrivParams}'$ in Algorithm 6 to be $\rho' = \frac{\rho}{2k}$.

Now, we show that the intermediate steps in the call to Algorithm 6 yield $O(\rho)$-zCDP. As before, it is enough to prove the privacy guarantees of those intermediate steps, since the output of the algorithm itself wouldn't be released, but would be used by Algorithm 7, instead.

In Algorithm 6, we start working on the private dataset X inside the loop, specifically, when we use private PCA. In each iteration i, by the same argument as in the proof of Theorem C.23, the sensitivity of Y^TY in the call to Algorithm 4 is $2R_i^2$. Therefore, adding Gaussian noise calibrated to ρ' to Y^TY, that is, setting $f_{PCA}(\rho') = \frac{1}{\sqrt{2\rho'}}$, implying that $\sigma_P = \frac{2R_i^2}{\sqrt{2\rho'}}$ is enough to ensure ρ'-zCDP in this step (Lemmata A.22 and A.18). The next step in Algorithm 6 that works with the private data is at the call to Algorithm 5. In this algorithm, only one call to PCount is performed, and this is the only time it works with the private dataset. Therefore, by the guarantees of Lemma A.23, we know that this step is ρ'-zCDP. Therefore, each iteration is $2\rho'$-zCDP (by Lemma A.19). Applying composition over all $2k$ iterations, we have that all operations together in the entire run of the loop in the algorithm are 2ρ-zCDP.

In Theorem C.21, setting $f'_{PCA}(\mathrm{PrivParams}) = f_{PCA}(\rho) \in O\left(\frac{1}{\sqrt{\rho}}\right)$ give us the right sample complexity for accuracy for all calls to Algorithm 4 (by Lemma A.19), since we already multiply the $\sqrt{k}$ factor in the numerator. Similarly, setting $f'_{\mathrm{PCount}}(\mathrm{PrivParams}) = f_{\mathrm{PCount}}(\sqrt{2\rho}) = \sqrt{2\rho}$ in Theorem C.21 gives us the right sample complexity for all calls to Algorithm 5.

Now, each call to the zCDP Gaussian learner (Lemma A.24) is on a disjoint part of the private dataset X. Therefore, all the k calls together are ρ-zCDP because changing one point in X can change one point in only one of the clusters. By the same reasoning, all calls to PCount together are ρ-zCDP.

For the accuracy guarantees, the first two terms in the sample complexity ensures that enough points go to each call to the Gaussian estimator from Lemma A.24 and to each call to PCount because of Lemma C.7. Therefore, our GMM estimator (α, β)-learns the mixture. □

C.3 $\widetilde{O}(d/w_{\min})$ Public Samples

In this subsection, we assume that we have $\widetilde{O}(d/w_{\min})$ public data samples available. In this case, the number of public samples is enough to do PCA accurately. So, the approach would be much simpler than before, since we don't need to find a supercluster anymore. Also, because we have enough public samples, we could simply use the public dataset itself to partition the data when in low dimensions. Therefore, the entire clustering operations to partition the private dataset could be done using the public dataset itself, without having to touch the private dataset at all. As in the previous subsection, we will first give a general private algorithm for estimating the GMM, but will instantiate it separately for approximate DP and zCDP.

We start with the low-dimensional partitioner first, which just uses the public dataset, and the public query, as defined in 3.

Algorithm 8: Public Low-Dimensional Partitioner $\mathrm{LowDimPartitioner}_{m,w_{\min}}(\widetilde{Y}, r_{\max}, r_{\min})$

Input: Public Samples $\widetilde{Y}_1, \ldots, \widetilde{Y}_{m'} \in \mathbb{R}^d$. Parameters $m \geq m', w_{\min} > 0$.
Output: A tuple of centre $c \in \mathbb{R}^d$, radius $R \in \mathbb{R}$, or $\perp$.

For $i \leftarrow 0, \ldots, \log(r_{\max}/r_{\min})$
  $r_i \leftarrow \frac{r_{\max}}{2^i}$
  For $j \leftarrow 1, \ldots, m'$
    $c_j \leftarrow \widetilde{Y}_j$
    If $\mathrm{Q}_{\mathrm{Pub}}(\widetilde{Y}, c_j, r_i, \frac{m w_{\min}}{2}) = \textsc{True}$
      Return $(c = c_j, R = 2r_i)$.

Return $\perp$.

