Differentially-Private Clustering of Easy Instances
Full Version

Edith Cohen* Haim Kaplan* Yishay Mansour* Uri Stemmer†
Eliad Tsfadia*

June 10, 2021

Abstract

Clustering is a fundamental problem in data analysis. In differentially private clustering, the goal is to identify k cluster centers without disclosing information on individual data points. Despite significant research progress, the problem had so far resisted practical solutions. In this work we aim at providing simple implementable differentially private clustering algorithms that provide utility when the data is "easy," e.g., when there exists a significant separation between the clusters.

We propose a framework that allows us to apply non-private clustering algorithms to the easy instances and privately combine the results. We are able to get improved sample complexity bounds in some cases of Gaussian mixtures and k-means. We complement our theoretical analysis with an empirical evaluation on synthetic data.

*Google Research and Blavatnik School of Computer Science, Tel Aviv University. E-mails:{edith@alumni.stanford.edu, haimk@tau.ac.il, mansour.yishay@gmail.com, eliadtsfadia@gmail.com}

†Google Research and Ben-Gurion University. E-mail: {u@uri.co.il}.

Contents

1 Introduction

Differential privacy [DMNS06] is a mathematical definition of privacy, that aims to enable statistical analyses of databases while providing strong guarantees that individual-level information does not leak. Privacy is achieved in differentially private algorithms through randomization and the introduction of "noise" to obscure the effect of each individual, and thus differentially private algorithms can be less accurate than their non-private analogues. In most cases, this loss in accuracy is studied theoretically, using asymptotic tools. As a result, there is currently a significant gap between what is known to be possible *theoretically* and what can be done *in practice* with differential privacy. In this work we take an important step towards bridging this gap in the context of *clustering related tasks*.

The construction of differentially private clustering algorithms has attracted a lot of attention over the last decade, and many different algorithms have been suggested.[1] However, to the best of our knowledge, none of these algorithms have been implemented: They are not particularly simple and suffer from large hidden constants that translate to a significant loss in utility, compared to non-private implementations.

Question 1.1. *How hard is it to cluster privately with a practical implementation?*

We take an important step in this direction using the following approach. Instead of directly tackling "standard" clustering tasks, such as k-means clustering, we begin by identifying a very simple clustering problem that still seems to capture many of the challenges of practical implementations (we remark that this problem is completely trivial without privacy requirements). We then design effective (private) algorithms for this simple problem. Finally, we reduce "standard" clustering tasks to this simple problem, thereby obtaining private algorithms for other tasks.

In more detail, we introduce the following problem, called the *k-tuple clustering* problem.

Definition 1.2 (informal, revised in Definition 3.7)**.** *An instance of the* k-tuple clustering *problem is a collection of k-tuples. Assuming that the input tuples can be partitioned into k "obvious clusters", each consisting of one point of each tuple, then the goal is to report k "cluster-centers" that correctly partition the input tuples into clusters. If this assumption on the input structure does not hold, then the outcome is not restricted.*

Remark 1.3.

1. *By "obvious clusters" we mean clusters which are far away from each other.*
2. *The input tuples are* unordered*. This means, e.g., that the "correct" clustering might place the first point of one tuple with the fifth point of another tuple.*
3. *Of course, we want to solve this problem while guaranteeing differential privacy. Intuitively, this means that the outcome of our algorithm should not be significantly effected when arbitrarily modifying one of the input tuples.*

Observe that without the privacy requirement this task is trivial: We can just take one arbitrary input tuple $(x_1, ..., x_k)$ and report it. With the privacy requirement, this task turns out to be non-trivial. It's not that this problem cannot be solved with differential privacy. It can. It's not even

[1] [BDMN05a; NRS07; FFKN09; McS09; GLM+10; MTS+12; WWS15; NCBN16; SCL+16; NSV16; FXZR17; BDL+17; NS18; HL18b; KS18; Ste20; SSS20; GKM20; Ngu20]

that the problem requires large amounts of data asymptotically. It does not. However, it turns out that designing an implementation with a practical privacy-utility tradeoff, that is effective on finite datasets (of reasonable size), is quite challenging.

1.1 Our algorithms for the k-tuple problem

We present two (differentially private) algorithms for the k-tuple clustering problem, which we call PrivatekAverages and PrivatekNoisyCenters. Both algorithms first privately test if indeed the input is partitioned into k obvious clusters and quit otherwise. They differ by the way they compute the centers in case this test passes. Algorithm PrivatekAverages privately averages each identified cluster. Algorithm PrivatekNoisyCenters, on the other hand, does not operate by averaging clusters. Instead, it selects one of the input k-tuples, and then adds a (relatively small) Gaussian noise to every point in this tuple. We prove that this is private if indeed there are k obvious clusters in the input. We evaluate these two algorithms empirically, and show that, while algorithm PrivatekAverages is "better in theory", algorithm PrivatekNoisyCenters is much more practical for some interesting regimes of parameters.

We now give a simplified overview of the ideas behind our algorithms. For concreteness, we focus here on PrivatekAverages. Recall that in the k-tuple clustering problem, we are only required to produce a good output assuming the data is "nice" in the sense that the input tuples can be clustered into k "far clusters" such that every cluster contains exactly one point from every tuple. However, with differential privacy we are "forced" to produce good outputs even when this niceness assumption does not hold. This happens because if the input data is "almost nice" (in the sense that modifying a small number of tuples makes it nice) then differential privacy states that the outcome of the computation should be close to what it is when the input data is nice.

So, the definition of differential privacy forces us to cope with "almost nice" datasets. Therefore, the niceness test that we start with has to be a bit clever and "soft" and succeed with some probability also for data which is "almost nice". Then, in order to achieve good performances, we have to utilize the assumption that the data is "almost nice" when we compute the private centers. To compute these centers, Algorithm PrivatekAverages determines (*non-privately*) a clustering of the input tuples, and then averages (with noise) each of the clusters. The conceptual challenge here is to show that even though the clustering of the data is done non-privately, it is stable enough such that the outcome of this algorithm still preserves privacy.

1.2 Applications

The significance of algorithms PrivatekAverages and PrivatekNoisyCenters is that many clustering related tasks can be privately solved by a reduction to the k-tuple clustering problem. In this work we explore two important use-cases: (1) Privately approximating the k-means under stability assumption, and (2) Privately learning the parameters of a mixture of well-separated Gaussians.

k-Means Clustering

In k-means clustering, we are given a database $\mathcal{P}$ of n input points in $\mathbb{R}^d$, and the goal is to identify a set C of k *centers* in $\mathbb{R}^d$ that minimizes the sum of squared distances from each input point to its nearest center. This problem is NP-hard to solve exactly, and even NP-hard to approximate to within a multiplicative factor smaller than 1.0013 [LSW17]. The current (non-private) state-of-the-art algorithm achieves a multiplicative error of 6.357 [ANFSW19].

One avenue that has been very fruitful in obtaining more accurate algorithms (non-privately) is to look beyond worst-case analysis [ORSS12; ABS10; ABS12; BBG09; BL12; KK10]. In more details, instead of constructing algorithms which are guaranteed to produce an approximate clustering for any instance, works in this vain give stronger accuracy guarantees by focusing only on instances that adhere to certain "nice" properties (sometimes called stability assumptions or separation conditions). The above mentioned works showed that such "nice" inputs can be clustered much better than what is possible in the worst-case (i.e., without assumptions on the data).

Given the success of non-private stability-based clustering, it is not surprising that such stability assumptions were also utilized in the privacy literature, specifically by Nissim, Raskhodnikova, and Smith [NRS07], Wang, Wang, and Singh [WWS15], Huang and Liu [HL18b], and Shechner, Sheffet, and Stemmer [SSS20]. While several interesting concepts arise from these four works, none of their algorithms have been implemented, their algorithms are relatively complex, and their practicability on finite datasets is not clear.

We show that the problem of stability-based clustering (with privacy) can be reduced to the k-tuple clustering problem. Instantiating this reduction with our algorithms for the k-tuple clustering problem, we obtain a simple and practical algorithm for clustering "nice" k-means instances privately.

Learning Mixtures of Gaussians. Consider the task of *privately* learning the parameters of an unknown mixtures of Gaussians given i.i.d. samples from it. By now, there are various private algorithms that learn the parameters of a *single* Gaussian [KV18; KLSU19; CWZ19; BS19; KSU20; BDKU20]. Recently, [KSSU19] presented a private algorithm for learning mixtures of well-separated (and bounded) Gaussians. We remark, however, that besides the result of [BDKU20], which is a practical algorithm for learning a single Gaussian, all the other results are primarily theoretical.

By a reduction to the k-tuples clustering problem, we present a simple algorithm that privately learns the parameters of a separated (and bounded) *mixture* of k Gaussians. From a practical perspective, compared with the construction of the main algorithm of [KSSU19], our algorithm is simple and implementable. From a theoretical perspective, our algorithm offers reduced sample complexity, weaker separation assumption, and modularity. See Section 6.4 for the full comparison.

1.3 Other Related Work

The work of Nissim, Raskhodnikova, and Smith [NRS07] presented the sample-and-aggregate method to convert a non-private algorithm into a private algorithm, and applied it to easy clustering problems. However, their results are far from being tight, and they did not explore certain considerations (e.g., how to minimize the impact of a large domain in learning mixture of Gaussians).

Another work by Bun *et al.* [BKSW21] provides a general method to convert from a cover of a class of distributions to a private learning algorithm for the same class. The work gets a near-optimal sample complexity, but the algorithms have exponential running time in both k and d and their learning guarantees are incomparable to ours (they perform proper learning, while we provide clustering and parameter estimation).

In the work of [KSSU19], they presented an alternative algorithm for learning mixtures of Gaussians, which optimizes the sample-and-aggregate approach of [NRS07], and is somewhat similar to our approach. That is, their algorithm executes a non-private algorithm several times, each time for obtaining a new "k-tuple" of means estimations, and then aggregates the findings by privately determine a new k-tuple of means estimation. But their approach has two drawbacks. First,

in order to privately do that, their algorithm ignores the special k-tuples structure, and apply a more wasteful and complicated "minimal enclosing ball" algorithm from [NS17; NSV16]. Second, in contrast to them, for creating a k-tuple, our algorithm only applies a non-private algorithm for *separating* the samples in the mixture (i.e., for determine which samples belong to the same Gaussian), and not for estimating their parameters. This yields that we need less samples per invocation of the non-private algorithm for creating a single k-tuple, which results with an improved sample complexity (each k-tuple in our case is just the averages of each set of samples, which might not necessarily be very close to the true means, but is close enough for our setting where the Gaussians are well-separated). Finally, given a private separation of the sample, we just apply some private algorithm for estimating the parameters of each (single) Gaussian (e.g., [KV18; KLSU19; CWZ19; BS19; KSU20; BDKU20]). For more details about our construction, see Section 6.

Furthermore, there are many differentially-private algorithms that are related to learning mixture of Gaussians (notably PCA) [BDMN05b; KT13; CSS13; DTTZ14], and differentially-private algorithms for clustering [NRS07; GLM+10; NSV16; NS17; BDL+17; KS18; HL18b; GKM20]. We remark that for the learning Gaussians mixtures problem, applying these algorithms naively would introduce a polynomial dependence on the range of the data, which we seek to avoid.

2 Preliminaries

2.1 Notation

In this work, a k-tuple $X = \{\mathbf{x}_1, \ldots, \mathbf{x}_k\}$ is an *unordered* set of k vectors $\mathbf{x}_i \in \mathbb{R}^d$. For $\mathbf{x} \in \mathbb{R}^d$, we denote by $\|\mathbf{x}\|$ the ℓ_2 norm of $\mathbf{x}$. For $\mathbf{c} \in \mathbb{R}^d$ and $r > 0$, we denote $B(\mathbf{c}, r) := \{\mathbf{x} \in \mathbb{R}^d \colon \|\mathbf{x} - \mathbf{c}\| \leq r\}$. For a multiset $\mathcal{P} \in (\mathbb{R}^d)^*$ we denote by $\mathrm{Avg}(\mathcal{P}) := \frac{1}{|\mathcal{P}|} \cdot \sum_{\mathbf{x} \in \mathcal{P}} \mathbf{x}$ the average of all points in $\mathcal{P}$. Throughout this work, a database $\mathcal{D}$ is a multiset. For two multisets $\mathcal{D} = \{x_1, \ldots, x_n\}$ and $\mathcal{D}' = \{x'_1, \ldots, x'_m\}$, we let $\mathcal{D} \cup \mathcal{D}'$ be the multiset $\{x_1, \ldots, x_n, x'_1, \ldots, x'_m\}$. For a multiset $\mathcal{D} = \{x_1, \ldots, x_n\}$ and a set S, we let $\mathcal{M} \cap S$ be the multiset $\{x_i\}_{i \in \mathcal{I}}$ where $\mathcal{I} = \{i \in [n] \colon x_i \in S\}$. say that X is a k-tuple, if X is a multiset of size k (i.e., an unordered tuple). All logarithms considered here are natural logarithms (i.e., in base e).

2.2 Indistinguishability and Differential Privacy

Definition 2.1 (Neighboring databases)**.** *Let $\mathcal{D} = \{x_1, \ldots, x_n\}$ and $\mathcal{D}' = \{x'_1, \ldots, x'_n\}$ be two databases over a domain $\mathcal{X}$. We say that $\mathcal{D}$ and $\mathcal{D}'$ are **neighboring** if there is exactly one index $i \in [n]$ with $x_i \neq x'_i$.*

Definition 2.2 ((ε, δ)-indistinguishable)**.** *Two random variable X, X' over a domain $\mathcal{X}$ are called (ε, δ)-indistinguishable, iff for any event $T \subseteq \mathcal{X}$, it holds that $\Pr[X \in T] \leq e^{\varepsilon} \cdot \Pr[X' \in T] + \delta$. If $\delta = 0$, we say that X and X' are ε-indistinguishable.*

Definition 2.3 ((ε, δ)-differential privacy [DMNS06])**.** *An algorithm $\mathcal{A}$ is called (ε, δ)-differentially private, if for any two neighboring databases $\mathcal{D}, \mathcal{D}'$ it holds that $\mathcal{A}(\mathcal{D})$ and $\mathcal{A}(\mathcal{D}')$ are (ε, δ)-indistinguishable. If $\delta = 0$ (i.e., pure privacy), we say that $\mathcal{A}$ is ε-differentially private.*

Lemma 2.4 ([BS16])**.** *Two random variable X, X' over a domain $\mathcal{X}$ are (ε, δ)-indistinguishable, iff there exist events $E, E' \subseteq \mathcal{X}$ with $\Pr[X \in E], \Pr[X' \in E'] \geq 1 - \delta$ such that $X|_E$ and $X'|_{E'}$ are ε-indistinguishable.*

2.2.1 Basic Facts

The following fact is a corollary of Lemma 2.4.

Fact 2.5. *Let X, X' be two random variables over a domain $\mathcal{X}$, and let $E, E' \subseteq \mathcal{X}$ be two events. If $X|_E$ and $X'|_{E'}$ are (ε, δ_1)-indistinguishable and $\Pr[X \in E], \Pr[X' \in E'] \geq 1 - \delta_2$, then X and X' are $(\varepsilon, \delta_1 + \delta_2)$-indistinguishable.*

Proof. Since $X|_E$ and $X'|_{E'}$ are (ε, δ_1)-indistinguishable, we deduce by Lemma 2.4 that there exists events $F \subseteq E$ and $F' \subseteq E'$ with $\Pr[X \in F \mid E], \Pr[X' \in F' \mid E'] \geq 1 - \delta_1$ such that $X|_F$ and $X'|_{F'}$ are $(\varepsilon, 0)$-indistinguishable. In addition, note that

$$\Pr[X \in F] = \Pr[X \in E] \cdot \Pr[X \in F \mid E] \geq (1 - \delta_2)(1 - \delta_1) \geq 1 - (\delta_1 + \delta_2).$$

Similarly, it holds that $\Pr[X' \in F'] \geq 1 - (\delta_1 + \delta_2)$. Therefore, by applying the opposite direction of Lemma 2.4 on the events F and F', we deduce that X and X' are $(\varepsilon, \delta_1 + \delta_2)$-indistinguishable. □

In addition, we use the following facts.

Fact 2.6. *Let X, X' be two ε-indistinguishable random variables over a domain $\mathcal{X}$, and let $E, E' \subseteq \mathcal{X}$ be two events with $\Pr[X \in E], \Pr[X' \in E'] \geq 1 - \delta$ for $\delta \leq 1/2$. Then $X|_E$ and $X'|_{E'}$ are $(\varepsilon + 2\delta)$-indistinguishable.*

Proof. Compute

$$\Pr[X = x \mid E] \leq \frac{\Pr[X = x]}{\Pr[E]} \leq \frac{e^{\varepsilon} \cdot \Pr[X' = x]}{1 - \delta} \leq e^{\varepsilon + 2\delta} \Pr\big[X' = x\big],$$

where the last inequality holds since $1 - \delta \geq e^{-2\delta}$ for $\delta \leq 1/2$. □

Fact 2.7. *Let X, X' be two random variables over a domain $\mathcal{X}$. Assume there exist events $E, E' \subseteq \mathcal{X}$ such that the following holds:*

- $\Pr[X \in E] \in e^{\pm\varepsilon} \cdot \Pr[X' \in E']$, *and*
- $X|_E$ *and* $X'|_{E'}$ *are* (ε^*, δ)-*indistinguishable, and*
- $X|_{\neg E}$ *and* $X'|_{\neg E'}$ *are* (ε^*, δ)-*indistinguishable.*

Then X, X' are $(\varepsilon + \varepsilon^*, \delta e^{\varepsilon})$-indistinguishable.

Proof. Fix an event $T \subseteq \mathcal{X}$ and compute

$$\begin{aligned}
&\Pr[X \in T] = \Pr[X \in T \mid E] \cdot \Pr[X \in E] + \Pr[X \in T \mid \neg E] \cdot \Pr[X \notin E] \\
&\leq \Big(e^{\varepsilon^*} \cdot \Pr\big[X' \in T \mid E'\big] + \delta\Big) \cdot e^{\varepsilon} \cdot \Pr\big[X' \in E'\big] + \Big(e^{\varepsilon^*} \cdot \Pr\big[X' \in T \mid \neg E'\big] + \delta\Big) \cdot e^{\varepsilon} \cdot \Pr\big[X' \notin E'\big] \\
&= e^{\varepsilon + \varepsilon^*} \cdot \Pr\big[X' \in T\big] + \delta e^{\varepsilon}.
\end{aligned}$$

□

2.2.2 Group Privacy and Post-Processing

Fact 2.8 (Group Privacy). *If $\mathcal{A}$ is (ε, δ)-differentially private, then for all pairs of databases $\mathcal{S}$ and $\mathcal{S}'$ that differ by k points it holds that $\mathcal{A}(\mathcal{S})$ and $\mathcal{A}(\mathcal{S}')$ are $(k\varepsilon, ke^{k\varepsilon}\delta)$-indistinguishable.*

Fact 2.9 (Post-processing). *If $\mathcal{A}$ is (ε, δ)-differentially private, then for every (randomized) function F it holds that $F \circ \mathcal{A}$ is (ε, δ)-differentially private.*

2.2.3 Composition

Theorem 2.10 (Basic composition, adaptive case [DRV10]). *If $\mathcal{A}_1$ and $\mathcal{A}_2$ satisfy $(\varepsilon_1, \delta_1)$ and $(\varepsilon_2, \delta_2)$ differential privacy (respectively), then any algorithm that adaptively uses $\mathcal{A}_1$ and $\mathcal{A}_2$ (and does not access the database otherwise) ensures $(\varepsilon_1 + \varepsilon_2, \delta_1 + \delta_2)$-differential privacy.*

Theorem 2.11 (Advanced composition [DRV10]). *Let $0 < \varepsilon_0, \delta' \leq 1$, and let $\delta_0 \in [0, 1]$. An algorithm that adaptively uses k algorithms that preserve $(\varepsilon_0, \delta_0)$-differential privacy (and does not access the database otherwise) ensures (ε, δ)-differential privacy, where $\varepsilon = \sqrt{2k\ln(1/\delta')} \cdot \varepsilon_0 + 2k\varepsilon_0^2$ and $\delta = k\delta_0 + \delta'$.*

2.2.4 The Laplace Mechanism

Definition 2.12 (Laplace distribution). *For $\sigma \geq 0$, let $\mathrm{Lap}(\sigma)$ be the Laplace distribution over $\mathbb{R}$ with probability density function $p(z) = \frac{1}{2\sigma}\exp\left(-\frac{|z|}{\sigma}\right)$.*

Fact 2.13. *Let $\varepsilon > 0$. If $X \sim \mathrm{Lap}(1/\varepsilon)$ then for all $t > 0$: $\quad \Pr[|X| > t/\varepsilon] \leq e^{-t}$.*