Theorem C.25. *Let Y be a clean subset of a set of public samples from $D \in \mathcal{G}(d,k)$, Π be a projection matrix to ℓ dimensions, $\widetilde{Y} = Y\Pi$ be the input to Algorithm 8, $r_{\max}, r_{\min} > 0$, such that for some $j \in [m']$, $\widetilde{Y} \cap B_{r_{\max}}\left(\widetilde{Y}_i\right) = \widetilde{Y}$ and $B_{r_{\max}}\left(\widetilde{Y}_i\right)$ is pure with respect to the components of D projected on to the subspace of Π having points in $\widetilde{Y}$. Suppose $\sigma_{\max}^2$ is the largest directional variance among the Gaussians that have points in Y. Let that Gaussian be $\mathcal{N}(\mu, \Sigma)$. Suppose,*

$$m \geq O\left(\frac{d\ln(k/\beta)}{w_{\min}}\right).$$

Then we have the following with probability at least $1-\beta$.

1. *Suppose there are at least two Gaussians that have points in Y. For any other Gaussian $\mathcal{N}(\mu', \Sigma')$ that has points in Y, suppose for $N > m$, $\|\mu\Pi - \mu'\Pi\| \geq \Omega(\sigma_{\max}\sqrt{\ell \ln(N\ell/\beta)})$. Let $r_{\min} \leq \sigma_{\max}\sqrt{2\ell\ln(2N\ell/\beta)}$. Then the ball returned by Algorithm 8 is pure with respect to the components of D projected by Π, which have points in $\widetilde{Y}$, such that $\mathbb{R}^d \setminus B_R(c)$ only contains components from D projected by Π that have points in $\widetilde{Y}$, and so does $B_R(c)$.*
2. *Suppose there is only one Gaussian that has points in Y. Then the algorithm returns $\perp$.*

Proof. The proof for the first part is the same as that for the first part of Theorem C.20. So, we don't discuss that any further.

For the next part, note that we could relax the query $\mathrm{Q}_{\mathrm{Pub}}$ by modifying the second constraint to say $|X \cap (B_{11r}(c) \setminus B_r(c))| < \frac{t}{320}$. If the answer to the original query is true, then the answer to the

modified query would be true, as well. In other words, if the answer to the modified query is false, then the answer to the original query is false, too. Therefore, it is enough to show that with probability at least $1-\beta$, there cannot be an 11-terrific ball in $\widetilde{Y}$. As argued in the proof of Theorem C.20, it is enough to show that there exists no 3-terrific ball in $\widetilde{Y}$. We already proved this in the proof of Theorem C.20. Hence, we have the claim. □

Next, we provide a clustering algorithm to partition the private dataset that just operates on the public dataset (Algorithm 9). The algorithm is the same as Algorithm 6, but uses the public data solely to partition the private data. It uses tools like PCA and Algorithm 8 on public data.

Algorithm 9: GMM Easy Clustering $\mathrm{EasyClustering}_{\beta,k,w_{\min}}(\widetilde{X}, X)$

Input: Private samples $X = (X_1, \ldots, X_n) \in \mathbb{R}^{n\times d}$. Public samples $\widetilde{X} = (\widetilde{X}_1, \ldots, \widetilde{X}_m) \in \mathbb{R}^{m\times d}$. Parameters $\beta, k, w_{\min} > 0$.
Output: Set $C \subset \mathcal{P}(X)$.

Let $Y \leftarrow \widetilde{X}$ and $Z \leftarrow X$.
Let $C \leftarrow \emptyset$.
Let Q_{Priv} and Q_{Pub} be queues of sets of points.
Add set Z to Q_{Priv}, and set Y to Q_{Pub}.
Set $count \leftarrow 0$ and $i \leftarrow 1$.

While $count < k$ *and* $i \leq 2k$
  Pop Q_{Priv} to get Z, and Q_{Pub} to get Y.

  `// PCA: project points of both datasets onto returned subspace.`
  Let Π_i be the top-k subspace of $Y^T Y$.
  $Y' \leftarrow Y\Pi_i$ and $Z' \leftarrow Z\Pi_i$.