Definition 2.14 (Sensitivity). *We say that a function $f \colon \mathcal{U}^n \to \mathbb{R}$ has sensitivity λ if for all neigboring databases $\mathcal{S}, \mathcal{S}'$ it holds that $|f(\mathcal{S}) - f(\mathcal{S}')| \leq \lambda$.*

Theorem 2.15 (The Laplace Mechanism [DMNS06]). *Let $\varepsilon > 0$, and assume $f \colon \mathcal{U}^n \to \mathbb{R}$ has sensitivity λ. Then the mechanism that on input $\mathcal{S} \in \mathcal{U}^n$ outputs $f(\mathcal{S}) + \mathrm{Lap}(\lambda/\varepsilon)$ is ε-differentially private.*

2.2.5 The Gaussian Mechanism

Definition 2.16 (Gaussian distribution). *For $\mu \in \mathbb{R}$ and $\sigma \geq 0$, let $\mathcal{N}(\mu, \sigma^2)$ be the Gaussian distribution over $\mathbb{R}$ with probability density function $p(z) = \frac{1}{\sqrt{2\pi}}\exp\left(-\frac{(z-\mu)^2}{2\sigma^2}\right)$.*

Fact 2.17. *Let $\mathbf{X} = (X_1, \ldots, X_d)$, where the X_i's are i.i.d. random variables, distributed according to $\mathcal{N}(0, \sigma^2)$. Then for all $\beta > 0$: $\quad \Pr\left[\|\mathbf{X}\| \leq \left(\sqrt{d} + \sqrt{2\log(1/\beta)}\right) \cdot \sigma\right] \geq 1 - \beta$.*

Definition 2.18 (ℓ_2-sensitivity). *We say that a function $f \colon \mathcal{U}^n \to \mathbb{R}^d$ has ℓ_2-sensitivity λ if for all neigboring databases $\mathcal{S}, \mathcal{S}'$ it holds that $\|f(\mathcal{S}) - f(\mathcal{S}')\| \leq \lambda$.*

Theorem 2.19 (The Gaussian Mechanism [DKM+06]). *Let $\varepsilon, \delta \in (0, 1)$, and assume $f \colon \mathcal{U}^n \to \mathbb{R}^d$ has ℓ_2-sensitivity λ. Let $\sigma \geq \frac{\lambda}{\varepsilon}\sqrt{2\log(1.25/\delta)}$. Then the mechanism that on input $\mathcal{S} \in \mathcal{U}^n$ outputs $f(\mathcal{S}) + \left(\mathcal{N}(0, \sigma^2)\right)^d$ is (ε, δ)-differentially private.*

Observation 2.20. *For the case that $\mathcal{S} \in (\mathbb{R}^d)^n$ and $f(\mathcal{S}) = \mathrm{Avg}(\mathcal{S})$, if we are promised that each coordinate of the points is bounded by a segment of length Λ, then the sensitivity is bounded by $\lambda = \Lambda/n$, and therefore, by taking $\sigma = O(\frac{\Lambda}{\varepsilon n}\sqrt{\log(1/\delta)})$ we get by Fact 2.17 that with probability $1-\beta$, the resulting point $\mathbf{z}$ of the mechanism satisfies $\|\mathbf{z} - \mathrm{Avg}(\mathcal{S})\| \leq \frac{\Lambda\sqrt{\log(1.25/\delta)}}{\varepsilon n}\left(\sqrt{d} + \sqrt{2\log(1/\beta)}\right)$.*

Remark 2.21. *Theorem 2.19 guarantees differential-privacy whenever two neighboring databases have equal size. However, it can be easily extended to a more general case in which the privacy guarantee also holds in cases of addition and deletion of a point, with essentially the same noise magnitude (e.g., see Appendix A in [NSV16]).*

The following proposition states the following: Assume that $\mathbf{X} \sim \mu + (\mathcal{N}(0,\sigma^2))^d$ for some $\mu \in \mathbb{R}^d$, and let $\mathbf{y} \in \mathbb{R}^d$ such that $\|\mathbf{y} - \mu\|$ is "large enough" (i.e., larger than $\Omega\left(\sigma\sqrt{\log(1/\beta)}\right)$). Then with probability $1-\beta$ (over $\mathbf{X}$) it holds that $\|\mathbf{X} - \mu\| < \|\mathbf{X} - \mathbf{y}\|$. Note that such an argument is trivial when $\|\mathbf{y} - \mu\|$ is at least $\Omega(\sigma\sqrt{d\log(1/\beta)})$, but using a standard projection argument, we can avoid the dependency in d. The proof appears at Appendix B.3 as a special case of Proposition B.6.

Proposition 2.22. *Let $\mathbf{X} \sim \mu + (\mathcal{N}(0,\sigma^2))^d$ and let $\mathbf{y} \in \mathbb{R}^d$ with $\|\mathbf{y} - \mu\| > 2\sqrt{2\log\left(\frac{1}{\beta}\right)} \cdot \sigma$. Then with probability $1-\beta$ (over the choice of $\mathbf{X}$), it holds that $\|\mathbf{X} - \mu\| < \|\mathbf{X} - \mathbf{y}\|$.*

2.2.6 Estimating the Average of Points

As mentioned in Observation 2.20, the Gaussian mechanism (Theorem 2.19) allows for privately estimating the average of points in $B(\mathbf{0},\Lambda) \subseteq \mathbb{R}^d$ within ℓ_2 error of $\approx \frac{\Lambda\sqrt{d}}{\varepsilon n}$. In some cases, we could relax the dependency on Λ. For example, using the following proposition.

Proposition 2.23 (Estimating the Average of Bounded Points in $\mathbb{R}^d$)**.** *Let $\varepsilon \in (0,1)$, $d, \Lambda > 0$ and let $r_{\min} \in [0,\Lambda]$. There exists an efficient (ε,δ)-differentially private algorithm that takes an n-size database $\mathcal{S}$ of points inside the ball $B(\mathbf{0},\Lambda)$ in $\mathbb{R}^d$ and satisfy the following utility guarantee: Let $r > 0$ be the minimal radius of a d-dimensional ball that contains all points in $\mathcal{S}$. Then with probability $1-\beta$, the algorithm outputs $\hat{\mathbf{a}} \in \mathbb{R}^d$ such that*

$$\|\hat{\mathbf{a}} - \mathrm{Avg}(\mathcal{S})\| \leq O\left(\max\{r, r_{\min}\} \cdot \frac{d\sqrt{\log(1/\delta)}}{\varepsilon n}\left(\sqrt{\log(d/\delta)\log(d/\beta)} + \log\left(\frac{\Lambda d}{r_{\min}\beta}\right)\right)\right).$$

The algorithm runs in time $\tilde{O}(dn)$ (ignoring logarithmic factors).

Proposition 2.23 can be seen as a simplified variant of [NSV16]'s private average algorithm. The main difference is that [NSV16] first uses the Johnson Lindenstrauss (JL) transform [JL84] to randomly embed the input points in $\mathbb{R}^{d'}$ for $d' \approx \log n$, and then estimates the average of the points in each axis of $\mathbb{R}^{d'}$. As a result, they manage to save a factor of $\sqrt{d}$ upon Proposition 2.23 (at the cost of paying a factor of $\log n$ instead). However, for simplifying the construction and the implementation, we chose to omit the JL transform step, and we directly estimate the average along each axis of $\mathbb{R}^d$. For completeness, we present the full details of Proposition 2.23 in Appendix A.1.3.

2.2.7 Sub-Sampling

Lemma 2.24 ([BKN10; KLN+11]). *Let $\mathcal{A}$ be an $(\varepsilon^*, \delta^*)$-differentially private algorithm operating on databases of size m. Fix $\varepsilon \leq 1$, and denote $n = \frac{m}{\varepsilon}(3 + \exp(\varepsilon^*))$. Construct an algorithm $\mathcal{B}$ that on an input database $\mathcal{D} = (z_i)_{i=1}^n$, uniformly at random selects a subset $\mathcal{I} \subseteq [n]$ of size m, and executes $\mathcal{A}$ on the multiset $\mathcal{D}_{\mathcal{I}} = (z_i)_{i \in \mathcal{I}}$. Then $\mathcal{B}$ is (ε, δ)-differentially private, where $\delta = \frac{n}{4m} \cdot \delta^*$.*

The following lemma states that switching between sampling with replacement and without replacement has only a small effect on privacy.

Lemma 2.25 ([BNSV15]). *Fix $\varepsilon \leq 1$ and let $\mathcal{A}$ be an (ε, δ)-differentially private algorithm operating on databases of size m. For $n \geq 2m$, construct an algorithm $\mathcal{A}'$ that on input a database $\mathcal{D}$ of size n, subsamples (with replacement) m rows from $\mathcal{D}$, and runs $\mathcal{A}$ on the result. Then $\mathcal{A}'$ is (ε', δ')-differentially private for $\varepsilon' = 6\varepsilon m/n$ and $\delta' = \exp(6\varepsilon m/n) \cdot \frac{4m}{n} \cdot \delta$.*

2.3 Concentration Bounds

Fact 2.26 (Hoeffding's inequality). *Let $X_1, \ldots, X_n$ be independent random variables, each X_i is strictly bounded by the interval $[a_i, b_i]$, and let $\bar{X} = \frac{1}{n}\sum_{i=1}^n X_i$. Then for every $t \geq 0$:*

$$\Pr\left[\left|\bar{X} - \mathrm{E}\left[\bar{X}\right]\right| \geq t\right] \leq 2\exp\left(-\frac{2n^2t^2}{\sum_{i=1}^n (b_i - a_i)^2}\right)$$

Fact 2.27 ([CO13, Theorem 5.3]). *Let $X \sim \mathrm{Bin}(n, p)$, then for all $t \geq 0$:*

1. $\Pr[X \geq \mathrm{E}[X] + t] \leq \exp\left(-\frac{t^2}{2(np + t/3)}\right)$.
2. $\Pr[X \leq \mathrm{E}[X] - t] \leq \exp\left(-\frac{t^2}{2np}\right)$.

3 k-Tuples Clustering

We first introduce a new property of a collection of (unordered) k-tuples $\{\mathbf{x}_1, \ldots, \mathbf{x}_k\} \in (\mathbb{R}^d)^k$, which we call *partitioned by Δ-far balls*.

Definition 3.1 (Δ-far balls). *A set of k balls $\mathcal{B} = \{B_i = B(\mathbf{c}_i, r_i)\}_{i=1}^k$ over $\mathbb{R}^d$ is called **Δ-far balls**, if for every $i \in [k]$ it holds that $\|\mathbf{c}_i - \mathbf{c}_j\| \geq \Delta \cdot \max\{r_i, r_j\}$ (i.e., the balls are relatively far from each other).*

Definition 3.2 (partitioned by Δ-far balls). *A tuple $X \in (\mathbb{R}^d)^k$ is partitioned by a given set of k Δ-far balls $\mathcal{B} = \{B_1, \ldots, B_k\}$, if for every $i \in [k]$ it holds that $|X \cap B_i| = 1$. A multiset of k-tuples $\mathcal{T} \in ((\mathbb{R}^d)^k)^*$ is **partitioned by** $\mathcal{B}$, if all $X \in \mathcal{T}$ are partitioned by $\mathcal{B}$. We say that $\mathcal{T}$ is **partitioned by Δ-far balls** if such a set $\mathcal{B}$ of k Δ-far balls exists.*

In some cases we want to use a notion of *almost* partitioned property of a database of k-tuples $\mathcal{T}$. This is defined below using the additional parameter ℓ.

Definition 3.3 (ℓ-nearly partitioned by Δ-far balls). *A multiset $\mathcal{T} \in ((\mathbb{R}^d)^k)^*$ is **ℓ-nearly partitioned by** a given set of Δ-far balls $\mathcal{B} = \{B_1, \ldots, B_k\}$, if there are at most ℓ tuples in $\mathcal{T}$ that are not partitioned by $\mathcal{B}$. We say that $\mathcal{T}$ is **ℓ-nearly partitioned by Δ-far balls** if such a set of Δ-far balls $\mathcal{B} = \{B_1, \ldots, B_k\}$ exists.*

For a database of k-tuples $\mathcal{T} \in ((\mathbb{R}^d)^k)^n$, we let $\text{Points}(\mathcal{T})$ be the collection of all the points in all the k-tuples in $\mathcal{T}$.

Definition 3.4 (k-tuple clustering)**.** *For $\mathcal{T} = \{\{\mathbf{x}_{1,j}\}_{j=1}^k, \ldots, \{\mathbf{x}_{n,j}\}_{j=1}^k\} \in ((\mathbb{R}^d)^k)^n$, we define $\text{Points}(\mathcal{T}) = \{\mathbf{x}_{i,j}\}_{i\in[n],j\in[k]} \in (\mathbb{R}^d)^{kn}$.*

The following proposition states that if $\mathcal{T}$ is partitioned by Δ-far balls for $\Delta > 2$, then each choice of Δ-far balls that partitions $\mathcal{T}$ induces the same partition.

Proposition 3.5. *Let $\mathcal{T} \in ((\mathbb{R}^d)^k)^*$ be a multiset that is partitioned by a set of Δ-far balls $\mathcal{B} = \{B_1, \ldots, B_k\}$ for $\Delta > 2$. Then for every k-tuple $X = \{\mathbf{x}_1, \ldots, \mathbf{x}_k\} \in \mathcal{T}$ and for every $i \in [k]$, there exists a ball in $\mathcal{B}$ (call it B_i), such that $\text{Points}(\mathcal{T}) \cap B_i = \{\mathbf{x} \in \text{Points}(\mathcal{T}) \colon i = \text{argmin}_{j\in[k]} \|\mathbf{x} - \mathbf{x}_j\|\}$.*

Proof. Let $X = \{\mathbf{x}_1, \ldots, \mathbf{x}_k\} \in \mathcal{T}$, and for every $i \in [k]$ let $B_i = B(\mathbf{c}_i, r_i) \in \mathcal{B}$ be the ball that contains $\mathbf{x}_i$. We prove the proposition by showing that for every i and every $\mathbf{x} \in \text{Points}(\mathcal{T}) \cap B_i$, it holds that $i = \text{argmin}_{j\in[k]} \|\mathbf{x} - \mathbf{x}_j\|$.

In the following, fix $\mathbf{x} \in \text{Points}(\mathcal{T}) \cap B_i$. On the one hand, since $\mathbf{x} \in B_i$, it holds that $\|\mathbf{x} - \mathbf{x}_i\| \leq r_i$. On the other hand, for any $j \neq i$ it holds that

$$\|\mathbf{x} - \mathbf{x}_j\| \geq \|\mathbf{x}_i - \mathbf{x}_j\| - \|\mathbf{x} - \mathbf{x}_i\| > 2r_i - r_i \geq r_i,$$

where the strict inequality holds since B_i, B_j are Δ-far balls for $\Delta > 2$. Namely, we deduce that $\|\mathbf{x} - \mathbf{x}_i\| < \|\mathbf{x} - \mathbf{x}_j\|$, as required. □

We next define $\text{Partition}(\mathcal{T})$ of a database $\mathcal{T} \in ((\mathbb{R}^d)^k)^*$ which is partitioned by Δ-far balls for $\Delta > 2$.

Definition 3.6 ($\text{Partition}(\mathcal{T})$)**.** *Given a multiset $\mathcal{T} \in ((\mathbb{R}^d)^k)^*$ which is partitioned by Δ-far balls for $\Delta > 2$, we define the partition of $\mathcal{T}$, which we denote by $\text{Partition}(\mathcal{T}) = \{\mathcal{P}_1, \ldots, \mathcal{P}_k\}$, by fixing an (arbitrary) k-tuple $X = \{\mathbf{x}_1, \ldots, \mathbf{x}_k\} \in \mathcal{T}$ and setting $\mathcal{P}_i = \{\mathbf{x} \in \text{Points}(\mathcal{T}) \colon i = \text{argmin}_{j\in[k]} \|\mathbf{x} - \mathbf{x}_j\|\}$.*

Note that by Proposition 3.5, the partition is unique (i.e., is independent of the choice of the k-tuple X).

We now define the k-tuple clustering problem.

Definition 3.7 (k-tuple clustering)**.** *The input to the problem is a database $\mathcal{T} \in ((\mathbb{R}^d)^k)^n$ and a parameter $\Delta > 2$. The goal is to output a k-tuple $Y = \{\mathbf{y}_1, \ldots, \mathbf{y}_k\} \in (\mathbb{R}^d)^k$ such that the following holds: If $\mathcal{T}$ is partitioned by Δ-far balls, then for every $i \in [k]$, there exists a cluster in $\text{Partition}(\mathcal{T})$ (call it $\mathcal{P}_i$) such that $\mathcal{P}_i = \{\mathbf{x} \in \text{Points}(\mathcal{T}) \colon i = \text{argmin}_{j\in[k]} \|\mathbf{x} - \mathbf{y}_j\|\}$.*

Namely, in the k-tuple clustering problem, the goal is to output a k-tuple Y that partitions $\mathcal{T}$ correctly. We remark that for applications, we are also interested in the quality of the solution. Namely, how small is the distance between $\mathbf{y}_i$ and $\mathcal{P}_i$, compared to the other clusters in $\text{Partition}(\mathcal{T})$. We also remark that without privacy, the problem is completely trivial, since any k-tuple $X \in \mathcal{T}$ is a good solution by definition.

We next prove that if $\mathcal{T}$ is partitioned by Δ-far balls for $\Delta > 6$, and $\mathcal{B}$ partitions at least one tuple in $\mathcal{T}$, then by partitioning the points in $\text{Points}(\mathcal{T})$ using a single Lloyd step w.r.t. the centers of the balls in $\mathcal{B}$, we obtain exactly $\text{Partition}(\mathcal{T})$.

Proposition 3.8. *Let $\mathcal{B} = \{B_i = B(\mathbf{c}_i, r_i)\}_{i=1}^k$ and $\mathcal{B}' = \{B'_i = B(\mathbf{c}'_i, r'_i)\}_{i=1}^k$ be two sets of Δ-far balls for $\Delta > 6$ s.t. for every $i \in [k]$ it holds that $B_i \cap B'_i \neq \emptyset$. Then for every $i \in [k]$ and every $\mathbf{x} \in B_i \cap B'_i$, it holds that $i = \operatorname{argmin}_{j \in [k]} \|\mathbf{x} - \mathbf{c}_j\|$.*

Proof. Fix $i \in [k]$ and $\mathbf{x} \in B_i \cap B'_i$. If $\mathbf{x} \in B_i$, the proof trivially follows. Therefore, in the following we assume that $\mathbf{x} \notin B_i$, and therefore, $\mathbf{x} \in B'_i$.