  `// If there is only one Gaussian, add it to` C`, otherwise further partition the datasets.`
  Let $R_i \leftarrow 4 \max_{y_1,y_2\in Y}\{\|y_1 - y_2\|\}$ and $r_i \leftarrow \frac{\sqrt{2k\ln(2(n+m)k/\beta)}}{4\sqrt{d}} \min_{y_1,y_2\in Y}\{\|y_1 - y_2\|\}$.
  $B \leftarrow \mathrm{LowDimPartitioner}_{m,w_{\min}}(Y', R_i, r_i)$.
  If $B = \perp$
    $C \leftarrow C \cup \{Z^i\}$.
    $count \leftarrow count + 1$.
  Else
    B is an ordered pair (c'_i, r'_i).
    $S' \leftarrow B_{r'_i}(c') \cap Y'$ and $T' \leftarrow B_{r'_i}(c') \cap Z'$.
    Let S be points in Y^i corresponding to S', and T be points in Z^i corresponding to T'.
    Add S to Q_{Pub} and T to Q_{Priv}.
    Add $Y^i \setminus S$ to Q_{Pub} and $Z^i \setminus T$ to Q_{Priv}.

  $i \leftarrow i + 1$
Return C.

Theorem C.26. *There exists an algorithm (Algorithm 9) that takes n private samples X, and m public samples from $D \in \mathcal{G}(d,k,s)$ satisfying Condition 2, and outputs a partition C of X, such that given Conditions C.1 to C.6 hold, and*

$$s \geq \Omega(\sqrt{k\ln((n+m)k/\beta)}),$$

and

$$m \geq O\left(\frac{d\ln(k/\beta)}{w_{\min}}\right),$$

then with probability at least $1 - O(\beta)$, $|C| = k$ and each $S \in C$ is clean and non-empty.

Proof. As in the proof of Theorem C.21, it is enough to show that $\forall i \geq 0$, at the end of the i-th iteration, (1) a new Gaussian component is isolated in C and $count$ equals the number of isolated

components in C, (2) or the private dataset X is further partitioned into non-empty, clean subsets, and the public dataset $\widetilde{X}$ is further partitioned into clean subsets, such that $|Q_{\text{Pub}}| = |Q_{\text{Priv}}|$ and for each $j \in [|Q_{\text{Pub}}|]$, $Q_{\text{Pub}}[j]$ and $Q_{\text{Priv}}[j]$ have points from the same Gaussian components. We can prove this via induction on i.

Base Case: The proof is exactly the same as in the proof of Theorem C.21.

Inductive Step: We assume for all iterations, up to and including some $i \geq 0$, the claim holds. Then we show that it holds for iteration $i+1$, as well. By the inductive hypothesis, we know that Z and Y are clean subsets, and contain points from the same components.

We first reason about the correctness of the PCA step. Using Lemma A.7 and our bound on m, we know that for each component (μ', Σ', w'), the empirical mean of all the points from that Gaussian in $\widetilde{X}$ would be at most $\frac{\sqrt{\|\Sigma'\|_2}}{\sqrt{w'}}$ away from μ'. By the same reasoning, the empirical mean of the same projected points would be at most $\frac{\sqrt{\|\Sigma'\|_2}}{\sqrt{w'}}$ away from $\mu'\Pi$ because the projection of a Gaussian is still a Gaussian. Lemmata A.13 (by setting $B = 0$) and A.14 together guarantee that the distance between the empirical mean of those points and that of the projected points of that component would be at most $O\left(\frac{\sigma_{\max}}{\sqrt{w'}}\right)$. The triangle inequality implies that $\|\mu' - \mu'\Pi\| \leq O\left(\frac{\sigma_{\max}}{\sqrt{w'}}\right)$. This happens with probability at least $1 - \frac{\beta}{2k}$.

Now, if there is just one component that has points in Y (hence, in Z), then on feeding Y' to Algorithm 5, we will get $\perp$ (by the guarantees from Theorem C.20) with probability at least $1 - \frac{\beta}{2k}$. Because Z is a clean subset, C now contains one more clean subset of X, and we have isolated a new component, and updated the value of $count$ to reflect the change.

Suppose there are at least two components that have points in Y (hence, in Z). Let the largest directional variance among the Gaussians that have points in Y be $\sigma^2_{\max}$, its mean be μ, and its mixing weight be w. From the guarantees of the PCA step, we know that $\|\mu - \mu\Pi_i\| \leq O\left(\frac{\sigma_{\max}}{\sqrt{w}}\right)$. Because of the separation condition, applying the triangle inequality, we know that the distance between $\mu\Pi_i$ and the projected mean of any other Gaussian having points in $Z\Pi$ is at least $\Omega(\sigma_{\max}\sqrt{k\ln((n+m)k)/\beta})$. Also, we know by the construction of the algorithm that for any $y \in Y$, $Y \subseteq B_{R_i}(y)$, and from Lemmata C.8, C.9 and C.11 that $B_{R_i}(y)$ would contain all the components of D that have points in Y. Also, by Lemmata C.9, and C.11, $r_i \leq \sigma_{\max}\sqrt{2k\ln(2(n+m)k/\beta)}$. Then from the guarantees of Theorem C.25, we have that a ball is returned, and it is pure with respect to the components of the mixture projected by Π_i that have points in Y'. Therefore, subsets S' and T' are clean, and so are S and T. This implies that $Y \setminus S$ and $Z \setminus T$ are clean, too. By adding them in order to Q_{Pub} and Q_{Priv} respectively, we partition the datasets $\widetilde{X}$ and X further into clean subsets, and preserve the ordering of the components in the respective queues. This proves the claim.

As argued before in the proof Theorem C.21, allowing up to $2k$ iterations is enough. Therefore, we get a clean partition of X. □

Finally, we provide the DP algorithm for learning GMM's in this regime of public data. The algorithm is the same as Algorithm 7, except that it calls Algorithm 9, instead of Algorithm 6.

Theorem C.27. *For all $\alpha, \beta > 0$ and sets of privacy parameters* PrivParams*, there exists an algorithm (Algorithm 10) that takes n private samples and m public samples from $D \in \mathcal{G}(d, k, s)$, such that if $D = \{(\mu_1, \Sigma_1, w_1), \ldots, (\mu_k, \Sigma_k, w_k)\}$ satisfies Condition 2, and*

$$s \in \Omega(\sqrt{k\ln((n+m)k/\beta)})$$
$$n \in O\left(\frac{n_{GE}\ln(k/\beta)}{w_{\min}}\right)$$
$$m \in O\left(\frac{d\ln(k/\beta)}{w_{\min}}\right),$$

where n_{GE} is the sample complexity of privately learning a Gaussian using Lemma A.24 and Theorem B.5 according to the type of DP required, then it (α, β)-learns D.

Algorithm 10: DP GMM Easy Case Estimator $\mathrm{DPEasyEstimator}_{\mathrm{PrivParams},\alpha,\beta,k,w_{\min}}(\widetilde{X}, X)$

Input: Private samples $X = (X_1, \ldots, X_n) \in \mathbb{R}^{n\times d}$. Public samples $\widetilde{X} = (\widetilde{X}_1, \ldots, \widetilde{X}_m) \in \mathbb{R}^{m\times d}$. Parameters $\mathrm{PrivParams} \subset \mathbb{R}, \beta, k, w_{\min} > 0$.

Output: Set $\widehat{D} = \{(\widehat{\mu}_1, \widehat{\Sigma}_1, \widehat{w}_1), \ldots, (\widehat{\mu}_k, \widehat{\Sigma}_k, \widehat{w}_k)\}$.

`// Clustering.`

Set $C \leftarrow \mathrm{EasyClustering}_{\beta,k,w_{\min}}(X, \widetilde{X})$.

`// Parameter estimation.`

$\widehat{D} \leftarrow \emptyset$.

For $i \leftarrow 1, \ldots, k$

  Set $(\widehat{\mu}_i, \widehat{\Sigma}_i) \leftarrow \mathrm{DPGaussianEstimator}_{\mathrm{PrivParams},\frac{\beta}{k}}(C_i)$.

  Set $\widehat{w}_i \leftarrow \max\left\{\frac{1}{n} \cdot \mathrm{PCount}_{\mathrm{PrivParams}}(|C_i|), \frac{\alpha}{2k}\right\}$.

  $\widehat{D} \leftarrow \widehat{D} \cup \{(\widehat{\mu}_i, \widehat{\Sigma}_i, \widehat{w}_i)\}$.

Return $\widehat{D}$.

Proof. By Theorem C.26, we know that with probability at least $1 - \frac{\beta}{3}$, $|C| = k$, and for each $i \in [k]$, C_i is a clean subset of X.

Next, by the guarantees of Lemma A.24 and Theorem B.5, for each $i \in [k]$, with probability at least $1 - \frac{\beta}{3k}$, $\mathrm{d_{TV}}(\mathcal{N}(\widehat{\mu}_i, \widehat{\Sigma}_i), \mathcal{N}(\mu_i, \Sigma_i)) \le \alpha$.