Note that on the one hand, it holds that

$$\|\mathbf{x} - \mathbf{c}_i\| \leq \left\|\mathbf{x} - \mathbf{c}'_i\right\| + \left\|\mathbf{c}'_i - \mathbf{c}_i\right\| \leq r'_i + (r_i + r'_i) = 2r'_i + r_i \tag{1}$$

On the other hand, fix $j \neq i$, and note that

$$\begin{aligned} \|\mathbf{x} - \mathbf{c}_j\| &\geq \|\mathbf{c}_i - \mathbf{c}_j\| - \|\mathbf{x} - \mathbf{c}_i\| \\ &> 6\max\{r_i, r_j\} - (2r'_i + r_i) \\ &\geq 5\max\{r_i, r_j\} - 2r'_i, \end{aligned} \tag{2}$$

where the second inequality holds by Equation (1) along with the fact that $\mathcal{B}$ are Δ-far balls for $\Delta > 6$. Therefore, if $\max\{r_i, r_j\} \geq r'_i$, we deduce by Equations (1) and (2) that $\|\mathbf{x} - \mathbf{c}_i\| < \|\mathbf{x} - \mathbf{c}_j\|$. Otherwise (i.e., $\max\{r_i, r_j\} < r'_i$), note that

$$\begin{aligned} \|\mathbf{x} - \mathbf{c}_j\| &\geq \left\|\mathbf{c}'_i - \mathbf{c}'_j\right\| - \left\|\mathbf{x} - \mathbf{c}'_i\right\| - \left\|\mathbf{c}'_j - \mathbf{c}_j\right\| \\ &> 6\max\{r'_i, r'_j\} - r'_i - (r'_i + r_i) \\ &> 3r'_i. \end{aligned} \tag{3}$$

Hence, we deduce by Equations (1) and (3) that $\|\mathbf{x} - \mathbf{c}_i\| < \|\mathbf{x} - \mathbf{c}_j\|$ also in this case, which concludes the proof of the proposition. □

Proposition 3.9. *Let $\mathcal{T} \in ((\mathbb{R}^d)^k)^n$ be a multiset that is partitioned by Δ-far balls for $\Delta > 6$, let $\mathcal{B} = \{B_1, \ldots, B_k\}$ be a set of far balls that partitions at least one k-tuple of $\mathcal{T}$, and let $\mathbf{c}_1, \ldots, \mathbf{c}_k$ be the centers of $B_1, \ldots, B_k$, respectively. In addition, for every $i \in [k]$ let $\mathcal{Q}_i = \{\mathbf{x} \in \text{Points}(\mathcal{T}) \colon i = \operatorname{argmin}_{j \in [k]} \|\mathbf{x} - \mathbf{c}_j\|\}$. Then $\{\mathcal{Q}_1, \ldots, \mathcal{Q}_k\} = \text{Partition}(\mathcal{T})$.*

Proof. Let $X = \{\mathbf{x}_1, \ldots, \mathbf{x}_k\} \in \mathcal{T}$ be the assumed k-tuple that is partitioned by $\mathcal{B}$, let $\mathcal{B}^* = \{B^*_1, \ldots, B^*_k\}$ be a set of Δ-far balls that partitions (all of) $\mathcal{T}$, and assume w.l.o.g. that $\mathbf{x}_i \in B_i \cap B^*_i$ for every $i \in [k]$. Proposition 3.8 yields that for every $i \in [k]$ and $\mathbf{x} \in B^*_i$ it holds that $\mathbf{x} \in \mathcal{Q}_i$, yielding that $B^*_i \cap \text{Points}(\mathcal{T}) \subseteq \mathcal{Q}_i$. Since both sets $\{B^*_i\}_{i=1}^k$ and $\{\mathcal{Q}_i\}_{i=1}^k$ consist of disjoints sets that cover all the points in $\text{Points}(\mathcal{T})$, we conclude that $\{\mathcal{Q}_1, \ldots, \mathcal{Q}_k\} = \text{Partition}(\mathcal{T})$. □

4 Our Algorithms

In this section we present two (ε, δ)-differentially private algorithms for the k-tuple clustering problem: PrivatekAverages and PrivatekNoisyCenters. Algorithm PrivatekAverages attempts to solve the problem by determining the clusters in $\text{Partition}(\mathcal{T})$ and then privately estimating the average of each cluster using the algorithm from Proposition 2.23. Algorithm PrivatekNoisyCenters, on the other hand, does not operate by averaging clusters. Instead, it first selects one of the input tuples $X \in \mathcal{T}$ (in a special way), and then adds a (relatively small) Gaussian noise to this tuple.[2]

[2]We remind that all the tuples in this work are *unordered*, and indeed the privacy analysis of our algorithms relies on it (i.e., the domain of outputs is all the unordered k-tuples, and (ε, δ)-indistinguishability holds for each subset of this domain).

Both algorithms share the same first step, which is to call Algorithm PrivateTestPartition (Figure 2) that privately decides whether $\mathcal{T}$ is ℓ-nearly partitioned by Δ-far balls or not (for small ℓ), and if so, determines (non-privately) a set of Δ-far balls $\mathcal{B} = \{B_1, \ldots, B_k\}$ that ℓ-nearly partitions $\mathcal{T}$. In Section 4.1 we describe Algorithm PrivateTestCloseTuples, which is the main component of PrivateTestPartition. In Section 4.2 we describe PrivateTestPartition and state its properties. Then, in Section 4.3 we describe PrivatekAverages and prove its guarantees, and in Section 4.4 we describe PrivatekNoisyCenters and prove its guarantees.

4.1 Algorithm PrivateTestCloseTuples

In this section we describe Algorithm PrivateTestCloseTuples, which given two multisets of k-tuples $\mathcal{T}_1$ and $\mathcal{T}_2$, privately checks whether the tuples in $\mathcal{T}_1$ are close to the tuples in $\mathcal{T}_2$. The algorithm is described in Figure 1.

Algorithm PrivateTestCloseTuples

Input: Multisets $\mathcal{T}_1 \in ((\mathbb{R}^d)^k)^m$ and $\mathcal{T}_2 \in ((\mathbb{R}^d)^k)^n$, a privacy parameter $\varepsilon_1 \in (0, 1]$ for $\mathcal{T}_1$, a privacy parameters $\varepsilon_2 \in (0, 1]$ for $\mathcal{T}_2$, a confidence parameter $\beta \in (0, 1]$, and a separation parameter $\Delta > 6$.

1. For each $X = \{\mathbf{x}_1, \ldots, \mathbf{x}_k\} \in \mathcal{T}_1$:
   (a) Let $\mathcal{B}_X = \{B_i^X = B(\mathbf{x}_i, r_i^X)\}_{i=1}^k$, where $r_i^X = \frac{1}{\Delta} \cdot \min_{j \neq i} \|\mathbf{x}_i - \mathbf{x}_j\|$.
   (b) Let $\ell_X = |\{Y \in \mathcal{T}_2 \colon Y \text{ is } \textbf{not} \text{ partitioned by } \mathcal{B}_X\}|$.
   (c) Let $\hat{\ell}_X = \ell_X + \mathrm{Lap}\left(\frac{m}{\varepsilon_2}\right)$.
   (d) Set $\mathrm{pass}_X = \begin{cases} 1 & \hat{\ell}_X \leq \frac{m}{\varepsilon_2} \cdot \log\left(\frac{m}{\beta}\right) \\ 0 & \text{otherwise} \end{cases}$.
2. Let $s = \sum_{X \in \mathcal{T}_1} \mathrm{pass}_X$ and compute $\hat{s} \leftarrow s + \mathrm{Lap}\left(\frac{1}{\varepsilon_1}\right)$.
3. If $\hat{s} < m - \frac{1}{\varepsilon_1} \log\left(\frac{1}{\beta}\right)$, set Status = "Failure". Otherwise, set Status = "Success".
4. If Status = "Success" and $\mathrm{pass}_X = 1$ for at least one $X \in \mathcal{T}_1$, let X^* be the first tuple in $\mathcal{T}_1$ with $\mathrm{pass}_{X^*} = 1$ and set $\mathcal{B} = \mathcal{B}_{X^*}$. Otherwise, set $\mathcal{B}$ to be a set of k empty balls.
5. Output $(\mathit{Status}, \mathcal{B})$.

Figure 1: Algorithm PrivateTestCloseTuples for privately checking if Δ-far balls around each k-tuples in $\mathcal{T}_1$ partitions the tuples in $\mathcal{T}_2$.

4.1.1 Properties of PrivateTestCloseTuples

The properties of PrivateTestCloseTuples are summarized by the following claims.

Claim 4.1 (Correctness). *Assume that $\mathcal{T} = \mathcal{T}_1 \cup \mathcal{T}_2$ is partitioned by $(2\Delta+2)$-far balls. Then with probability $1-\beta$, when executing* PrivateTestCloseTuples *on input $\mathcal{T}_1, \mathcal{T}_2, \varepsilon_1, \varepsilon_2, \beta, \Delta$, it outputs ("Success", $\mathcal{B}$), where $\mathcal{B}$ is a set of Δ-far balls that partitions $\mathcal{T}$.*

Proof. We first prove that for every $X \in \mathcal{T}_1$, the set of balls $\mathcal{B}_X = \{B_i^X = B(\mathbf{x}_i, r_i^X)\}_{i=1}^k$ from Step 1a is a set of Δ-far balls that partitions $\mathcal{T}_2$. Fix $X = \{\mathbf{x}_1, \ldots, \mathbf{x}_k\} \in \mathcal{T}_1$, let $\mathcal{B} = \{B_i = B(\mathbf{c}_i, r_i)\}_{i=1}^k$ be a set of $(2\Delta+2)$-far balls that partitions $\mathcal{T}$ (such a set exists by assumption), and assume w.l.o.g. that $\forall i \in [k]\colon \mathbf{x}_i \in B_i$. In addition, recall that $r_i^X = \frac{1}{\Delta} \cdot \min_{j \neq i} \|\mathbf{x}_i - \mathbf{x}_j\|$ (Step 1a), and therefore, by definition it holds that $\mathcal{B}_X$ is a set of Δ-far balls. It is left to prove that it partitions $\mathcal{T}$. Note that for every $i \neq j$ it holds that

$$\begin{aligned}\|\mathbf{x}_i - \mathbf{x}_j\| &\geq \|\mathbf{c}_i - \mathbf{c}_j\| - \|\mathbf{x}_i - \mathbf{c}_i\| - \|\mathbf{x}_j - \mathbf{c}_j\| \\ &> (2\Delta+2) \cdot \max\{r_i, r_j\} - r_i - r_j \\ &\geq 2\Delta \cdot \max\{r_i, r_j\}\end{aligned}$$

Therefore, for every $i \in [k]$, $r_i^X = \frac{1}{\Delta} \cdot \min_{j \neq i} \|\mathbf{x}_i - \mathbf{x}_j\| > 2 \cdot r_i$. Since $\mathbf{x}_i \in B_i$, we conclude that $\mathcal{B}_i \subseteq B_i^X$, which yields that $\mathcal{B}_X$ partitions $\mathcal{T}$.

Therefore, for every $X = \{\mathbf{x}_1, \ldots, \mathbf{x}_k\} \in \mathcal{T}_1$ it holds that ℓ_X, the value from Step 1b, is 0. Hence, by Fact 2.13 and the union bound, with probability $1 - \frac{\beta}{2}$ it holds that $\forall X \in \mathcal{T}_1 :\ \text{pass}_X = 1$, which yields that $s = m$ (where $m = |\mathcal{T}_1|$). When $s = m$, we obtain by Fact 2.13 that with probability $1 - \frac{\beta}{2}$ it holds that $\hat{s} \geq s - \frac{1}{\varepsilon_1}\log(1/\beta) = m - \frac{1}{\varepsilon_1}\log(1/\beta)$, i.e., *Status* = "Success". This concludes the proof of the claim. □

Claim 4.2 (*Status* is ε_1-DP w.r.t. $\mathcal{T}_1$). *Let $\mathcal{T}_1, \mathcal{T}_1' \in ((\mathbb{R}^d)^k)^m$ be two neighboring databases, let $\mathcal{T}_2 \in ((\mathbb{R}^d)^k)^n$, and consider two independent executions* PrivateTestCloseTuples$(\mathcal{T}_1, \mathcal{T}_2)$ *and* PrivateTestCloseTuples$(\mathcal{T}_1', \mathcal{T}_2)$ *(with the same parameters $\varepsilon_1, \varepsilon_2, \beta, \Delta$). Let Status and Status$'$ be the status outcomes of the two executions (respectively). Then Status and Status$'$ are ε_1-indistinguishable.*

Proof. Note that each k-tuple $X \in \mathcal{T}_1$ can affect only the bit pass_X. Therefore, by the properties of the Laplace mechanism (Theorem 2.15) and post-processing (Fact 2.9), it holds that *Status* and *Status*$'$ are ε_1-indistinguishable. □

Claim 4.3 (*Status* is ε_2-DP w.r.t. $\mathcal{T}_2$). *Let $\mathcal{T}_2, \mathcal{T}_2' \in ((\mathbb{R}^d)^k)^n$ be two neighboring databases, let $\mathcal{T}_1 \in ((\mathbb{R}^d)^k)^m$, and consider two independent executions* PrivateTestCloseTuples$(\mathcal{T}_1, \mathcal{T}_2)$ *and* PrivateTestCloseTuples$(\mathcal{T}_1, \mathcal{T}_2')$ *(with the same parameters $\varepsilon_1, \varepsilon_2, \beta$). Let Status and Status$'$ be the status outcomes of the two executions (respectively). Then Status and Status$'$ are ε_2-indistinguishable.*

Proof. For each $X \in \mathcal{T}_1$, let ℓ_X, pass_X and ℓ_X', pass_X' be the values computed in the loop 1 in the two executions (respectively). Since $|\ell_X - \ell_X'| \leq 1$, we obtain by the properties of the Laplace mechanism, along with post-processing, that pass_X and pass_X' are $\frac{\varepsilon_2}{m}$-indistinguishable. Hence, by basic composition (Theorem 2.10) we deduce that $\{\text{pass}_X\}_{X \in \mathcal{T}_1}$ and $\{\text{pass}_X'\}_{X \in \mathcal{T}_1}$ are ε_2-indistinguishable, and we conclude by post-processing that *Status* and *Status*$'$ are ε_2-indistinguishable. □

The following claim states that when PrivateTestCloseTuples$(\mathcal{T}_1, \mathcal{T}_2)$ outputs ("Success", $\mathcal{B}$), then with high probability, $\mathcal{T}_2$ is almost partitioned by $\mathcal{B}$.

Claim 4.4 (On success, $\mathcal{B}$ almost partitions $\mathcal{T}_2$)**.** *Let $\delta > 0$, let $\mathcal{T}_1 \in ((\mathbb{R}^d)^k)^m$ and $\mathcal{T}_1 \in ((\mathbb{R}^d)^k)^n$, and assume that $m > \frac{1}{\varepsilon_1} \cdot (2\log(1/\delta) + \log(1/\beta))$. Consider a random execution of* PrivateTestCloseTuples$(\mathcal{T}_1, \mathcal{T}_2, \varepsilon_1, \varepsilon_2, \beta)$*, and let $(Status, \mathcal{B})$ be the outcome of the execution. Let S be the event that $Status =$ "Success", and let $E \subseteq S$ be the event that $\mathcal{T}_2$ is ℓ-nearly partitioned by $\mathcal{B}$, where $\ell = \frac{m}{\varepsilon_2} \cdot \log\left(\frac{m}{\beta\delta}\right)$. Then the following holds: If $\Pr[S] \geq \delta$, then $\Pr[E \mid S] \geq 1 - \delta$.*

Proof. Let $\{\text{pass}_X\}_{X \in \mathcal{T}_1}$ be the values from Figure 2 in the execution PrivateTestCloseTuples$(\mathcal{T}_1, \mathcal{T}_2, \varepsilon_1, \varepsilon_2, \beta)$, and let W be the event that there exists $X \in \mathcal{T}_1$ with $\text{pass}_X = 1$. Note that

$$\Pr[\neg W \mid S] \leq \frac{\Pr[S \mid \neg W]}{\Pr[S]} \leq \frac{\Pr\left[\text{Lap}(1/\varepsilon_1) > \frac{2}{\varepsilon_1} \cdot \log\left(\frac{1}{\delta}\right)\right]}{\delta} \leq \frac{\delta^2}{2\delta} \leq \frac{\delta}{2},$$

where the second inequality holds since $\Pr[S] \geq \delta$ and since $m - \frac{1}{\varepsilon_1}\log\left(\frac{1}{\beta}\right) > \frac{2}{\varepsilon_1} \cdot \log\left(\frac{1}{\delta}\right)$, and the third one holds by Fact 2.13. Therefore, in the following we prove the claim by showing that

$$\Pr[E \mid W \wedge S] \geq 1 - \frac{\delta}{2} \tag{4}$$

Let X^* be the tuple from Step 4 (it exists when $W \wedge S$ occurs), and recall that $\mathcal{B} = \mathcal{B}_{X^*}$ and that ℓ_{X^*} is the minimal value such that $\mathcal{T}_2$ is ℓ_{X^*}-nearly partitioned by $\mathcal{B}$. Since $\text{pass}_{X^*} = 1$, it holds that $\hat{\ell}_{X^*} = \ell_{X^*} + \text{Lap}(m/\varepsilon_2) \leq \frac{m}{\varepsilon_2} \cdot \log\left(\frac{m}{\beta}\right)$. Equation (4) now follows by the following calculation.

$$\begin{aligned}
\Pr[E \mid W \wedge S] &= \Pr\left[\ell_{X^*} > \frac{m}{\varepsilon_2} \cdot \log\left(\frac{m}{\beta\delta}\right) \mid \hat{\ell}_{X^*} \leq \frac{m}{\varepsilon_2} \cdot \log\left(\frac{m}{\beta}\right)\right] \\
&\leq \Pr\left[\text{Lap}(m/\varepsilon_2) < -\frac{m}{\varepsilon_2} \cdot \log\left(\frac{1}{\delta}\right)\right] \\
&\leq \frac{\delta}{2},
\end{aligned}$$

where the last inequality holds by Fact 2.13. □

4.2 Algorithm PrivateTestPartition

In this section we describe PrivateTestPartition and state its properties. The algorithm is described in Figure 2. In the following, we define m and ε_1 (functions of $n, \varepsilon, \delta, \beta$) that are used by PrivateTestPartition.

Definition 4.5. *Let $m = m(n, \varepsilon, \delta, \beta)$ be the smallest integer that satisfies $m > \frac{1}{\varepsilon_1} \cdot (2\log(1/\delta) + \log(1/\beta))$, where $\varepsilon_1 = \log(\frac{\varepsilon n}{2m} - 3)$.*

The dependence between m and ε_1 for Algorithm PrivateTestPartition is due to the choice of $\mathcal{T}_1$ as an m-size random sample of $\mathcal{T}$. A smaller m allows for a larger value of ε_1 for the same overall privacy, by a sub-sampling argument (e.g., Lemma 2.24). We note that for $n \gg 1/\varepsilon$ and $\beta, \delta \geq \frac{1}{\text{poly}(n)}$, we have $\varepsilon_1 = \Theta(\log n)$, which yields that $m = O(1)$. For smaller values of δ, we obtain that $m = O\left(\frac{\log(1/\delta)}{\log n}\right)$.

Algorithm PrivateTestPartition

Input: A multiset $\mathcal{T} \in ((\mathbb{R}^d)^k)^n$, privacy parameters $\varepsilon, \delta \in (0,1]$, confidence parameter $\beta \in (0,1]$, and separation parameter $\Delta > 6$.

1. Let m and ε_1 be the values from Definition 4.5 w.r.t. $n, \varepsilon, \delta, \beta$, and let $\varepsilon_2 = \varepsilon/2$.
2. Let $\mathcal{T}_1$ be a uniform sample of m k-tuples from $\mathcal{T}$ (without replacement), and let $\mathcal{T}_2 = \mathcal{T}$.
3. Output $(\mathit{Status}, \mathcal{B}) = \text{PrivateTestCloseTuples}(\mathcal{T}_1, \mathcal{T}_2, \varepsilon_1, \varepsilon_2, \beta, \Delta)$.

Figure 2: Algorithm PrivateTestPartition for privately checking if Δ-far balls around each k-tuples in $\mathcal{T}_1$ partitions the tuples in $\mathcal{T}_2$.

4.2.1 Properties of PrivateTestPartition

The following claim is an immediate corollary of Claim 4.1

Claim 4.6 (Correctness)**.** *Assume that $\mathcal{T}$ is partitioned by $(2\Delta+2)$-far balls. Then with probability $1-\beta$, when executing* PrivateTestCloseTuples *on input $\mathcal{T}, \varepsilon, \delta, \beta, \Delta$, it outputs ("Success", $\mathcal{B}$), where $\mathcal{B}$ is a set of Δ-far balls that partitions $\mathcal{T}$.*

The following claim is a corollary of Claims 4.2 and 4.3.

Claim 4.7 (*Status* is private)**.** *Let $\mathcal{T}$ and $\mathcal{T}'$ be two neighboring databases, and consider two independent executions* PrivateTestPartition$(\mathcal{T})$ *and* PrivateTestPartition$(\mathcal{T}')$ *(with the same parameters $\varepsilon, \delta, \beta$). Let Status and Status$'$ be the status outcomes of the two executions (respectively). Then Status and Status$'$ are ε-indistinguishable.*

Proof. As a first step, assume that we have two (different) copies of $\mathcal{T}$, call them $\tilde{\mathcal{T}}_1$ and $\tilde{\mathcal{T}}_2$, where $\mathcal{T}_1$ is chosen from the copy $\tilde{\mathcal{T}}_1$, and $\mathcal{T}_2$ is chosen from the copy $\tilde{\mathcal{T}}_2$, and let $(\tilde{\mathcal{T}}_1', \tilde{\mathcal{T}}_2')$ be a neighboring database of $(\tilde{\mathcal{T}}_1, \tilde{\mathcal{T}}_2)$. If $\tilde{\mathcal{T}}_2$ and $\tilde{\mathcal{T}}_2'$ are neighboring (and $\tilde{\mathcal{T}}_1 = \tilde{\mathcal{T}}_1'$), we obtain by Claim 4.3 that *Status* and *Status$'$* are $\varepsilon/2$-indistinguishable. Therefore, assume that $\tilde{\mathcal{T}}_1$ and $\tilde{\mathcal{T}}_1'$ are neighboring (and $\tilde{\mathcal{T}}_2 = \tilde{\mathcal{T}}_2'$). By Claim 4.2, *Status* and *Status$'$* are ε_1-indistinguishable if the resulting samples $\mathcal{T}_1$ and $\mathcal{T}_1'$ in the two executions are neighboring. Since $\mathcal{T}_1$ is just an m-size sample from $\tilde{\mathcal{T}}_1$, and since $\varepsilon_1 = \log(\frac{\varepsilon n}{2m} - 3)$, we obtain by subsampling argument (Lemma 2.24) that *Status* and *Status$'$* are $\varepsilon/2$-indistinguishable also in this case.

Finally, going back to our case where $\tilde{\mathcal{T}}_1 = \tilde{\mathcal{T}}_2 = \mathcal{T}$, we deduce by the above analysis along with group privacy (of 2) that *Status* and *Status$'$* are ε-indistinguishable. □

The following claim is an immediate corollary of Claim 4.4. It states that when the tests succeed, then w.h.p., $\mathcal{T}$ is ℓ-nearly partitioned by $\mathcal{B}$, for the value of ℓ defined below.

Definition 4.8. *Let $\ell = \ell(n, \varepsilon, \delta, \beta) = \frac{2m}{\varepsilon} \cdot \log\left(\frac{m}{\beta\delta}\right)$, where $m = m(n, \varepsilon, \delta, \beta)$ is the value from Definition 4.5.*

We note that $\ell = O\left(\frac{\log^2(1/\delta)}{\varepsilon \log n}\right)$. When $\beta, \delta \geq 1/\text{poly}(n)$, we have that $\ell = O\left(\frac{1}{\varepsilon}\log n\right)$.

Claim 4.9 (On success, $\mathcal{B}$ almost partitions $\mathcal{T}$)**.** *Let $\mathcal{T} \in ((\mathbb{R}^d)^k)^n$ and $\delta > 0$. Consider a random execution of* PrivateTestPartition$(\mathcal{T}, \varepsilon, \delta, \beta, \Delta)$*, and let $(Status, \mathcal{B})$ be the outcome of the execution. Let S be the event that $Status =$ "Success", and let $E \subseteq S$ be the event that $\mathcal{T}$ is ℓ-nearly partitioned by $\mathcal{B}$, where $\ell = \ell(n, \varepsilon, \delta, \beta)$ is the value from Definition 4.8. Then the following holds: If $\Pr[S] \geq \delta$, then $\Pr[E \mid S] \geq 1 - \delta$.*

Proof. Immediately holds by Claim 4.4 since $\ell = \frac{m}{\varepsilon_2} \cdot \log\left(\frac{m}{\beta\delta}\right)$, and since it holds that $m > \frac{1}{\varepsilon_1} \cdot (2\log(1/\delta) + \log(1/\beta))$ (by definition), as required by Claim 4.4. □

Recall that Algorithm PrivateTestPartition has two outputs: A bit *Status* and a set of balls $\mathcal{B}$. As we stated in Claim 4.7, the bit *Status* preserves privacy. The set of balls $\mathcal{B}$, however, does *not*. Still, in the following sections we use Algorithm PrivateTestPartition as a subroutine in our two main algorithms PrivatekAverages and PrivatekNoisyCenters. To argue about the privacy properties of these algorithms, we rely on the following key property of algorithm PrivateTestPartition.

Claim 4.10. *Let $\mathcal{A}^*$ be an algorithm that gets as input a multiset $\mathcal{T} \in ((\mathbb{R}^d)^k)^n$ and a set of balls $\mathcal{B} = \{B_1, \ldots, B_k\}$, and let $\ell = \ell(n, \varepsilon/2, \delta/4, \beta/2)$ be the value from Definition 4.8. Assume that $\mathcal{A}^*$ has the property that for any neighboring multisets $\mathcal{T}, \mathcal{T}'$ and any sets of Δ-far balls $\mathcal{B}, \mathcal{B}'$ that ℓ-nearly partitions $\mathcal{T}$ and $\mathcal{T}'$ (respectively), it holds that $\mathcal{A}^*(\mathcal{T}, \mathcal{B})$ and $\mathcal{A}^*(\mathcal{T}', \mathcal{B}')$ are $(\varepsilon^*, \delta/4)$-indistinguishable. Let $\mathcal{A}$ be the algorithm that on input $\mathcal{T}$, does the following steps: (1) Compute $(Status, \mathcal{B}) =$ PrivateTestPartition$(\mathcal{T}, \varepsilon/2, \delta/4, \beta/2, \Delta)$, and (2) If $Status =$ "Failure", output $\perp$ and abort, and otherwise output $\mathcal{A}^*(\mathcal{T}, \mathcal{B})$. Then $\mathcal{A}$ is $(\varepsilon/2 + \varepsilon^*, \delta)$-differentially private.*

Proof. Let $\mathcal{T}$ and $\mathcal{T}'$ be two neighboring multisets of size n. In the following we consider two independent executions: $\mathcal{A}(\mathcal{T})$ and $\mathcal{A}(\mathcal{T}')$. In $\mathcal{A}(\mathcal{T})$, let O be the outcome, let S, E be the events from Claim 4.9 w.r.t. the execution of PrivateTestPartition in step (1), and let $(Status, \mathcal{B})$ be the resulting output of PrivateTestPartition. Similarly, let $O', S', E', Status', \mathcal{B}'$ be the events and random variables w.r.t. the execution $\mathcal{A}(\mathcal{T}')$. Let $q = \Pr[S]$ and $q' = \Pr[S']$. By Claim 4.7 and by group privacy (Fact 2.8), *Status* and *Status*$'$ are $\frac{\varepsilon}{2}$-indistinguishable. Therefore, $q \in e^{\pm\varepsilon/2} \cdot q'$. Recall that $\tilde{\mathcal{A}}$ outputs $\perp$ and aborts whenever $Status =$ "Failure", and therefore, $\Pr[O = \perp] = 1 - q$ and $\Pr[O' = \perp] = 1 - q'$. If $q < \frac{\delta}{2}$ then $q' < e^{\varepsilon/2} \cdot \frac{\delta}{2} \leq \delta$ (recall that $\varepsilon \leq 1$), and therefore, $\Pr[O = \perp], \Pr[O' = \perp] \geq 1 - \delta$. This means that O and O' are $(0, \delta)$-indistinguishable in the case that $q < \frac{\delta}{2}$ (by Lemma 2.4). Similarly, it holds that O and O' are $(0, \delta)$-indistinguishable when $q' < \frac{\delta}{2}$. Hence, in the rest of the analysis we assume that $q, q' \geq \frac{\delta}{2}$.

By Fact 2.7, since $O|_{\neg S} \equiv O'|_{\neg S'}$ (both outcomes equal to $\perp$ when $Status = Status' =$ "Failure") and since $\Pr[S] \in e^{\pm\varepsilon/2} \cdot \Pr[S']$, it is enough to prove that $O|_S$ and $O'|_{S'}$ are $(\varepsilon^*, \frac{\delta}{2})$-indistinguishable. Furthermore, since $\Pr[E \mid S], \Pr[E' \mid S'] \geq 1 - \frac{\delta}{4}$ (by Claim 4.9), we deduce by Fact 2.5 that it is enough to prove that $O|_E$ and $O'|_{E'}$ are $(\varepsilon^*, \frac{\delta}{4})$-indistinguishable, meaning that we only need to prove indistinguishability in the case that $\mathcal{T}$ and $\mathcal{T}'$ are ℓ-nearly partitioned by $\mathcal{B}$ and $\mathcal{B}'$, respectively. The proof of the claim now follows since $\mathcal{A}^*(\mathcal{T}, \mathcal{B})|_E$ and $\mathcal{A}^*(\mathcal{T}', \mathcal{B}')|_{E'}$ are $(\varepsilon^*, \delta/4)$-indistinguishable by the assumption on the algorithm $\mathcal{A}^*$. □

Remark 4.11. *Note that* PrivateTestPartition *runs in time $O(mdk^2n) = \tilde{O}(dk^2n)$ since for each iteration $X \in \mathcal{T}_1$ in* PrivateTestCloseTuples*, Step 1a takes $O(dk^2)$ time, and Step 1b takes $O(dk^2n)$ times.*

4.3 Algorithm PrivatekAverages

In this section we describe and state the properties of Algorithm PrivatekAverages which is our first algorithm for k-tuple clustering. The algorithm is described in Figure 3.

Algorithm PrivatekAverages

Input: A multiset $\mathcal{T} \in \left(B(0,\Lambda)^k\right)^n \subseteq ((\mathbb{R}^d)^k)^n$, privacy parameters $\varepsilon, \delta \in (0,1]$, a confidence parameter $\beta \in (0,1]$, and a lower bound on the radii $r_{\min} \in [0,\Lambda]$.

1. Compute $(Status, \mathcal{B} = \{B_1, \ldots, B_k\}) = \text{PrivateTestPartition}(\mathcal{T}, \varepsilon/2, \delta/4, \beta/2, \Delta)$ for $\Delta = 7$.
2. If $Status =$ "Failure", output $\perp$ and abort.
3. Let $\mathbf{c}_1, \ldots, \mathbf{c}_k$ be the centers of $B_1, \ldots, B_k$ (respectively), and let $\mathcal{Q}_i = \{\mathbf{x} \in \text{Points}(\mathcal{T}) \colon i = \text{argmin}_{j\in[k]} \|\mathbf{x} - \mathbf{c}_j\|\}$.
4. Let $\ell = \ell(n, \varepsilon/2, \delta/4, \beta/2)$ be the value from Definition 4.8.
5. For $i = 1$ to k:
   (a) Compute a noisy average $\hat{\mathbf{a}}_i$ of $\mathcal{Q}_i$ by executing the algorithm from Proposition 2.23 with parameters $\Lambda, r_{\min}, \hat{\beta} = \frac{\beta}{2k}, \hat{\varepsilon} = \frac{\varepsilon}{4k(\ell+1)}, \hat{\delta} = \frac{\delta}{8k\exp(\varepsilon/2)(\ell+1)}$.
6. Output $\hat{A} = \{\hat{\mathbf{a}}_1, \ldots, \hat{\mathbf{a}}_k\}$.

Figure 3: Algorithm PrivatekAverages for privately finding the k centers.

4.3.1 Properties of PrivatekAverages

The properties of PrivatekAverages are given in the following theorems.

Theorem 4.12 (Utility of PrivatekAverages)**.** *Let $d, k, \Lambda > 0$, $r_{\min} \in [0,\Lambda]$, $\varepsilon, \delta, \beta \in (0,1]$, and let $\mathcal{T} \in \left(B(0,\Lambda)^k\right)^n \subseteq ((\mathbb{R}^d)^k)^n$. Assume that $\mathcal{T}$ is partitioned by Δ-far balls for $\Delta = 16$, let $\{\mathcal{P}_1, \ldots, \mathcal{P}_k\} = \text{Partition}(\mathcal{T})$ (according to Definition 3.6), let r_i the radius of the ball that contains $\mathcal{P}_i$. Then there exists a universal constant $\lambda > 0$ such that w.p. $\geq 1-\beta$, algorithm* PrivatekAverages *on inputs $\mathcal{T}, r_{\min}, \varepsilon, \delta, k$, outputs k points $\hat{A} = \{\hat{\mathbf{a}}_1, \ldots, \hat{\mathbf{a}}_k\}$ such that for any $i \in [k]$, there exists a cluster (call it $\mathcal{P}_i$) with*

$$\|\hat{\mathbf{a}}_i - \text{Avg}(\mathcal{P}_i)\| \leq \max\{r_i, r_{\min}\} \cdot \frac{\lambda dk\ell\sqrt{\log\left(\frac{k\ell}{\delta}\right)}}{\varepsilon n}\left(\sqrt{\log\left(\frac{dk\ell}{\delta}\right)\log\left(\frac{dk\ell}{\beta}\right)} + \log\left(\frac{\Lambda dk}{r_{\min}\beta}\right)\right)$$

where $\ell = \ell(n, \frac{\varepsilon}{2}, \frac{\delta}{4}, \frac{\beta}{2})$ is the value from Definition 4.8.

We note that when $\min\{r_i\} \geq r_{\min}$, $\beta = O(1)$ and $\delta \geq \frac{1}{\text{poly}(n)}$, it holds that $\ell = O\left(\frac{1}{\varepsilon}\log n\right)$, and therefore, $\|\hat{\mathbf{a}}_i - \text{Avg}(\mathcal{P}_i)\| \leq \tilde{O}\left(\frac{dk\log^{1.5} n\left(\sqrt{\log n}+\log\left(\frac{\Lambda}{r_{\min}}\right)\right)}{\varepsilon^2 n}\right)\cdot r_i$. For smaller values of δ, it holds that $\ell = O\left(\frac{\log^2(1/\delta)}{\varepsilon\log n}\right)$, and we obtain that $\|\hat{\mathbf{a}}_i - \text{Avg}(\mathcal{P}_i)\| \leq \tilde{O}\left(\frac{dk\cdot\log^{2.5}(1/\delta)\left(\sqrt{\log(1/\delta)}+\log\left(\frac{\Lambda}{r_{\min}}\right)\right)}{\varepsilon^2 n\log n}\right)\cdot r_i$.

Proof. Consider a random execution of PrivatekAverages$(\mathcal{T},\varepsilon,\delta,\beta)$, and let $\tilde{\beta} = \frac{\beta}{2}$ be the value from Step 1. Since $\mathcal{T}$ is partitioned by $(2\cdot 7+2)$-far balls, Claim 4.6 yield that with probability $1-\beta/2$, the set $\mathcal{B} = \{B_1,\ldots,B_k\}$ (computed in Step 1) partitions $\mathcal{T}$. In the following we assume that this event occurs. Let $\{\mathcal{Q}_1,\ldots,\mathcal{Q}_k\}$ be the clusters that were computed in Step 3 of PrivatekAverages. By Proposition 3.9, it holds that $\{\mathcal{P}_1,\ldots,\mathcal{P}_k\} = \{\mathcal{Q}_1,\ldots,\mathcal{Q}_k\}$. Now let r_i be the radius of the ball that contains $\mathcal{P}_i$. The proof now follows by the utility guarantee of Proposition 2.23 for each $i\in[k]$ with the parameters defined in Step 5a of the algorithm. □

Theorem 4.13 (Privacy of PrivatekAverages)**.** *Let $d,k,\Lambda>0$, $r_{\min}\in[0,\Lambda]$, $\varepsilon,\delta,\beta\in(0,1]$. Then for any integer $n\geq 2\cdot\ell(n,\varepsilon/2,\delta/4,\beta/2)+2$ (where ℓ is the function from Definition 4.8), algorithm* PrivatekAverages$(\cdot,\varepsilon,\delta,\beta,r_{\min})$ *is (ε,δ)-differentially private for databases $\mathcal{T}\in(B(\mathbf{0},\Lambda)^k)^n\subseteq((\mathbb{R}^d)^k)^n$.*

Proof. Let $\mathcal{T}$ and $\mathcal{T}'$ be two neighboring multisets of size n. In the following we consider two independent executions: PrivatekAverages$(\mathcal{T})$ and PrivatekAverages$(\mathcal{T}')$ (both with the same parameters $r_{\min},\varepsilon,\delta,\beta$). In PrivatekAverages$(\mathcal{T})$, let O be the output, and let $\mathcal{B} = \{B_1,\ldots,B_k\}, \mathcal{Q}_1,\ldots,\mathcal{Q}_k$ be the values from Figure 3. Similarly, we let $O', \mathcal{B}' = \{B'_1,\ldots,B'_k\}, \mathcal{Q}'_1,\ldots,\mathcal{Q}'_k$ be the these values w.r.t. the execution PrivatekAverages$(\mathcal{T}')$. By Claim 4.10, if we treat Step 3 to 6 as algorithm $\mathcal{A}^*$ of the claim, it is enough to prove that $O=\hat{A}$ and $O'=\hat{A}'$ are $(\varepsilon/2,\delta/4)$-indistinguishable only in the case that $\mathcal{T}$ and $\mathcal{T}'$ are ℓ-nearly partitioned by $\mathcal{B}$ and $\mathcal{B}'$, respectively. In addition, note that since $\mathcal{T}$ and $\mathcal{T}'$ are neighboring, and since $n\geq 2\ell+2$, there exists at least one k-tuple that is partitioned by both $\mathcal{B}$ and $\mathcal{B}'$, yielding that for each ball $B_i\in\mathcal{B}$, there exists a balls in $\mathcal{B}'$ (call it B'_i), such that $B_i\cap B'_i\neq\emptyset$. Since $\mathcal{B}$ and $\mathcal{B}'$ are sets of Δ-far balls for $\Delta=7$, Proposition 3.8 yields that for every $\mathbf{x}\in B_i$ (or B'_i), it holds that $i = \text{argmin}_{j\in[k]}\|\mathbf{x}-\mathbf{c}_j\| = \text{argmin}_{j\in[k]}\left\|\mathbf{x}-\mathbf{c}'_j\right\|$. Therefore, in the two executions, $\{\mathcal{Q}_1,\ldots,\mathcal{Q}_k\}$ and $\{\mathcal{Q}'_1,\ldots,\mathcal{Q}'_k\}$ agree on all the points of all the common $(n-1)$ k-tuples of $\mathcal{T}$ and $\mathcal{T}'$ that are partitioned by $\mathcal{B}$ or $\mathcal{B}'$. Since there are at least $k\cdot(n-1-\ell)$ such points, we deduce that there are at most $k(\ell+1)$ points that the partitions $\{\mathcal{Q}_1,\ldots,\mathcal{Q}_k\}$ and $\{\mathcal{Q}'_1,\ldots,\mathcal{Q}'_k\}$ disagree on.

In the following, let s_i be the number of points that the multisets $\mathcal{Q}_i$ and $\mathcal{Q}'_i$ differ by. Note that each point that the partitions disagree on contributes at most 1 to at most two of the s_i's. Hence, $\sum_{i=1}^k s_i\leq 2k(\ell+1)$.

By the privacy guarantee of Proposition 2.23 (see Remark A.8) along with group privacy (Fact 2.8), for each $i\in[k]$, the resulting noisy averages $\hat{\mathbf{a}}_i$ of the execution PrivatekAverages$(\mathcal{P})$, and the resulting $\hat{\mathbf{a}}'_i$ of the execution PrivatekAverages$(\mathcal{P}')$, which computed in Step 5a, are $(\frac{\varepsilon s_i}{4k(\ell+1)},\frac{\delta s_i}{8k(\ell+1)})$-indistinguishable. Thus, by basic composition (Theorem 2.10) we deduce that $\{\hat{\mathbf{a}}_1,\ldots,\hat{\mathbf{a}}_k\}$ and $\{\hat{\mathbf{a}}'_1\ldots,\hat{\mathbf{a}}'_k\}$ are $(\frac{\varepsilon\sum_{i=1}^k s_i}{4k(\ell+1)},\frac{\delta\sum_{i=1}^k s_i}{8k(\ell+1)}) = (\frac{\varepsilon}{2},\frac{\delta}{4})$-indistinguishable, as required.

□

Remark 4.14 (Run time of PrivatekAverages)**.** *Step 1 of* PrivatekAverages *takes* $\tilde{O}(dk^2n)$ *time (see Remark 4.11). By Proposition 2.23, the* k *executions of Step 5a takes time* $\sum_{i=1}^{k}\tilde{O}(|\mathcal{T}_i|) = \tilde{O}(dkn)$ *(ignoring logarithmic factors). Overall, the running time of* PrivatekAverages *is* $\tilde{O}(dk^2n)$.

4.3.2 Reducing the dependency in the dimension d

When the dimension d is large, algorithm PrivatekAverages estimates the average of each cluster $\mathcal{P}_i$ with radius r_i up to an additive error of $\tilde{O}\left(\frac{d}{n}\cdot r_i\right)$ (ignoring $\mathrm{poly}(k, 1/\varepsilon)$ and $\mathrm{polylog}(n, \delta, \beta, \Lambda, 1/r_{\min})$ factors). This means that if we want an additive error which is much smaller than r_i, we must take $n \gg d$, and in some settings, such a dependency in the dimension might be expensive. Yet, we can easily reduce the d into $\sqrt{d}$ by replacing in Step 5a the average algorithm of Proposition 2.23 by the average algorithm of [NSV16] that uses the JL transform for saving a factor of $\sqrt{d}$ (see the last paragraph in Section 2.2.6 for more details).

4.4 Algorithm PrivatekNoisyCenters

In this section we describe Algorithm PrivatekNoisyCenters which is our second algorithm for k-tuple clustering. The algorithm is described in Figure 4.

Algorithm PrivatekNoisyCenters

Input: A multiset $\mathcal{T} \in ((\mathbb{R}^d)^k)^n$, privacy parameters $\varepsilon \in (0,1]$, $\delta \in (0, 1/2]$, confidence parameter $\beta \in (0,1]$, and a separation parameter $\Delta \gg 6$.

1. Compute $(\mathit{Status}, \mathcal{B} = \{B_1, \ldots, B_k\}) = \mathrm{PrivateTestPartition}(\mathcal{T}, \varepsilon/2, \delta/4, \beta/2, \Delta)$.
2. If Status = "Failure", output $\perp$ and abort.
3. Let $\mathbf{c}_1, \ldots, \mathbf{c}_k$ be the centers of $B_1, \ldots, B_k$ (respectively).
4. For $i = 1$ to k:
   (a) Let $\lambda_i = \frac{2}{\Delta}(1+\gamma_i)\min_{j\neq i}\|\mathbf{c}_i - \mathbf{c}_j\|$ where $\gamma_i = \frac{4}{\Delta-2}\cdot\left(\mathrm{Lap}(4k/\varepsilon) + \frac{4k}{\varepsilon}\log(4k/\delta) + 1\right)$.
   (b) Let $\hat{\mathbf{c}}_i = \mathbf{c}_i + (\mathcal{N}(\mathbf{0}, \sigma_i^2))^d$, where $\sigma_i = \frac{4k\lambda_i}{\varepsilon}\sqrt{2\log(10k/\delta)}$.
5. Output $\hat{C} = \{\hat{\mathbf{c}}_1, \ldots, \hat{\mathbf{c}}_k\}$.