Finally, by the same argument as in the proof of Theorem C.22, the mixing weights are estimated accurately, as well.

Therefore, taking the union bound over all failure events, we have the required result. □

C.3.1 (ε, δ)-DP Algorithm

Here, we instantiate Algorithm 10 for the case of approximate DP.

Theorem C.28. *For all $\alpha, \beta, \varepsilon, \delta > 0$, there exists an $(O(\varepsilon), O(\delta))$-DP algorithm $\mathcal{M}$ that takes n private samples and m public samples from $D \in \mathcal{G}(d, k, s)$, and is private with respect to the private samples, such that if $D = \{(\mu_1, \Sigma_1, w_1), \ldots, (\mu_k, \Sigma_k, w_k)\}$ satisfies Condition 2, and*

$$s \in \Omega(\sqrt{k \ln((n+m)k/\beta)})$$

$$n \in O\left(\frac{d^2 \ln\left(\frac{k}{\beta}\right)}{w_{\min}\alpha^2} + \frac{(d^2 \log(k/\delta) + d \log^{1.5}(k/\delta)) \cdot \mathrm{polylog}\left(d, \frac{k}{\beta}, \frac{1}{\alpha}, \frac{\sqrt{k}}{\varepsilon}, \ln\left(\frac{k}{\delta}\right)\right)}{w_{\min}\alpha\varepsilon}\right)$$

$$m \in O\left(\frac{d \log(k/\beta)}{w_{\min}}\right),$$

then $\mathcal{M}$ (α, β)-learns D.

Proof. We again focus on the privacy guarantees here because the accuracy would follow from Theorem C.27 after setting the parameters appropriately. Note that the privacy parameters in this case in the set $\mathrm{PrivParams}$ are ε, δ. We just have to show the privacy guarantees for the calls to the Gaussian estimator and PCount because all other steps involve computations on the public dataset.

By the same argument as in the proof for Theorem C.23, all k calls together to the approximate DP learner from Lemma A.24 are (ε, δ)-DP, and all calls together to PCount are (ε, δ)-DP. Therefore, we get $(2\varepsilon, 2\delta)$-DP guarantee in the end.

For the accuracy goal, the sample complexity ensures that enough points go to each call to the Gaussian estimator from Lemma A.24 and to each call to PCount because of Lemma C.7. Therefore, our GMM estimator (α, β)-learns the mixture. □

C.3.2 ρ-zCDP Algorithm

Finally, we instantiate Algorithm 10 for the case of zCDP. Note that we don't need any bounds on the range parameters in this case because we are using the Gaussian learner from Section B.

Theorem C.29. *For all $\alpha, \beta, \rho > 0$, there exists an $O(\rho)$-zCDP algorithm $\mathcal{M}$ that takes n private samples and m public samples from $D \in \mathcal{G}(d, k, s)$, and is private with respect to the private samples, such that if $D = \{(\mu_1, \Sigma_1, w_1), \ldots, (\mu_k, \Sigma_k, w_k)\}$ satisfies Condition 2, and*

$$s \in \Omega(\sqrt{k \ln((n+m)k/\beta)})$$
$$n \in O\left(\frac{d^2 \ln\left(\frac{k}{\beta}\right)}{w_{\min}\alpha^2} + \frac{d^2 \cdot \mathrm{polylog}\left(\frac{dk}{\alpha\beta\rho}\right)}{w_{\min}\alpha\sqrt{\rho}}\right)$$
$$m \in O\left(\frac{d \log(k/\beta)}{w_{\min}}\right),$$

then $\mathcal{M}$ (α, β)-learns D.

Proof. We mainly prove the privacy guarantees because the accuracy would follow from Theorem C.27 after setting the parameters correctly. Note that the privacy parameter in this case in the set $\mathrm{PrivParams}$ is ρ.

Using the same argument as in the proof of Theorem C.23, we have ρ-zCDP for all k calls to the learner in Section B (by Theorem B.5), and ρ-zCDP for all calls together to PCount. Therefore, we have 2ρ-zCDP in the end.

By the same argument as before, the sample complexity is enough to send enough points to each call to the learner in Section B and to PCount. Therefore, our GMM estimator (α, β)-learns the mixture. □

D A Proof-of-Concept Numerical Result

For Gaussian mean estimation, our approach is relatively simple to implement on top of an existing private algorithm. We offer some proof-of-concept simulations that demonstrate the effectiveness of public data in private statistical estimation.[5] In Figure 1, we show plots that evaluate 1-public-sample private mean estimation (the algorithm described in Section 2.1 and in more detail in Appendix B.1.1).