Figure 4: Algorithm PrivatekNoisyCenters for privately finding the k centers.

4.4.1 Properties of PrivatekNoisyCenters

The properties of PrivatekNoisyCenters are given in the following theorems.

Theorem 4.15 (Utility of PrivatekNoisyCenters)**.** *Let* $d, k > 0$, $\varepsilon, \beta, \delta \in (0,1]$ *with* $\delta < \beta$, *let* $\mathcal{T} \in ((\mathbb{R}^d)^k)^n$, *and assume that* $\mathcal{T}$ *is partitioned by* $(2\Delta+2)$*-far balls, for* $\Delta = \Omega\left(\frac{k\log(k/\delta)\sqrt{\log(k/\beta)}}{\varepsilon}\right)$.

Then when executing PrivatekNoisyCenters$(\mathcal{T}, \varepsilon, \delta, \beta, \Delta)$*, with probability* $1-\beta$*, the output* $\hat{C} = \{\hat{\mathbf{c}}_1, \ldots, \hat{\mathbf{c}}_k\}$ *satisfy for every* i *and* $j \neq i$ *that* $\|\hat{\mathbf{c}}_i - \mathbf{c}_i\| < \|\hat{\mathbf{c}}_i - \mathbf{c}_j\|$.

We remark that the k factor in the Δ in Theorem 4.15, comes from applying basic composition (Theorem 2.10) over the k noisy centers $\hat{C}$. This however can be reduced to $\tilde{O}(\sqrt{k})$ factor by applying advanced composition (Theorem 2.11).

Proof. By the union bound on all the choices of γ_i, w.p. $1-\delta/8 \geq 1-\beta/8$, for each $i \in [k]$ it holds that $\min_{j\neq i}\|\mathbf{c}_i - \mathbf{c}_j\| \geq \Omega\Big(\sqrt{\log(k/\beta)}\Big)\cdot\sigma_i$. Therefore, for every $i \neq j$ we can apply Proposition 2.22 with $\mu = \mathbf{c}_i$ and $\mathbf{y} = \mathbf{c}_j$ to obtain that with proper choices of the constants in Δ, with probability $1 - \frac{\beta}{2k^2}$ it holds that $\|\hat{\mathbf{c}}_i - \mathbf{c}_i\| < \|\hat{\mathbf{c}}_i - \mathbf{c}_j\|$. By the union bound over all $i \neq j$ we deduce that with probability $1-\beta/2$ this holds for every $i \neq j$, as required. □

Theorem 4.16 (Privacy of PrivatekNoisyCenters)**.** *Let* $d, k > 0$*,* $\varepsilon, \beta \in (0,1]$*,* $\delta \in (0, 1/2]$*,* $\Delta > 6$*. Then for any integer* $n \geq 2 \cdot \ell(n, \varepsilon/2, \delta/4, \beta/2) + 2$ *(where* ℓ *is the function from Definition 4.8),* PrivatekNoisyCenters$(\cdot, \varepsilon, \delta, \beta, \Delta)$ *is* $(\varepsilon + \delta/4, \delta)$*-differentially private for databases* $\mathcal{T} \in ((\mathbb{R}^d)^k)^n$.

Proof. Let $\mathcal{T}$ and $\mathcal{T}'$ be two neighboring multisets of size n. In the following we consider two independent executions: PrivatekNoisyCenters$(\mathcal{T})$ and PrivatekNoisyCenters$(\mathcal{T}')$ (both with the same parameters $r_{\min}, \varepsilon, \delta, \beta$). In PrivatekNoisyCenters$(\mathcal{T})$, let O be the output, and let $\mathcal{B} = \{B_i = B(\mathbf{c}_i, r_i)\}_{i=1}^k$ be the Δ-far balls from Figure 3. Similarly, we let $O', \mathcal{B}' = \{B'_i = B(\mathbf{c}'_i, r'_i)\}_{i=1}^k$ be the these values w.r.t. the execution PrivatekNoisyCenters$(\mathcal{T}')$. By Claim 4.10, it is enough to prove that the resulting outputs $O = \tilde{C}$ and $O' = \tilde{C}'$ of Steps 3 to 5 are $(\varepsilon/2 + \delta/4, \delta/4)$-indistinguishable only in the case that $\mathcal{T}$ and $\mathcal{T}'$ are ℓ-nearly partitioned by $\mathcal{B}$ and $\mathcal{B}'$, respectively. Since $2\ell \leq n-2$ and since $\mathcal{T}$ and $\mathcal{T}'$ are neighboring, there must exists a k-tuple $X = \{\mathbf{x}_1, \ldots, \mathbf{x}_k\} \in \mathcal{T}$ that is partitioned by both $\mathcal{B}$ and $\mathcal{B}'$. In the rest of the analysis we assume (w.l.o.g.) that $\mathbf{x}_i \in B_i \cap B'_i$ for every $i \in [k]$.

In the following, we prove that for every $i \in [k]$ it holds that $\min_{j\neq i}\|\mathbf{c}_i - \mathbf{c}_j\|$ is close to $\min_{j\neq i}\Big\|\mathbf{c}'_i - \mathbf{c}'_j\Big\|$. For every $i \neq j$ it holds that

$$\begin{aligned}
\|\mathbf{c}_i - \mathbf{c}_j\| &\leq \big\|\mathbf{c}_i - \mathbf{c}'_i\big\| + \big\|\mathbf{c}_j - \mathbf{c}'_j\big\| + \big\|\mathbf{c}'_i - \mathbf{c}'_j\big\| \\
&\leq \|\mathbf{c}_i - \mathbf{x}_i\| + \big\|\mathbf{c}'_i - \mathbf{x}_i\big\| + \|\mathbf{c}_j - \mathbf{x}_j\| + \big\|\mathbf{c}'_j - \mathbf{x}_j\big\| + \big\|\mathbf{c}'_i - \mathbf{c}'_j\big\| \\
&\leq r_i + r'_i + r_j + r'_j + \big\|\mathbf{c}'_i - \mathbf{c}'_j\big\| \\
&\leq \frac{2}{\Delta}\|\mathbf{c}_i - \mathbf{c}_j\| + \frac{2}{\Delta}\big\|\mathbf{c}'_i - \mathbf{c}'_j\big\| + \big\|\mathbf{c}'_i - \mathbf{c}'_j\big\|.
\end{aligned}$$

Therefore,

$$\|\mathbf{c}_i - \mathbf{c}_j\| \leq \frac{\Delta+2}{\Delta-2}\big\|\mathbf{c}'_i - \mathbf{c}'_j\big\| = \left(1 + \frac{4}{\Delta-2}\right)\big\|\mathbf{c}'_i - \mathbf{c}'_j\big\|.$$

Now let $i \in [k]$, and let $s = \operatorname{argmin}_{j\neq i}\|\mathbf{c}_i - \mathbf{c}_j\|$ and $t = \operatorname{argmin}_{j\neq i}\Big\|\mathbf{c}'_i - \mathbf{c}'_j\Big\|$. We deduce that

$$\min_{j\neq i}\|\mathbf{c}_i - \mathbf{c}_j\| = \|\mathbf{c}_i - \mathbf{c}_s\| \leq \|\mathbf{c}_i - \mathbf{c}_t\| \leq \left(1 + \frac{4}{\Delta-2}\right)\big\|\mathbf{c}'_i - \mathbf{c}'_t\big\| = \left(1 + \frac{4}{\Delta-2}\right)\cdot \min_{j\neq i}\big\|\mathbf{c}'_i - \mathbf{c}'_j\big\| \tag{5}$$

Similarly, it holds that $\min_{j\neq i}\left\|\mathbf{c}'_i-\mathbf{c}'_j\right\| \leq \left(1+\frac{4}{\Delta-2}\right)\cdot\min_{j\neq i}\|\mathbf{c}_i-\mathbf{c}_j\|$. Therefore, by the properties of the laplace mechanism, we deduce that for each i, the values of λ_i and λ'_i are $\frac{\varepsilon}{4k}$-indistinguishable, and by basic composition we deduce that $\{\lambda_i\}_{i=1}^k$ and $\{\lambda'_i\}_{i=1}^k$ are all together $\varepsilon/4$-indistinguishable.

In the following, let L be the event that $\forall i\in[k]:\ \gamma_i \geq \frac{4}{\Delta-2}$, and L' be the event that $\forall i\in[k]:\ \gamma'_i \geq \frac{4}{\Delta-2}$. By Fact 2.13 and the union bound, it holds that $\Pr[L],\Pr[L'] \leq \delta/8$. Therefore, by Fact 2.5, it is enough to prove that $\tilde{C}|_L$ and $\tilde{C}'|_{L'}$ are $(\varepsilon/2+\delta/4,\delta/8)$-indistinguishable.

First, by Fact 2.6, we deduce that $\{\lambda_i\}_{i=1}^k|_L$ and $\{\lambda'_i\}_{i=1}^k|_{L'}$ are $(\varepsilon/4+\delta/4)$-indistinguishable. We now continue with the analysis assuming that $\lambda_i=\lambda'_i$ for all $i\in[k]$. Note that for every i it holds that

$$\begin{aligned}\left\|\mathbf{c}_i-\mathbf{c}'_i\right\| &\leq \|\mathbf{c}_i-\mathbf{x}_i\|+\left\|\mathbf{c}'_i-\mathbf{x}_i\right\| \\ &\leq r_i+r'_i \\ &\leq \frac{1}{\Delta}\cdot\left(\min_{j\neq i}\|\mathbf{c}_i-\mathbf{c}_j\|+\min_{j\neq i}\left\|\mathbf{c}'_i-\mathbf{c}'_j\right\|\right) \\ &\leq \lambda_i,\end{aligned}$$

where the last inequality holds by Equation (5) (assuming that L occurs). Therefore, by the properties of the Gaussian Mechanism (Theorem 2.19), we deduce that for each i, $\hat{\mathbf{c}}_i$ and $\hat{\mathbf{c}}'_i$ are $(\frac{\varepsilon}{4k},\frac{\delta}{8k})$-indistinguishable, and by basic composition (Theorem 2.10) we deduce that $\hat{C}$ and $\hat{C}'$ are $(\frac{\varepsilon}{4},\frac{\delta}{8})$-indistinguishable (assuming that $\lambda_i=\lambda'_i$ for all $i\in[k]$). Finally, recall that $\{\lambda_i\}_{i=1}^k|_L$ and $\{\lambda'_i\}_{i=1}^k|_{L'}$ are $(\varepsilon/4+\delta/4)$-indistinguishable, and therefore, we conclude by adaptive composition (Theorem 2.10) that $\hat{C}$ and $\hat{C}'$ are $(\varepsilon/2+\delta/4,\delta/8)$-indistinguishable. □

Remark 4.17 (Run time of PrivatekNoisyCenters)**.** *Step 1 of* PrivatekNoisyCenters *takes* $\tilde{O}(dk^2n)$ *time (see Remark 4.11). The foor-loop in Step 4 only takes* $O(dkn)$ *time. Overall, the running time of* PrivatekNoisyCenters *is* $\tilde{O}(dk^2n)$.

5 k-Means Clustering

In this section we present our first application of k-tuples clustering, which is an (ε,δ)-differentially private k-means approximation algorithm PrivatekMeans with utility guarantee that holds when the input is stable in the sense that we will define. We first start with preliminaries about k-means clustering.

5.1 Preliminaries

For a multiset $\mathcal{P}\in(\mathbb{R}^d)^*$ and a k-tuple of centers $C=\{\mathbf{c}_1,\ldots,\mathbf{c}_k\}\in(\mathbb{R}^d)^k$, we denote $\mathrm{COST}_{\mathcal{P}}(C) := \sum_{\mathbf{x}\in\mathcal{P}}\min_{i\in[k]}\|\mathbf{x}-\mathbf{c}_i\|^2$ and denote $\mathrm{OPT}_k(\mathcal{P}) := \min_{C\in(\mathbb{R}^d)^k}\mathrm{COST}_{\mathcal{P}}(C)$.

The following proposition states that given a multiset $\mathcal{P}\in(\mathbb{R}^d)^n$ and a ω-approximation algorithm $\mathcal{A}$ for k-means, then when sampling m i.i.d. points from $\mathcal{P}$ and executing $\mathcal{A}$ on these points, then with probability $1-\beta$ we obtain k centers with cost $\approx \omega\mathrm{OPT}_k(\mathcal{P})$ (up to a small additive error that depends on m and β). The proof appears at Appendix B.1.

Proposition 5.1. *Let $\mathcal{P}$ be a multiset of n points in $\mathcal{B}(\mathbf{0},\Lambda)\subseteq\mathbb{R}^d$ and let $\mathcal{A}$ be an ω-approximation algorithm for k-means. Consider the following random execution: (1) Construct a multiset $\mathcal{S}$ of s*

i.i.d. samples from $\mathcal{P}$, (2) Compute $\tilde{C} = \mathcal{A}(\mathcal{S}, k)$. Then for every $\beta > 0$, with probability $1 - \beta$ it holds that

$$\mathrm{COST}_{\mathcal{P}}(\tilde{C}) \leq \omega \cdot \mathrm{OPT}_k(\mathcal{P}) + \xi(s, \beta),$$

where $\xi(s,\beta) = 4\left(M(s,\beta) + \sqrt{M(s,\beta) \cdot \omega \mathrm{OPT}_k(\mathcal{P})}\right)$ *for* $M(s,\beta) := 25\Lambda^2 kd \log\left(\frac{2nd}{\beta}\right) \cdot \frac{n}{s}$.

The following proposition states that given a multiset of points $\mathcal{P}$ and given two k-tuples of centers $C = \{\mathbf{c}_1, \ldots, \mathbf{c}_k\}$ and $C' = \{\mathbf{c}'_1, \ldots, \mathbf{c}'_k\}$ such that each $\mathbf{c}'_i$ is relatively close to a unique center $\mathbf{c}_i$ in C, then by clustering the points according to C' and performing a single Lloyd step we get new centers whose k-means cost is almost bounded by $\mathrm{COST}_{\mathcal{P}}(C)$. The proof appears at Appendix B.2.

Proposition 5.2. *Let $k \in \mathbb{N}$ and $\gamma \in [0, 1/8]$. Let $\mathcal{P} \in (\mathbb{R}^d)^*$, let $C = \{\mathbf{c}_1, \ldots, \mathbf{c}_k\}$ and $C' = \{\mathbf{c}'_1, \ldots, \mathbf{c}'_k\}$ be two k-tuples of centers in $\mathbb{R}^d$ such that for every $i \in [k]$ it holds that $\|\mathbf{c}'_i - \mathbf{c}_i\| \leq \gamma \cdot D_i$, where $D_i = \min_{j \neq i} \|\mathbf{c}_i - \mathbf{c}_j\|$. In addition, for every $i \in [k]$ let $\mathcal{P}_i$ be the multiset of all points in $\mathcal{P}$ that $\mathbf{c}'_i$ is closest to them in C'. Then*

$$\sum_{i=1}^{k} \mathrm{OPT}_1(\mathcal{P}_i) \leq (1 + 32\gamma)\mathrm{COST}_{\mathcal{P}}(C).$$

5.2 Private k-Means Under Stability Assumption

In this section we describe our private algorithm PrivatekMeans for approximation the k-means when the input is stable in the sense that we will define next. The idea is the following: Fix a database $\mathcal{P} \in (\mathbb{R}^d)^n$, parameters $s, t \in \mathbb{N}$ and a (non-private) k-means approximation algorithm $\mathcal{A}$. Now execute $\mathcal{A}$ on s i.i.d. samples from $\mathcal{P}$, and repeat this process t times. Consider the event (over this process) that all the t sets of k centers are almost located at the same positions. More formally, consider a random execution of GenCenters$(\mathcal{P}, k, s, t; \mathcal{A})$ (Figure 5). For a k-tuple of centers $C = \{\mathbf{c}_1, \ldots, \mathbf{c}_k\} \in (\mathbb{R}^d)^k$ and a small stability parameter $\gamma > 0$ (say, $\gamma = 0.01$), let E_C^γ be the event that is defined below.

Definition 5.3 (Event E_C^γ over a random sampling of GenCenters)**.** *Let E_C^γ be the event that for every $j \in [t]$ and $i \in [k]$, there exists a center in $\tilde{C}_j$ (call it $\tilde{\mathbf{c}}_i^j$) such that $\left\|\tilde{\mathbf{c}}_i^j - \mathbf{c}_i\right\| \leq \gamma \cdot D_i$, where $D_i = \min_{j \neq i} \|\mathbf{c}_i - \mathbf{c}_j\|$.*

Namely, event E_C^γ implies that the output $\tilde{\mathcal{C}} \in ((\mathbb{R}^d)^k)^t$ of GenCenters is partitioned by Δ-far balls for $\Delta = 1/\gamma$, where Partition$(\tilde{\mathcal{C}})$ (according to Definition 3.6) is exactly $\{\{\tilde{\mathbf{c}}_1^j\}_{j=1}^t, \ldots, \{\tilde{\mathbf{c}}_k^j\}_{j=1}^t\}$ (i.e., for each $i \in [k]$, the centers $\{\tilde{\mathbf{c}}_i^j\}_{j=1}^t$ are very close to each other, compared to the distance from the other centers). Then in this section, we show how to construct an (ε, δ)-differentially private algorithm PrivatekMeans that invokes GenCenters with suitable choices for T and m, such that it achieves the following utility guarantee: For any k-centers C and a small enough γ, when the event E_C^γ occurs over GenCenters$(\mathcal{P}, k, s, t)$, then with probability $1 - \beta$, algorithm PrivatekMeans outputs $\hat{C} = \{\hat{\mathbf{c}}_1, \ldots, \hat{\mathbf{c}}_k\}$ with $\mathrm{COST}_{\mathcal{P}}(\hat{C}) \leq (1 + O(\gamma))\mathrm{COST}_{\mathcal{P}}(C)$ (plus some small additive error). Algorithm PrivatekMeans is described in Figure 6 and its properties are proven in Section 5.3. In Section 5.4 we show that a variant of the separation assumption in [ORSS12] implies that event $E_{C^*}^\gamma$ holds with high probability, where C^* are the optimal k means for $\mathcal{P}$.

In the following we define the parameter t used in Step 1 of Algorithm PrivatekMeans.

Procedure GenCenters

Input: A multiset $\mathcal{P}$ of points in $B(\mathbf{0}, \Lambda) \subseteq \mathbb{R}^d$, parameters $k, s, t \in \mathbb{N}$, and a (non-private) k-means algorithm $\mathcal{A}$.

1. For each $j \in [t]$:
   (a) Let $\mathcal{S}_j$ be a database containing s i.i.d. samples from $\mathcal{P}$ (with replacement).
   (b) Compute the k-tuple of centers $\tilde{C}_j = \mathcal{A}(\mathcal{S}_j, k)$.
2. Output $\mathcal{T} = \{\tilde{C}_1, \ldots, \tilde{C}_t\}$.

Figure 5: A procedure for generating t k-tuples of centers in $\mathbb{R}^d$.

Algorithm PrivatekMeans

Input: A multiset $\mathcal{P}$ of n points in $B(\mathbf{0}, \Lambda) \subseteq \mathbb{R}^d$, parameter $k \in \mathbb{N}$, privacy parameters $\varepsilon, \delta \in (0, 1]$, confidence parameter $\beta \in (0, 1]$, and a stability parameter $\gamma > 0$.
Additional input: A (non-private) k-means algorithm $\mathcal{A}$.

1. Let t be value from Definition 5.4 , and let $s = \lfloor \frac{n}{2t} \rfloor$.
2. Generate a t-size multiset of k-tuples $\mathcal{T} = \text{GenCenters}(\mathcal{P}, k, m, t; \mathcal{A})$.
3. Execute PrivatekAverages (Figure 3) over $\mathcal{T}$ with input parameters $\Lambda, \tilde{n} = t, \tilde{r}_{\min} = \frac{\gamma}{n}, \tilde{\varepsilon} = \frac{\varepsilon}{6}, \tilde{\delta} = \frac{\delta}{4e^{\varepsilon}}, \tilde{\beta} = \frac{\beta}{2}$. Let $\{\hat{\mathbf{a}}_1, \ldots, \hat{\mathbf{a}}_k\}$ be the outcome of the execution.
4. For each $i \in [k]$:
   - Let $\mathcal{P}_i$ be the points in $\mathcal{P}$ that $\hat{\mathbf{a}}_i$ is the closest point to them among $\{\hat{\mathbf{a}}_1, \ldots, \hat{\mathbf{a}}_k\}$.
   - Use the Gaussian Mechanism (2.19) with parameters $d, \Lambda, \hat{\varepsilon} = \frac{\varepsilon}{12}, \hat{\delta} = \frac{\delta}{8e^{\varepsilon}}, \hat{\beta} = \frac{\beta}{2k}$ to compute a noisy average $\hat{\mathbf{c}}_i$ of $\mathcal{P}_i$.
5. Output $\{\hat{\mathbf{c}}_1, \ldots, \hat{\mathbf{c}}_k\}$.

Figure 6: A private k-means approximation algorithm PrivatekMeans under stability assumption.

Definition 5.4. *Let $t = t(n, d, k, \beta, \gamma, \varepsilon, \delta)$ be the smallest integer that satisfies*

$$t \geq \frac{6\lambda dk\ell\sqrt{\log\left(\frac{12k\ell}{\delta}\right)}}{\varepsilon}\left(\sqrt{\log\left(\frac{12dk\ell}{\delta}\right)\log\left(\frac{2dk\ell}{\beta}\right)} + \log\left(\frac{12\Lambda dkn}{\delta\gamma}\right)\right)$$

where $\ell = \ell(\tilde{n} = t, d, k, \frac{\varepsilon}{12}, \frac{\delta}{16e^{\varepsilon}}, \frac{\beta}{4})$ is the value from Definition 4.8, and λ is the constant from Theorem 4.12.

We note that when $\beta = O(1)$ and $\delta \geq \frac{1}{\text{poly}(n)}$, it holds that $\ell = O\left(\frac{\log^2 n}{\varepsilon \log t}\right) \leq O\left(\frac{1}{\varepsilon}\log^2 n\right)$. This yields that $t = \tilde{O}\left(\frac{dk\log^{2.5}(n)\cdot\log\left(\frac{\Lambda n}{\gamma}\right)}{\varepsilon^2}\right)$ in this case. This means that for large enough n, we obtain that $t = \text{polylog}(n)$.

5.3 Properties of PrivatekMeans

The following theorem captures the privacy guarantee of PrivatekMeans.

Theorem 5.5 (Privacy of Algorithm PrivatekMeans)**.** *For every $d, k > 0$, every $\beta, \varepsilon, \delta, \gamma \in (0, 1]$ and every algorithm $\mathcal{A}$, Algorithm* PrivatekMeans$(\cdot, k, \alpha, \beta, \gamma; \mathcal{A})$ *is (ε, δ)-differentially private for databases $\mathcal{P}$ over $B(\mathbf{0}, \Lambda) \subset \mathbb{R}^d$.*

Proof. The proof builds on the fact that switching between sampling with replacement and without replacement has only a small effect on the privacy, as stated in Lemma 2.25.

Consider a different variant $\widetilde{\text{GenCenters}}$ of the procedure GenCenters, in which the sampling of the $\approx n/2$ points in all the iterations of Step 1a ($s = \lfloor n/(2t) \rfloor$ points in T iterations) is done without replacement, and consider a variant $\widetilde{\text{PrivatekMeans}}$ of PrivatekMeans in which it executes $\widetilde{\text{GenCenters}}$ in Step 2 rather than GenCenters. Let $\mathcal{P} = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ and $\mathcal{P}' = \{\mathbf{x}'_1, \ldots, \mathbf{x}'_n\}$ be two neighboring databases of points. In the following we consider two independent executions $\widetilde{\text{PrivatekMeans}}(\mathcal{P})$ and $\widetilde{\text{PrivatekMeans}}(\mathcal{P}')$ (both with the same parameters $k, \varepsilon, \delta, \beta, \mathcal{A}$). For $j \in [t]$ let $\mathcal{J}_j \subseteq [n]$ be the s chosen indices of the points $\mathcal{S}_j$ in Step 1a of $\widetilde{\text{GenCenters}}$ (i.e., $\mathcal{S}_j = \{\mathbf{x}_i\}_{i \in \mathcal{J}_j}$), and let $\mathcal{J}'_j$ be the same indices in the execution $\widetilde{\text{PrivatekMeans}}(\mathcal{P}')$. Since $\mathcal{J}_j$ and $\mathcal{J}'_j$ only depend on n and not on the content of $\mathcal{P}$ and $\mathcal{P}'$, it is enough to prove that the output of both executions is (ε, δ)-indistinguishable conditioned on the event that $\mathcal{J}_j = \mathcal{J}'_j$ for every $j \in [t]$. In the following, we assume that this event occurs.

Since $\mathcal{P}$ and $\mathcal{P}'$ are neighboring, there exists at most one index $j \in [t]$ such that $\mathcal{S}_j$ of the execution $\widetilde{\text{PrivatekMeans}}(\mathcal{P})$ is different than the corresponding set in $\widetilde{\text{PrivatekMeans}}(\mathcal{P}')$, and therefore, the outputs $\hat{\mathcal{C}}$ of $\widetilde{\text{GenCenters}}$ are different by at most one k-tuple. Therefore, by Theorem 4.13, we deduce that the outcome of Step 3 is $(\tilde{\varepsilon}, \tilde{\delta}) = (\frac{\varepsilon}{6}, \frac{\delta}{4e^{\varepsilon}})$-differentially private.

In the following, we prove that for any fixing of k averages $\tilde{\mathbf{a}}_1, \ldots, \tilde{\mathbf{a}}_k$, Step 4 is $(\frac{\varepsilon}{6}, \frac{\delta}{4e^{\varepsilon}})$-differentially private. Given that, we deduce that $\widetilde{\text{PrivatekMeans}}$ is $(\frac{\varepsilon}{3}, \frac{\delta}{2e^{\varepsilon}})$-differentially private by (adaptive) composition of Steps 3 and 4 (Theorem 2.10). Hence, we conclude that the original algorithm PrivatekMeans, that chooses the points with replacement, is (ε, δ)-differentially private by applying Lemma 2.25 with $m = n/2, \frac{\varepsilon}{3}, \frac{\delta}{2e^{\varepsilon}}$.

It is left to prove the privacy guarantee of Step 4. For that, fix k averages $\hat{A} = \{\tilde{\mathbf{a}}_1, \ldots, \tilde{\mathbf{a}}_k\}$, let $\mathcal{P}_1, \ldots, \mathcal{P}_k$ be the k multisets in Step 4 w.r.t $\mathcal{P}$ and $\hat{A}$, and let $\mathcal{P}'_1, \ldots, \mathcal{P}'_k$ be the same multisets

w.r.t $\mathcal{P}'$ and $\hat{A}$. Since $\mathcal{P}$ and $\mathcal{P}'$ are neighboring, there exist at most two indices $i \in [k]$ such that $\mathcal{P}_i \neq \mathcal{P}'_i$, and for each one of them, $\mathcal{P}_i$ and $\mathcal{P}'_i$ are neighboring. Therefore, by the privacy guarantee of the Gaussian mechanism along with basic composition (Theorem 2.10), Step 4 is $(2 \cdot \frac{\varepsilon}{12}, 2 \cdot \frac{\delta}{8e^{\varepsilon}})$-differentially private, as required.

□

The following theorem, which captures the utility guarantee of PrivatekMeans, states that when event E_C^{γ} (Definition 5.3) occurs by GenBalancedSamples in Step 2, then with probability at least $1-\beta$, the output $\hat{C} = \{\hat{\mathbf{c}}_1, \ldots, \hat{\mathbf{c}}_k\}$ has $\mathrm{COST}_{\mathcal{P}}(\hat{C}) \leq (1+O(\gamma))\mathrm{COST}_{\mathcal{P}}(C)+O\left(\frac{\Lambda^2 dk \log(1/\delta)\log(k/\beta)}{\varepsilon^2}\right)$.

Theorem 5.6 (Utility of Algorithm PrivatekMeans)**.** *Let $d, k > 0$, let $\beta, \varepsilon, \delta \in (0,1]$, let $\mathcal{P}$ be a multiset of n points in $B(\mathbf{0}, \Lambda) \subseteq \mathbb{R}^d$ and let $\mathcal{A}$ be an algorithm. In addition, let $C = \{\mathbf{c}_1, \ldots, \mathbf{c}_k\} \in (\mathbb{R}^d)^k$ with $\min_{i \neq j}\|\mathbf{c}_i - \mathbf{c}_j\| \geq 1/n$ and let $\gamma \in (0, \frac{1}{32}]$. Consider a random execution of* PrivatekMeans$(\mathcal{P}, k, \varepsilon, \delta, \beta, \gamma; \mathcal{A})$ *conditioned that the event E_C^{γ} occurs by* GenBalancedSamples *in Step 2 of the execution. Then with probability $1-\beta$ (over the above conditioned execution), the output $\hat{C} = \{\hat{\mathbf{c}}_1, \ldots, \hat{\mathbf{c}}_k\}$ of* PrivatekMeans *satisfies that*

$$COST_{\mathcal{P}}(\{\hat{\mathbf{c}}_1, \ldots, \hat{\mathbf{c}}_k\}) \leq (1+64\gamma)\mathrm{COST}_{\mathcal{P}}(C) + O\left(\frac{\Lambda^2 k \log(1/\delta) \cdot (d + \log(k/\beta))}{\varepsilon^2}\right).$$

Proof. Consider a random execution of PrivatekMeans$(\mathcal{P}, k, \varepsilon, \delta, \beta, \gamma; \mathcal{A})$ conditioned on the event E_C^{γ}. For $j \in [t]$, let $\tilde{C}_j = \{\tilde{\mathbf{c}}_1^j, \ldots, \tilde{\mathbf{c}}_k^j\}$ be the value from Step 1b of the j'th iteration of GenCenters, where we denote by $\tilde{\mathbf{c}}_i^j$ the center that is close to $\mathbf{c}_i$, i.e., $\left\|\hat{\mathbf{c}}_i^j - \mathbf{c}_i\right\| \leq \gamma \cdot D_i$, where $D_i = \min_{j \neq i}\|\mathbf{c}_i - \mathbf{c}_j\|$ (such center exists by event E_C^{γ}). In addition, for $i \in [k]$, let $\mathbf{a}_i = \mathrm{Avg}(\{\hat{\mathbf{c}}_i^j\}_{j=1}^t)$ and note that

$$\begin{aligned} \forall i \in [k] : \|\mathbf{a}_i - \mathbf{c}_i\| &= \left\|\frac{1}{t}\sum_{j=1}^{t}\hat{\mathbf{c}}_i^j - \mathbf{c}_i\right\| \\ &\leq \frac{1}{t}\sum_{j=1}^{t}\left\|\hat{\mathbf{c}}_i^j - \mathbf{c}_i\right\| \\ &\leq \gamma \cdot D_i. \end{aligned} \tag{6}$$

Now, let $\tilde{\mathcal{C}}$ be the output of GenCenters in Step 2 of Algorithm PrivatekMeans, and note that $\tilde{\mathcal{C}}$ is evenly-partitioned by the set of balls $\{B(\mathbf{c}_i, r_i = 2\gamma D_i)\}_{i=1}^k$ which are also Δ-far balls for $\Delta = 16$ since $\gamma < \frac{1}{32}$, and $\mathrm{Partition}(\tilde{\mathcal{C}}) = \{\{\hat{\mathbf{c}}_1^j\}_{j=1}^t, \ldots, \{\hat{\mathbf{c}}_k^j\}_{j=1}^t\}$. Therefore, when executing Algorithm PrivatekAverages in Step 3, we obtain by Theorem 4.12 a set of k points $\{\hat{\mathbf{a}}_1, \ldots, \hat{\mathbf{a}}_k\}$ such that

with probability $1-\tilde{\delta} \geq 1-\frac{\beta}{2}$ it holds that

$$\begin{aligned}
\forall i \in [k]: \ \|\mathbf{a}_i - \hat{\mathbf{a}}_i\| & \qquad (7)\\
&\leq \max\{r_i, \tilde{r}_{\min}\} \cdot \frac{\lambda dk\ell\sqrt{\log\left(\frac{k\ell}{\tilde{\delta}}\right)}}{\tilde{\varepsilon}\tilde{n}}\left(\sqrt{\log\left(\frac{dk\ell}{\tilde{\delta}}\right)\log\left(\frac{dk\ell}{\tilde{\beta}}\right)} + \log\left(\frac{\Lambda dk}{\tilde{r}_{\min}\tilde{\delta}}\right)\right)\\
&= \gamma D_i \cdot \frac{6\lambda dk\ell\sqrt{\log\left(\frac{12k\ell}{\delta}\right)}}{\varepsilon t}\left(\sqrt{\log\left(\frac{12dk\ell}{\delta}\right)\log\left(\frac{2dk\ell}{\beta}\right)} + \log\left(\frac{12\Lambda dkn}{\delta\gamma}\right)\right)\\
&\leq \gamma D_i,
\end{aligned}$$

where $\ell = \ell(\tilde{n}, d, k, \tilde{\varepsilon}/2, \tilde{\delta}/4, \tilde{\beta}/2)$ is the value from Definition 4.8, and λ is the constant from Theorem 4.12. In the second inequality we used the fact that $r_i \leq \gamma D_i$ and that $\tilde{r}_{\min} = \frac{\gamma}{n} \leq \gamma D_i$, and the last inequality holds by the definition of t (Definition 5.4). Therefore, we deduce by Equations (6) and (7) that with probability $1-\frac{\beta}{2}$ it holds that

$$\forall i \in [k] : \|\hat{\mathbf{a}}_i - \mathbf{c}_i\| \leq 2\gamma \cdot D_i. \qquad (8)$$

Let $\mathcal{P}_1, \ldots, \mathcal{P}_k$ be the clusters from Step 4 of the algorithm. If Equation (8) occurs, then by Proposition 5.2 we get that

$$\sum_{i=1}^{k}\sum_{\mathbf{x}\in\mathcal{P}_i}\|\mathbf{x} - \mathrm{Avg}(\mathcal{P}_i)\|^2 \leq (1+64\gamma)\mathrm{COST}_{\mathcal{P}}(C). \qquad (9)$$

Since the algorithm computes a noisy estimation $\hat{\mathbf{c}}_i$ of each $\mathrm{Avg}(\mathcal{P}_i)$, we get by the properties of the Gaussian mechanism (see Observation 2.20) and the union bound that with probability $1-k\hat{\beta} = 1-\frac{\beta}{2}$ it holds that

$$\forall i \in [k]: \quad \|\hat{\mathbf{c}}_i - \mathrm{Avg}(\mathcal{P}_i)\| \leq O\left(\frac{\Lambda\sqrt{\log(1/\delta)}}{\varepsilon|\mathcal{P}_i|}\left(\sqrt{d} + \sqrt{\log(k/\beta)}\right)\right) \qquad (10)$$

Finally, since Equation (9) occurs with probability $1-\frac{\beta}{2}$, and Equation (10) occurs with probability $1-\frac{\beta}{2}$, we conculde that with probability $1-\beta$ both of them occurs, which implies that

$$\begin{aligned}
&\mathrm{COST}_{\mathcal{P}}(\{\hat{\mathbf{c}}_1, \ldots, \hat{\mathbf{c}}_k\})\\
&\leq \sum_{i=1}^{k}\sum_{\mathbf{x}\in\mathcal{P}_i}\|\mathbf{x} - \hat{\mathbf{c}}_i\|^2\\
&= \sum_{i=1}^{k}\sum_{\mathbf{x}\in\mathcal{P}_i}\left(\|\mathbf{x} - \mathrm{Avg}(\mathcal{P}_i)\|^2 + \|\hat{\mathbf{c}}_i - \mathrm{Avg}(\mathcal{P}_i)\|^2 + 2\|\mathbf{x} - \mathrm{Avg}(\mathcal{P}_i)\| \cdot \|\hat{\mathbf{c}}_i - \mathrm{Avg}(\mathcal{P}_i)\|\right)\\
&\leq (1+64\gamma)\mathrm{COST}_{\mathcal{P}}(C) + O\left(\frac{\Lambda^2 k\log(1/\delta)}{\varepsilon^2}(d+\log(k/\beta))\right) + 2\Lambda \cdot O\left(\frac{\Lambda\sqrt{\log(1/\delta)}}{\varepsilon}\left(\sqrt{d}+\sqrt{\log(k/\beta)}\right)\right)\\
&= (1+64\gamma)\mathrm{COST}_{\mathcal{P}}(C) + O\left(\frac{\Lambda^2 k\log(1/\delta)}{\varepsilon^2}(d+\log(k/\beta))\right),
\end{aligned}$$

where in the last term of the second inequality we used the fact that $\|\mathbf{x} - \mathrm{Avg}(\mathcal{P}_i)\| \leq \Lambda$ for all $i \in [k]$ and $\mathbf{x} \in \mathcal{P}_i$. □

Remark 5.7 (Run time of PrivatekMeans)**.** *The algorithm calls the non-private algorithm $\mathcal{A}$ t-times (where $t \leq \text{polylog}(n)$), each time over a collection of points of size $s = O(n/t)$. Then, the most expensive step is executing* PrivatekAverages*, which takes $\tilde{O}(dk^2 n)$ time.*

5.4 Private k-Means under Separation Assumption

In this section we show that our stability assumption holds with high probability when the multiset $\mathcal{P}$ is separated according to Ostrovsky *et al.* [ORSS12]. Formally, a multiset of points $\mathcal{P}$ is *ϕ-separated* for k-means if $\text{OPT}_k(\mathcal{P}) \leq \phi^2 \text{OPT}_{k-1}(\mathcal{P})$. In Definition 5.8 we strength this definition of [ORSS12] to include also an additive separating term ξ.

Definition 5.8 ((ϕ,ξ)-separated)**.** *A multiset $\mathcal{P} \in (\mathbb{R}^d)^*$ is (ϕ,ξ)-separated for k-means if $\text{OPT}_k(\mathcal{P}) + \xi \leq \phi^2 \cdot \text{OPT}_{k-1}(\mathcal{P})$. Note that $\mathcal{P}$ is ϕ-separated iff it is $(\phi,0)$-separated.*

We use the following theorem from [ORSS12] which states that when $\mathcal{P}$ is ϕ-separated for k-means for sufficiently small ϕ, then any set of k centers that well approximate the k means cost, must have the property that each of its centers is relatively close to an optimal center.

Theorem 5.9 ([ORSS12])**.** [3] *Let ν and ϕ be such that $\frac{\nu+\phi^2}{1-\phi^2} < \frac{1}{16}$. Suppose that $\mathcal{P} \in (\mathbb{R}^d)^*$ is ϕ-separated for k-means. Let $C^* = \{\mathbf{c}_1^*, \ldots, \mathbf{c}_k^*\}$ be a set of optimal centers for $\mathcal{P}$, and let $C = \{\mathbf{c}_1, \ldots, \mathbf{c}_k\}$ be centers such that $\text{COST}_{\mathcal{P}}(C) \leq \nu \cdot \text{OPT}_{k-1}(\mathcal{P})$. Then for each $\mathbf{c}_i$ there is a distinct optimal center, call it $\mathbf{c}_i^*$, such that $\|\mathbf{c}_i - \mathbf{c}_i^*\| \leq 2 \cdot \frac{\nu+\phi}{1-\phi} \cdot D_i$, where $D_i = \min_{j \neq i} \left\| \mathbf{c}_i^* - \mathbf{c}_j^* \right\|$.*

The following lemma states that for suitable choices of ϕ and λ, if $\mathcal{P}$ is (ϕ,λ)-separated for k-means, then with high probability, the event $E^{\gamma}_{C^*}$ over a random execution of PrivatekMeans (Definition 5.3) occurs, where C^* is the optimal k-means for $\mathcal{P}$

Lemma 5.10 (Bounding the stability probability)**.** *Let $\varepsilon, \delta, \beta, \phi \in (0,1)$, $\gamma \in [0, \frac{1}{16}]$, , $d, k, n \in \mathbb{N}$, let $t = t(n, d, k, \beta, \gamma, \varepsilon, \delta)$ be the value from Definition 5.4, let $\mathcal{A}$ be a (non-private) ω-approximation algorithm for k-means, let $\mathcal{P} \in (B(\mathbf{0}, \Lambda))^n$ and let $C^* = \{\mathbf{c}_1^*, \ldots, \mathbf{c}_k^*\} \in (\mathbb{R}^d)^k$ be the optimal k-means for $\mathcal{P}$. If the following holds:*

- *$\mathcal{P}$ is (ϕ,ξ)-separated for k-means, where*

$$\xi = \xi\left(\left\lfloor \frac{n}{2t} \right\rfloor, \frac{\beta}{t}\right) = \tilde{O}\left(\Lambda^2 tkd \log(nt/\beta) + \Lambda\sqrt{tkd \log(n/\beta) \cdot \omega \text{OPT}_k(\mathcal{P})}\right)$$

  is the function from Proposition 5.1, and

- $\frac{(1+\omega)\phi^2}{1-\phi^2} < \frac{1}{16}$*, and*

- $\gamma \geq 2 \cdot \frac{\omega\phi^2+\phi}{1-\phi}$*,*

then when executing PrivatekMeans *on inputs $\mathcal{P}, k, \varepsilon, \delta, \beta, \gamma, \mathcal{A}$, the event $E^{\gamma}_{C^*}$ (Definition 5.3) occurs with probability at least $1 - \beta$.*

[3] The statement of this theorem was taken from [SSS20].