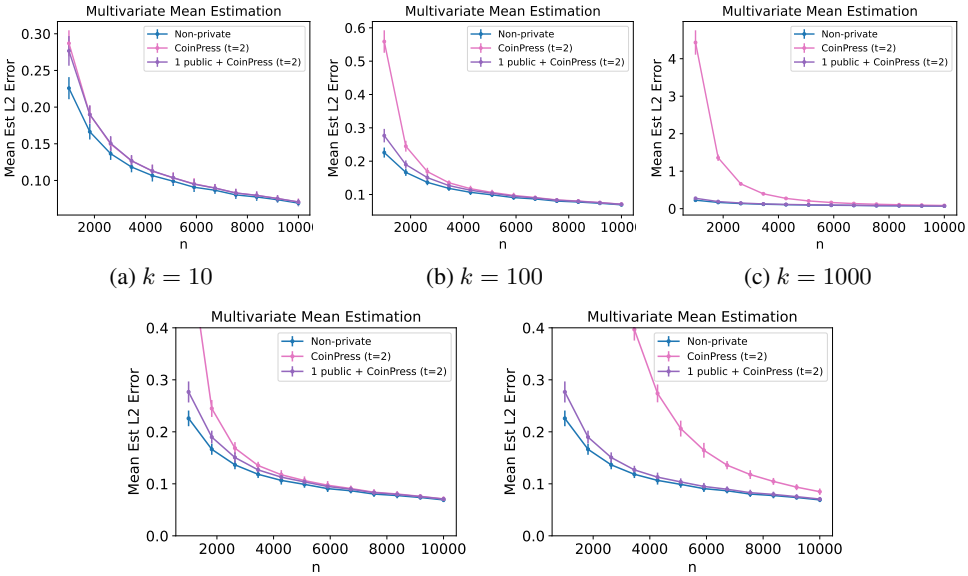
We examine the effect of 1 public sample on the performance of CoinPress [BDKU20] with its best parameter setting ($t = 2$), in a case where the initial a priori bounds on the mean are weak. Concretely, we draw n samples from a $d = 50$ dimensional Gaussian $\mathcal{N}(\mu_k, I_d)$, where $\mu_k = k \cdot [1, ..., 1]^T$ and correspondingly set our a priori bound $R = k\sqrt{d}$ for CoinPress. We show results for $k \in [10, 100, 1000]$, representing varying levels of strength in our a prior bounds on the mean.

We follow the evaluation protocol from [BDKU20]: we target zCDP with $\rho = 0.5$, and at each sample size n we run the estimator 100 times and report the 10% trimmed mean of error from the ground truth (we additionally report the 10% trimmed standard deviation as error bars). We also follow their practice of treating target failure probabilities β_i of various steps of the algorithm as hyperparameters that can be tuned for the best empirical results. The new step we introduce (using the public sample to set R based on d, β) uses $\beta = 0.01$, which is the same value used for all the β_i's in CoinPress.

The numerical result demonstrates the promise of utilizing public data for private data analysis, and confirms the takeaway that very little public data can help greatly when a priori knowledge of the private data is weak. As is visible from the plots, the error of our public-private algorithm tracks the non-private algorithm closer when a priori bounds on the mean are weak ($k = 1000$ case).

Note that these results are only meant to be a proof-of-concept simulation to demonstrate the promise of public data – thorough tuning and evaluation of these algorithms (which is necessary to bring these algorithms to practice) is an important direction for future work.

[5] Our code is available at https://github.com/alexbie98/1pub-priv-mean-est. Experiments run within two minutes on a 2020 M1 MacBook Air.



(a) $k = 10$ (b) $k = 100$ (c) $k = 1000$

(d) $k = 100$ (zoomed in) (e) $k = 1000$ (zoomed in)

Figure 1: Comparing the error of CoinPress [BDKU20] under its best setting, against CoinPress with 1 public sample for mean estimation of $\mathcal{N}(k \cdot [1, ..., 1]^T, I_d)$ for $d = 50$, targeting zCDP at $\rho = 0.5$. Larger k corresponds to weaker a priori bounds on the mean for CoinPress. For large k, a single public sample significantly improves results.