Proof. For $j \in [t]$, let $\mathcal{S}_j$ and $\tilde{C}_j$ be the k-tuple in the execution of PrivatekMeans in steps 1a and 1b of GenCenters (respectively). Note that by Proposition 5.1 and the union bound, with probability at least $1-\beta$ it holds that

$$\forall j \in [t]: \quad \mathrm{COST}_{\mathcal{P}}(\tilde{C}_j) \leq \omega \cdot \mathrm{OPT}_k(\mathcal{P}) + \lambda \leq \omega\phi^2 \mathrm{OPT}_{k-1}(\mathcal{P}), \tag{11}$$

where the last inequality holds by the assumption that $\mathcal{P}$ is (ϕ, λ)-separated for k-means and that $\omega \geq 1$. In the following, assume that (11) occurs. Since $\mathcal{P}$ is (in particular) ϕ-separated, and since the conditions of Theorem 5.9 hold with $\nu = \omega\phi^2$, we obtain from Theorem 5.9 that for every $i \in [k]$ and $j \in [t]$, there exists $\tilde{\mathbf{c}}_i^j \in \tilde{C}_j$ such that $\left\|\mathbf{c}_i^* - \tilde{\mathbf{c}}_i^j\right\| \leq \gamma D_i$, meaning that event $E_{C^*}^{\gamma}$ occurs, as required. □

As a corollary of Theorem 5.6 and Lemma 5.10, we obtain our main application of algorithm PrivatekMeans.

Corollary 5.11. *Let $\varepsilon, \delta, \beta \in (0,1)$ and let $\phi, \mathcal{P}, \mathcal{A}, \omega, \gamma$ as in Lemma 5.10. Then when executing* PrivatekMeans *on inputs $\mathcal{P}, k, \varepsilon, \delta, \beta, \mathcal{A}$, with probability $1-2\beta$, the resulting centers $\hat{C} = \{\hat{\mathbf{c}}_1, \ldots, \hat{\mathbf{c}}_k\}$ satisfy*

$$\mathrm{COST}_{\mathcal{P}}(\hat{C}) \leq (1+64\gamma)\mathrm{OPT}_k(\mathcal{P}) + O\left(\frac{\Lambda^2 k \log(1/\delta)}{\varepsilon^2}(d + \log(k/\beta))\right)$$

Proof. The proof almost immediately holds by Theorem 5.6 and Lemma 5.10 when applying them to the optimal k-means of $\mathcal{P}$, which we denote by $C^* = \{\mathbf{c}_1^*, \ldots, \mathbf{c}_k^*\}$. The only missing requirement is to show that $D^* := \min_{i \neq j}\left\|\mathbf{c}_i^* - \mathbf{c}_j^*\right\| \geq 1/n$, as required by Theorem 5.6. For proving this, note that on the one hand it holds that $\mathrm{OPT}_{k-1}(\mathcal{P}) \leq D^* n + \mathrm{OPT}_k(\mathcal{P})$, and on the other hand, since we assume that $\mathcal{P}$ is (ϕ, λ)-separated for $\phi \leq 1$ and $\lambda \geq 1$ then it holds that $\mathrm{OPT}_k(\mathcal{P}) + 1 \leq \mathrm{OPT}_{k-1}(\mathcal{P})$. From the two inequalities we conclude that $D^* \geq 1/n$ and the corollary follows. □

6 Mixture of Gaussians

In this section we present our second application of k-tuple clustering, which is an (ε, δ)-differentially private algorithm PrivatekGaussians for learning a mixture of well separated and bounded k Gaussians. We first start with relevant preliminaries for this section.

6.1 Preliminaries

The total variation distance between two distributions P and Q over a universe $\mathcal{U}$ is defined by $\mathrm{d_{TV}}(P,Q) = \sup_{\mathcal{S} \subseteq \mathcal{U}} |P(\mathcal{S}) - Q(\mathcal{S})|$. Given a matrix $A = (a_{i,j})_{i,j \in [d]} \in \mathbb{R}^{d \times d}$, we let $\|A\| = \sup_{\|\mathbf{x}\|=1} \|A\mathbf{x}\|$ be its ℓ_2 norm.

6.1.1 Gaussians

Let $\mathcal{N}(0,1)$ be the standard Gaussian distribution over $\mathbb{R}$ with probability density function $p(z) = \frac{1}{\sqrt{2\pi}} e^{-\frac{z^2}{2}}$. In $\mathbb{R}^d$, let $\mathcal{N}(\mathbf{0}, \mathbb{I}_{d \times d})$ be the standard multivariate Gaussian distribution. That is, if $\mathbf{Z} \sim \mathcal{N}(\mathbf{0}, \mathbb{I}_{d \times d})$ then $\mathbf{Z} = (Z_1, \ldots, Z_d)$ where $Z_1, \ldots, Z_d$ are i.i.d. according to $N(0,1)$. Other

Gaussian distributions over $\mathbb{R}^d$ arise by applying (invertible) linear maps on $\mathcal{N}(\mathbf{0}, \mathbb{I}_{d\times d})$. That is, the distribution $\mathbf{X} \sim \mathcal{N}(\mu, \Sigma = AA^T)$ for $\mu \in \mathbb{R}^d$ and (invertible) $A \in \mathbb{R}^{d\times d}$ is defined by $\mathbf{X} = A\mathbf{Z} + \mu$, where $\mathbf{Z} \sim N(\mathbf{0}, \mathbb{I}_{d\times d})$, and it holds that $\mathrm{E}[\mathbf{X}] = \mu$ and $\mathrm{Cov}(\mathbf{X}) = (\mathrm{Cov}(X_i, X_j))_{i,j}$ (the covariance matrix) equals to Σ. The contours of equal density are ellipsoids around μ: $\{\mathbf{x} \in \mathbb{R}^d \colon (\mathbf{x}-\mu)^T\Sigma^{-1}(\mathbf{x}-\mu) = r^2\}$. We let $\mathcal{G}(d)$ be the family of all d-dimensional Gaussian — that is, the set of all distribution $\mathcal{N}(\mu, \Sigma)$ where $\mu \in \mathbb{R}^d$ and Σ is a $d \times d$ positive semidefinite (PSD) matrix.

Definition 6.1 (Bounded Gaussian)**.** *For $R, \sigma_{\max}, \sigma_{\min} > 0$, a Gaussian $\mathbf{G} = \mathcal{N}(\mu, \Sigma) \in \mathcal{G}(d)$ is $(R, \sigma_{\max}, \sigma_{\min})$-bounded if $\|\mu\| \leq R$ and $\sigma_{\min}^2 \leq \|\Sigma\|_2 \leq \sigma_{\max}^2$.*

We next define the properties of a general algorithm that privately learns a the parameters of a (single) bounded Gaussian.

Definition 6.2 (Private Algorithm for Learning a Bounded Gaussian)**.** *Let $\mathcal{A}$ be an algorithm that gets as input a database $\mathcal{P} \in (\mathbb{R}^d)^*$ and parameters $d, \varepsilon, \delta, \alpha, \beta, R, \sigma_{\max}, \sigma_{\min}$, and outputs $(\hat{\mu}, \hat{\Sigma})$. Let $s = s(d, \varepsilon, \delta, \alpha, \beta, R, \sigma_{\max}, \sigma_{\min})$ be a function. We say that $\mathcal{A}$ is a **private algorithm for learning a bounded Gaussian with sample complexity** υ if given the above parameters, $\mathcal{A}$ is an (ε, δ)-differentially private algorithm that satisfy the following utility guarantee: If $\mathcal{N}(\mu, \Sigma)$ is a $(R, \sigma_{\max}, \sigma_{\min})$-bounded Gaussian, and $\mathcal{P}$ consists of at least υ i.i.d. samples from $\mathcal{N}(\mu, \Sigma)$, then with probability at least $1-\beta$ it holds that $\mathrm{d_{TV}}(\mathcal{N}(\mu, \Sigma), \mathcal{N}(\hat{\mu}, \hat{\Sigma})) \leq \alpha$.*

The best known examples for such algorithms are the constructions of [KLSU19] and [BDKU20], which have sample complexity $\upsilon = \tilde{O}\left(\left(\frac{d^2}{\alpha^2} + \frac{d^2}{\varepsilon\alpha} + \frac{d^{3/2}\sqrt{\log\left(\frac{\sigma_{\max}}{\sigma_{\min}}\right)} + \sqrt{d\log R}}{\varepsilon}\right) \cdot \log(1/\beta)\right)$. We remark that without privacy, the required sample complexity is $\Theta\left(\frac{d^2\log(1/\beta)}{\alpha^2}\right)$, which means that privacy comes almost for free unless $\frac{1}{\varepsilon}$, $\frac{\sigma_{\max}}{\sigma_{\min}}$ or R are quite large.

6.1.2 Gaussian Mixtures

The class of Gaussian k-mixtures in $\mathbb{R}^d$ is

$$\mathcal{G}(d,k) := \{\sum_{i=1}^{k} w_i \mathbf{G}_i \colon \mathbf{G}_1, \ldots, \mathbf{G}_k \in \mathcal{G}(d), w_1, \ldots, w_k > 0, \sum_{i=1}^{k} w_i = 1\}$$

A Gaussian mixture can be specified by a set of k tuples: $\{(\mu_1, \Sigma_1, w_1), \ldots, (\mu_k, \Sigma_k, w_k)\}$, where each tuple represents the mean, covariance matrix, and mixing weight of one of its components.

Definition 6.3 (Bounded Mixture of Gaussians)**.** *For $R, \sigma_{\max}, \sigma_{\min}, w_{\min} > 0$, a Gaussian mixture $\mathcal{D} = \{(\mu_1, \Sigma_1, w_1), \ldots, (\mu_k, \Sigma_k, w_k)\} \in \mathcal{G}(d,k)$ is $(R, \sigma_{\max}, \sigma_{\min}, w_{\min})$-bounded if for all $i \in [k]$, the Gaussian $\mathcal{N}(\mu_i, \Sigma_i)$ is $(R, \sigma_{\max}, \sigma_{\min})$-bounded and $w_i \geq w_{\min}$.*

Definition 6.4 (Separated Mixture of Gaussians)**.** *Let $\mathcal{D} = \{(\mu_1, \Sigma_1, w_1), \ldots, (\mu_k, \Sigma_k, w_k)\}$ be a mixture of k Gauusians over $\mathbb{R}^d$, for $i \in [k]$ let $\sigma_i^2 = \|\Sigma_i\|_2$, and let $h > 0$. We say that $\mathcal{D}$ is h-separated if*

$$\forall 1 \leq i < j \leq k : \ \|\mu_i - \mu_j\| \geq h \cdot \max\{\sigma_i, \sigma_j\}.$$

We next define a labeling algorithm for a mixture $\mathcal{D}$.

Definition 6.5 (Labeling Algorithm for a Mixture of Gaussians)**.** *Let $n, k \in \mathbb{N}$, $\beta \in (0,1)$ and let $\mathcal{D} = \{(\mu_1, \Sigma_1, w_1), \ldots, (\mu_k, \Sigma_k, w_k)\}$ be a mixture of k Gaussians. We say that an Algorithm $\mathcal{A}$ is an (n, β)-**labeling algorithm for the mixture** $\mathcal{D}$ if with probability $1 - \beta$, when sampling a database $\mathcal{P}$ of n i.i.d. samples from $\mathcal{D}$, Algorithm $\mathcal{A}$ on inputs $\mathcal{P}, d, k, \beta$, outputs a labeling function $L \colon \mathcal{P} \to [k]$ such that for all $\mathbf{x}, \mathbf{x}' \in \mathcal{P}$: $L(\mathbf{x}) = L(\mathbf{x}') \iff \mathbf{x}$ and $\mathbf{x}'$ were drawn from the same Gaussian.*

There are various examples of non-private algorithms that learns the parameters of mixtures of Gaussian under different separations assumptions, and most of them can be easily converted into a labeling algorithm. For instance, [DS00; SK01] showed how to learn mixtures with separation that is only proportional to $d^{1/4}$. Moreover, there is a wide line of works that show how to handle mixtures with separation that is independent of d: Separation that is proportional to $\sqrt{k}$ [AM05], $k^{1/4}$ [VW04], k^{ε} [HL18a; KSS18; DKS18], or even $\sqrt{\log k}$ [RV17]. In Section 6.2 we show that our algorithm can transform each such non-private algorithm into a private one, as long as we are given n points from a mixture that is at least $\tilde{\Omega}(\log n)$−separated.

6.1.3 Concentration Bounds

Fact 6.6 (One-dimensional Gaussian)**.** *Let $\mathbf{X} \sim \mathcal{N}(0, \sigma^2)$. Then for any $\beta > 0$ it holds that*

$$\Pr\left[\mathbf{X} \geq \sigma\sqrt{2\log(1/\beta)}\right] \leq \beta.$$

Fact 6.7 (follows by the Hanson-Wright inequality [HW71])**.** *If $\mathbf{X} \sim \mathcal{N}(\mu, \Sigma)$ then with probability at least $1 - \beta$ it holds that*

$$\|\mathbf{X} - \mu\| \leq \left(\sqrt{d} + \sqrt{2\log(1/\beta)}\right) \cdot \sqrt{\|\Sigma\|}.$$

The following fact is an immediate corollary of Fact 6.7.

Fact 6.8. *Let $X_1, \ldots, X_m$ be i.i.d. random variables distributed according to a d-dimensional Gaussian $\mathcal{N}(\mu, \Sigma)$, and let $\sigma^2 = \|\Sigma\|$. Then with $1 - \beta$ it holds that*

$$\|\mathrm{Avg}(X_1, \ldots, X_m) - \mu\| \leq \frac{\sqrt{d} + \sqrt{2\log(1/\beta)}}{\sqrt{m}} \cdot \sigma,$$

Proof. Follows by Fact 6.7 since $\mathrm{Avg}(X_1, \ldots, X_m)$ is distributed according to $\mathcal{N}(\mu, \frac{1}{m} \cdot \Sigma)$. □

6.2 Algorithm PrivatekGaussians

In this section we describe our algorithm PrivatekGaussians that privately learns a mixture of separated and bounded k Gaussians $\mathcal{D} = \{(\mu_1, \Sigma_1, w_1), \ldots, (\mu_k, \Sigma_k, w_k)\}$. The formal description of the algorithm appear at Figure 8.

In the following we define the parameter t used in Step 2 of Algorithm PrivatekGaussians.

Algorithm CollectEmpiricalMeans

Input: A database $\mathcal{P}' = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ and parameters $k, s, t \in \mathbb{N}$, where $n \geq st$.
Additional inputs: a (non-private) labeling algorithm $\mathcal{A}$ for a mixture of Gaussians.

1. For each $j \in [t]$:
   (a) Let $\mathcal{S}_j = \{\mathbf{x}_{(j-1)s+1}, \ldots, \mathbf{x}_{js}\}$.
   (b) Execute $\mathcal{A}$ on inputs $\tilde{\mathcal{P}} = \mathcal{S}_j, \tilde{k} = k$, and let $L_j \colon \mathcal{S}_j \to [k]$ be the resulting labeling function.
   (c) For each $i \in [k]$:
      - Compute $\bar{\mu}_{j,i} = \mathrm{Avg}(\{\mathbf{x} \in \mathcal{S}_j \colon L_j(\mathbf{x}) = i\})$.
   (d) Set $M_j = \{\bar{\mu}_{j,1}, \ldots, \bar{\mu}_{j,k}\} \in (\mathbb{R}^d)^k$.
2. Output $\mathcal{T} = \{M_1, \ldots, M_t\} \in ((\mathbb{R}^d)^k)^t$.

Figure 7: A procedure for generating T balanced k-tuples from a mixture of k Gaussians.

Definition 6.9. *Let $t = t(n, d, k, \beta, \gamma, \varepsilon, \delta, R, \sigma_{\max}, \sigma_{\min})$ be the smallest integer that satisfies*

$$t \geq \frac{\lambda dk\ell\sqrt{\log\left(\frac{k\ell}{\delta}\right)}}{\varepsilon}\left(\sqrt{\log\left(\frac{dk\ell}{\delta}\right)\log\left(\frac{4dk\ell}{\beta}\right)} + \log\left(\frac{dk(16R + \gamma h\sigma_{\max})}{\gamma\delta h\sigma_{\min}}\right)\right)$$

where $h = 2\sqrt{2\log\left(\frac{8n}{\beta}\right)}$, $\ell = \ell(t, d, k, \varepsilon/2, \delta/4, \beta/8)$ is the value from Definition 4.8, and λ is the constant from Theorem 4.12.

Assuming that $\beta, \gamma = O(1)$, we obtain that $\ell = \frac{\log^2(1/\delta)}{\varepsilon \log t} \leq O(\frac{1}{\varepsilon}\log^2(1/\delta))$, which yields that $t = \tilde{O}\left(\frac{dk\log^{2.5}(1/\delta)\left(\sqrt{\log(1/\delta)} + \log\left(\frac{R\sigma_{\max}}{\sigma_{\min}}\right)\right)}{\varepsilon^2}\right)$.

6.2.1 Properties of PrivatekGaussians

The following theorem summarizes the privacy guarantee of PrivatekGaussians.

Theorem 6.10 (Privacy of Algorithm PrivatekGaussians)**.** *Let $\mathcal{A}'$ be a private algorithm for learning a (single) bounded Gaussian according to Definition 6.2. Then for every $d, k, R, \sigma_{\max}, \sigma_{\min}, w_{\min} > 0$, every $\alpha, \beta, \varepsilon, \delta, \gamma \in (0, 1)$ and every algorithm $\mathcal{A}$, Algorithm* PrivatekGaussians$(\cdot, k, \alpha, \beta, \varepsilon, \delta, R, \sigma_{\max}, \sigma_{\min}; \mathcal{A}, \mathcal{A}')$ *is (ε, δ)-differentially private for databases $\mathcal{P} \in \left(\mathbb{R}^d\right)^*$.*

Proof. Assume for simplicity (and without loss of generality) that the input algorithm $\mathcal{A}$ is deterministic, let $\mathcal{P}, \tilde{\mathcal{P}} \in \left(\mathbb{R}^d\right)^{2n}$ be two neigboring databases, and consider two independent executions

Algorithm PrivatekGaussians

Input: A database $\mathcal{P} = \{\mathbf{x}_1, \ldots, \mathbf{x}_{2n}\} \in \left(\mathbb{R}^d\right)^{2n}$, parameter $k \in \mathbb{N}$, an accuracy parameter $\alpha > 0$, a confidence parameter $\beta > 0$, privacy parameters $\varepsilon, \delta \in (0,1)$, separating parameter $\gamma > 0$, and bounding parameters $R, \sigma_{\max}, \sigma_{\min} > 0$.
Additional inputs: A (non-private) labeling algorithm $\mathcal{A}$ (according to Definition 6.5), and a private algorithm $\mathcal{A}'$ for learning the parameters of a (single) Gaussian (according to Definition 6.2).

1. Let $\mathcal{P}' = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ and $\mathcal{P}'' = \{\mathbf{x}_{n+1}, \ldots, \mathbf{x}_{2n}\}$.
2. Let $t = t(n, d, k, \beta, \gamma, \varepsilon, \delta, R, \sigma_{\max}, \sigma_{\min})$ be the value from Definition 6.9.
3. Let $s = \left\lfloor \frac{n}{t} \right\rfloor$.
4. Compute $\mathcal{T} = \text{CollectEmpiricalMeans}(\mathcal{P}', k, s, t, \frac{\beta}{8}; \mathcal{A})$.
5. Let $h = 2\sqrt{2\log\left(\frac{8n}{\beta}\right)}$ and let $\Lambda = R + \frac{\gamma h}{16} \cdot \sigma_{\max}$.
6. If $\mathcal{T} \not\subseteq (B(\mathbf{0}, \Lambda)^k)^t$, fail and abort.
7. Execute PrivatekAverages (Figure 3) on the database $\mathcal{T}$ with input parameters $\tilde{\Lambda} = \Lambda, \tilde{n} = t, \tilde{r}_{\min} = \frac{\gamma h}{16} \cdot \sigma_{\min}, \tilde{\varepsilon} = \varepsilon$, $\tilde{\delta} = \delta$, $\tilde{\beta} = \frac{\beta}{4}$.
   Let $\{\hat{\mathbf{a}}_1, \ldots, \hat{\mathbf{a}}_k\}$ be the outcome of the execution.
8. For each $i \in [k]$:
   (a) Let $\mathcal{P}_i''$ be the points in $\mathcal{P}''$ that $\hat{\mathbf{a}}_i$ is the closest point to them among $\{\hat{\mathbf{a}}_1, \ldots, \hat{\mathbf{a}}_k\}$.
   (b) Execute $\mathcal{A}'$ on input $\mathcal{P}_i''$ with bounding parameters $R, \sigma_{\max}, \sigma_{\min}$, privacy parameters $\hat{\varepsilon} = \frac{\varepsilon}{4}, \hat{\delta} = \frac{\delta}{2}$, accuracy parameter $\hat{\alpha} = \frac{\alpha}{2}$ and confidence parameter $\hat{\beta} = \frac{\beta}{8k}$.
   Let $(\hat{\mu}_i, \hat{\Sigma}_i)$ be the outcome of this execution.
   (c) Let $\hat{n}_i \leftarrow |\mathcal{P}_i''| + \text{Lap}(4/\varepsilon)$.
9. For each $i \in [k]$: Set $\hat{w}_i = \frac{\hat{n}_i}{\sum_j \hat{n}_j}$.
10. Output $\hat{\mathcal{D}} = \{(\hat{\mu}_1, \hat{\Sigma}_1, \hat{w}_1), \ldots, (\hat{\mu}_k, \hat{\Sigma}_k, \hat{w}_k)\}$.

Figure 8: Algorithm PrivatekGaussians for privately learning a mixture of k Gaussians.

PrivatekGaussians($\mathcal{P}$) and PrivatekGaussians($\tilde{\mathcal{P}}$) (both with the same other input parameters), let $\mathcal{P}'$, $\mathcal{P}''$, $\mathcal{T}$ be the multisets from the execution PrivatekGaussians($\mathcal{P}$), and let $\tilde{\mathcal{P}}'$, $\tilde{\mathcal{P}}''$, $\tilde{\mathcal{T}}$ be the corresponding multisets in the execution PrivatekGaussians($\tilde{\mathcal{P}}$). If $\mathcal{P}' \neq \tilde{\mathcal{P}}'$ (and therefore,

neighboring), then $\mathcal{T}$ and $\mathcal{T}'$ differ by at most 1 k-tuple. Therefore, by the privacy guarantee of PrivatekAverages (Theorem 4.13) along with group privacy (Fact 2.8) we obtain that the resulting outcome $\{\hat{\mathbf{a}}_1, \ldots, \hat{\mathbf{a}}_k\}$ in Step 7 of both executions is (ε, δ)-indistinguishable. Since $\mathcal{P}' \neq \tilde{\mathcal{P}}'$ implies that $\mathcal{P}'' = \tilde{\mathcal{P}}''$, we conclude by post-processing (Fact 2.9) that the final outcome $\hat{\mathcal{D}}$ is also (ε, δ)-indistinguishable.

In the rest of the analysis we focus on the case that $\mathcal{P}' = \tilde{\mathcal{P}}'$ and $\mathcal{P}'' \neq \tilde{\mathcal{P}}''$ (i.e., neighboring). In this case, the values of $\{\hat{\mathbf{a}}_1, \ldots, \hat{\mathbf{a}}_k\}$ in Step 7 is identical in both executions. Let $\mathcal{P}''_1, \ldots, \mathcal{P}''_k$ be the multisets from Step 8a in the execution PrivatekGaussians($\mathcal{P}$), and let $\tilde{\mathcal{P}}''_1, \ldots, \tilde{\mathcal{P}}''_k$ be these multisets in the execution PrivatekGaussians($\tilde{\mathcal{P}}$). Since $\mathcal{P}''$ and $\tilde{\mathcal{P}}''$ are neighboring, there exists at most two values $i, j \in [k]$ such that $\mathcal{P}''_i \neq \tilde{\mathcal{P}}''_i$ and $\mathcal{P}''_j \neq \tilde{\mathcal{P}}''_j$, and in both cases the multisets are neighboring (in the other indices the multisets are equal). By the properties of the private algorithm $\mathcal{A}'$ and basic composition (Theorem 2.10), the values of $((\hat{\mu}_1, \hat{\Sigma}_1), \ldots, (\hat{\mu}_k, \hat{\Sigma}_k))$ in Step 8b of both executions is $(2 \cdot \frac{\varepsilon}{4}, 2 \cdot \frac{\delta}{2})$-indistinguishable. Moreover, by the properties of the Laplace Mechanism along with basic composition, the values of $(\hat{n}_1, \ldots, \hat{n}_k)$ is $(2 \cdot \frac{\varepsilon}{4}, 0)$-indistinguishable. By applying again basic composition we deduce that all these values together are (ε, δ)-indistinguishable, and therefore we conclude by post-processing (Fact 2.9) that the resulting $\hat{\mathcal{D}}$ in both execution is (ε, δ)-indistinguishable. □

The following theorem summarizes the utility guarantee of PrivatekGaussians.

Theorem 6.11 (Utility of Algorithm PrivatekGaussians)**.** *Let $n, d, k, R, \sigma_{\max}, \sigma_{\min}, w_{\min}, \gamma > 0$, let $\alpha, \beta, \varepsilon, \delta \in (0,1)$, let $t = t(n, d, k, \beta, \gamma, \varepsilon, \delta, R, \sigma_{\max}, \sigma_{\min})$ be the value from Definition 6.9, and let $\mathcal{D} = \{(\mu_1, \Sigma_1, w_1), \ldots, (\mu_k, \Sigma_k, w_k)\}$ be an $(R, \sigma_{\max}, \sigma_{\min}, w_{\min})$-bounded $(1+\gamma)h$-separated mixture of k Gaussians in $\mathbb{R}^d$, for $h \geq 2\sqrt{2\log\left(\frac{8n}{\beta}\right)}$. In addition, let $\mathcal{A}$ be a (non-private) $\left(\lfloor \frac{n}{t} \rfloor, \frac{\beta}{8t}\right)$-labeling algorithm for $\mathcal{D}$ (Definition 6.5), and let $\mathcal{A}'$ be a private algorithm for learning a (single) bounded Gaussian with sample complexity υ (Definition 6.2). Assume that*

$$n \geq \max\left\{ \frac{900t\left(d + 2\log\left(\frac{16kt}{\beta}\right)\right)}{\min\{450, \gamma^2 h^2\} \cdot w_{\min}} + t, \quad \frac{2\upsilon}{w_{\min}}, \quad \frac{4k^2}{\varepsilon\alpha} \cdot \log\left(\frac{8k}{\beta}\right) \right\}$$

where $\upsilon = \upsilon\left(d, \frac{\varepsilon}{2}, \frac{\delta}{2}, \frac{\alpha}{2}, \frac{\beta}{8k}, R, \sigma_{\max}, \sigma_{\min}\right)$. Then with probability $1-\beta$, when sampling a database $\mathcal{P}$ of $2n$ i.i.d. samples from $\mathcal{D}$, Algorithm PrivatekGaussians *on inputs $\mathcal{P}, k, \alpha, \beta, \varepsilon, \delta, \gamma, R, \sigma_{\max}, \sigma_{\min}, \mathcal{A}, \mathcal{A}'$ outputs $\hat{\mathcal{D}}$ such that $\mathrm{d_{TV}}(\mathcal{D}, \hat{\mathcal{D}}) \leq \alpha$.*

The proof of the theorem appears at Appendix B.3. Very roughly, the first term in the maximum is the number of samples that are needed for guaranteeing that with probability at least $1 - \frac{3\beta}{4}$, the partition $\{\mathcal{P}''_1, \ldots, \mathcal{P}''_k\}$ in Step 8a of Algorithm PrivatekGaussians is exactly according to the labels of the points (i.e., two points belong to the same set $\iff$ they were sample from the same Gaussian), and that for each i it holds that $|\mathcal{P}_i| \geq \frac{n}{2w_i}$. The second and third terms in the maximum are the number of samples that are needed for guaranteeing that with probability $1 - \frac{\beta}{4}$, for each $i \in [k]$, the resulting $(\hat{\mu}_i, \hat{\Sigma}_i)$ in Step 8b satisfy $\mathrm{d_{TV}}(\mathcal{N}(\hat{\mu}_i, \hat{\Sigma}_i), \mathcal{N}(\mu_i, \Sigma_i)) \leq \frac{\alpha}{2}$ and the resulting $\hat{w}_i$ in Step 9 satisfy $|\hat{w}_i - w_i| \leq \frac{\alpha}{k}$, which yields that $\mathrm{d_{TV}}(\hat{\mathcal{D}}, \mathcal{D}) \leq \alpha$ (see Fact B.7). We remark that regardless of the non-private algorithm $\mathcal{A}$ that we are using and its assumption on $\mathcal{D}$, we only require that $\mathcal{D}$ is more than $2\sqrt{2\log\left(\frac{8n}{\beta}\right)}$-separated, which follows by the projection argument in Proposition B.6.

6.3 Remarks

It is tempting to think that our approach, which relies on the algorithm PrivatekAverages for aggregating the non-private findings by a reduction to k-tuple clustering, requires that the distance between the means should be proportional to $\sqrt{d}$, because this is the distance of the samples from their means. However, recall that PrivatekGaussians do not set the k-tuple to be some arbitrary k samples from different Gaussians. Rather, it sets it to the *averages* of the samples in each set (See Step 1c in Figure 7), which decreases the distance from the actual means. In particular, when there are $O(d)$ samples in each such set, the dependency in d is eliminated and the reduction to the k-tuple clustering follows (even when the distance between the means is much smaller than $\sqrt{d}$, as we consider).

Furthermore, note that our algorithm PrivatekGaussians in Step 8a relies on the fact that the output $\{\hat{\mathbf{a}}_1, \ldots, \hat{\mathbf{a}}_k\}$ of PrivatekAverages separates correctly fresh samples from the mixture. This might seem strange since even if $\{\hat{\mathbf{a}}_1, \ldots, \hat{\mathbf{a}}_k\}$ is very close to the actual means $\{\mu_1, \ldots, \mu_k\}$, the distance of each sample from its mean is proportional to $\sqrt{d}$, while the assumed separation between the means is independent of d. This yields that when d is large, then the samples are much far from their means compared to the distance between the means. Namely, if $\mathbf{x}$ is sampled from the i'th Gaussian and $\|\mu_i - \mu_j\|$ is independent of d (for large d), then $\|\mathbf{x} - \mu_i\| \gg \|\mu_i - \mu_j\|$. Yet, in our analysis we use a projection argument (see Proposition B.6) which yields that w.h.p. it holds that $\|\mathbf{x} - \mu_i\| < \|\mathbf{x} - \mu_j\|$, even though $\|\mathbf{x} - \mu_i\| \gg \|\mu_i - \mu_j\|$.

6.4 Comparison to the Main Algorithm of [KSSU19]

The main private algorithm of [KSSU19] mimics the approach of the (non-private) algorithm of [AM05], which is to use PCA to project the data into a low-dimensional space, and then clustering the data points in that low-dimensional space. This projection enable both algorithms to learn mixtures that have the following separation

$$\forall i, j: \quad \|\mu_i - \mu_j\| \geq C\left(\sqrt{k\log(nk/\beta)} + \frac{1}{\sqrt{w_i}} + \frac{1}{\sqrt{w_j}}\right) \cdot \max\{\sigma_i, \sigma_j\}, \tag{12}$$

for some constant $C > 0$ (albeit that the constant of [KSSU19] is much larger, say $C = 100$ instead of $C = 4$ as in [AM05]). But while [AM05] use a simple Kruskal-based clustering method, [KSSU19] developed alternative (and much more complicated) clustering methods that are more amenable to privacy. Finally, after the clustering phase, [KSSU19] use a variant of the private algorithm of [KLSU19] to learn the parameters of each Gaussian. Overall, the algorithm of [KSSU19] learns an $(R, \sigma_{\max}, \sigma_{\min}, w_{\min})$-bounded mixture of Gaussian that is separated as in Equation (12), with sample complexity

$$n \geq \left(\frac{d^2}{\alpha^2 w_{\min}} + \frac{d^2}{\varepsilon\alpha w_{\min}} + \frac{\mathrm{poly}(k)d^{3/2}}{w_{\min}\varepsilon}\right) \cdot \mathrm{polylog}\left(\frac{dkR\sigma_{\max}}{\alpha\beta\varepsilon\delta\sigma_{\min}}\right)$$

In the following, we compare between [KSSU19]'s algorithm and ours (Algorithm PrivatekGaussians) in two different aspects: separation assumption and sample complexity.

6.4.1 Separation Assumption

The utility guarantee of PrivatekGaussians (Theorem 6.11) only requires a separation of slightly more than $h = 2\sqrt{2\log(8n/\beta)}$. Therefore, our algorithm can transform any non-private algorithm

(in a modular way) that learns mixtures with separation X into a private algorithm that learns with separation $\max\{X, h\}$. In particular, we can use [AM05] as our non-private labeling algorithm $\mathcal{A}$ to learn mixtures with separation as in Equation (12) (with the small constant $C = 4$), and we can also use any other non-private algorithm (like [VW04; HL18a; KSS18; DKS18; RV17]) and inherent their separation assumption. In contrast, the approach of the main algorithm of [KSSU19] may only be extended to methods that use statistical properties of the data (like PCA), and not to other algorithmic machineries such as the sum-of-squares that are used for reducing the separation assumption.

6.4.2 Sample Complexity

The main algorithm of [KSSU19] learns an $(R, \sigma_{\max}, \sigma_{\min}, w_{\min})$-bounded mixture of Gaussians that is separated as in Equation (12), with sample complexity (roughly) $\tilde{O}\left(\frac{\upsilon}{w_{\min}} + \frac{k^9 d^{3/2}}{w_{\min}\varepsilon}\right)$ (ignoring logarithmic factors), where $\upsilon = \upsilon(d, \varepsilon, \delta, \alpha, \beta, R, \sigma_{\max}, \sigma_{\min}) = \tilde{O}\left(\frac{d^2}{\alpha^2} + \frac{d^2}{\varepsilon\alpha}\right)$ is the sample complexity of [KLSU19] for learning the parameters of a single Gaussian.

By Theorem 6.11, the sample complexity of our algorithm is $\tilde{O}\left(t \cdot \hat{\upsilon} + \frac{t \cdot d}{w_{\min}} + \frac{\upsilon}{w_{\min}} + \frac{4k^2}{\varepsilon\alpha}\right)$ (ignoring logarithmic factors), where $\hat{\upsilon}$ is the sample complexity needed by the non-private algorithm $\mathcal{A}$ for labeling correctly the samples with confidence $\leq \frac{\beta}{8t}$ (e.g., if we use the algorithm of [AM05], then $\hat{\upsilon} = \tilde{O}\left(\frac{dk}{w_{\min}}\right)$, and for simplifying the comparison, we assume that this is indeed the algorithm that we use). Since $t = \tilde{O}\left(\frac{dk}{\varepsilon^2}\right)$, we obtain a sample complexity of (roughly) $\tilde{O}\left(\frac{k^2 d^2}{\varepsilon^2 w_{\min}} + \frac{\upsilon}{w_{\min}} + \frac{4k^2}{\varepsilon\alpha}\right)$, which might me larger than the one of [KSSU19] if d or $1/\varepsilon$ are very large (compared to k). Yet, we can easily improve the dependency in both d and ε.

Using sub-sampling, we can execute Steps 2 to 7 of PrivatekGaussians on an εn-size random subset of $\mathcal{P}'$ (for the small desired ε), but now we only need a constant ε for these steps. This immediately reduces the $1/\varepsilon^2$ in our sample complexity into $1/\varepsilon$.

In addition, as mentioned in Section 4.3.2, using the average algorithm of [NSV16] in PrivatekAverages (instead of the average algorithm from Proposition 2.23), we can reduce a factor of $\sqrt{d}$.

For summary, using sub-sampling and the algorithm of [NSV16], we obtain an improved sample complexity of $\tilde{O}\left(\frac{k^2 d^{3/2}}{\varepsilon w_{\min}} + \frac{\upsilon}{w_{\min}} + \frac{4k^2}{\varepsilon\alpha}\right)$, which strictly improves the sample complexity of [KSSU19].

7 Empirical Results

We implemented in Python our two main algorithms for k-tuple clustering: PrivatekAverages and PrivatekNoisyCenters. We compared the two algorithms in terms of the sample complexity that is needed to privately separate the samples from a given mixture of Gaussians. Namely, how many k-tuples we need to sample such that, when executing PrivatekAverages or PrivatekNoisyCenters, the resulting k-tuple $Y = \{\mathbf{y}_1, \ldots, \mathbf{y}_k\}$ satisfies the following requirement: For every $i \in [k]$, there exists a point in Y (call it $\mathbf{y}_i$), such that for every sample $\mathbf{x}$ that was drawn from the i'th Gaussian, it holds that $i = \operatorname{argmin}_{j \in [k]} \|\mathbf{x} - \mathbf{y}_j\|$. We perform three tests, where in each test we considered a uniform mixture of k standard spherical Gaussians around the means $\{R \cdot \mathbf{e}_i, -R \cdot \mathbf{e}_i\}_{i=1}^{k/2}$, where $\mathbf{e}_i$ is the i'th standard basis vector. In all the tests, we generated each k-tuple by running algorithm k-means++ [AV07] over enough samples.

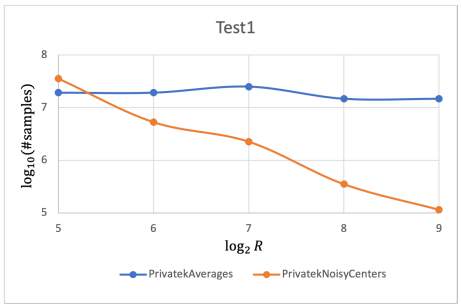

Figure 9: The case $d = 1$ and $k = 2$, for varies R.

Figure 10: The case $d = 4$ and $R = 512 \cdot k$, for varies k.

In Test1 (Figure 9) we examined the sample complexity in the case $d = 1$, $k = 2$, for $R \in \{2^5, 2^6, \ldots, 2^9\}$. In Test2 (Figure 10) we examined the case $d = 4$, $R = 512 \cdot k$, for $k \in \{2, 4, 6, 8\}$. In Test3 (Figure 11) we examined the case $k = 2$, $R = 256\sqrt{d}$, for $d \in \{4, 8, 12, 16\}$. In all the experiments we used privacy parameters $\varepsilon = 1$ and $\delta = e^{-28}$, and used $\beta = 0.05$. In all the tests of PrivatekNoisyCenters, we chose $\Delta = \frac{10}{\varepsilon} \cdot k \log(k/\delta)\sqrt{\log(k/\beta)}$, the number of k-tuples that we generated was exactly 3781 (the minimal value that is required for privacy), but the number of samples per k-tuple varied from test to test. In the tests of PrivatekAverages, we chose $\Lambda = 2^{10} \cdot k\sqrt{d}$ and $r_{\min} = 0.1$, we generated each k-tuple using $\approx 15 \cdot k$ samples, but the number of k-tuples varied from test to test.[4] All the experiments were tested in a MacBook Pro Laptop with 4-core Intel i7 CPU with 2.8GHz, and with 16GB RAM.

The graphs show the main bottleneck of Algorithm PrivatekAverages in practice. It requires only $O_{\varepsilon,\delta}(kd)$ tuples (or $O_{\varepsilon,\delta}(k\sqrt{d})$ for large values of d) in order to succeed, but the hidden constant is $\approx 500,000$ for our choice of ε and δ, and this does not improve even when the assumed separation R is very large. The cause of this large constant is the group privacy of size $O(k\ell)$ that we do in Step 5a, where recall that $\ell = O\left(\frac{\log^2(1/\delta)}{\varepsilon \log n}\right)$ (Definition 4.8). While in theory this ℓ is relatively small, with our choice of parameters we get $\ell \approx 1000$. This means that we need to execute the private average algorithm with $\hat{\varepsilon} \approx \frac{\varepsilon}{4000k}$. Internally, this $\hat{\varepsilon}$ is shared between other private algorithms, and in particular, with an Interior Point algorithm that is one of the internal components of the average algorithm from Proposition 2.23. This algorithm is implemented using the exponential mechanism [MT07], which simply outputs a random noise when the number of points is too small.

[4] By using $\tilde{\Omega}(kd)$ samples for creating each k-tuple, in Test3 (Figure 11) we could avoid the dependency of R in $\sqrt{d}$ (see Section 6.3 for more details). However, since we only used $O(k)$ samples for each k-tuple when testing PrivatekAverages, then we could not avoid this dependency.

We remark that prior work on differentially-private clustering, including in "easy" settings, is primarily theoretical. In particular, we are not aware of implemented methods that we could use as a baseline.[5] As a sanity check, we did consider the following naive baseline: For every sample point, add a Gaussian noise to make it private. Now, the resulting noisy samples are just samples from a new Gaussian mixture. Then, run an off-the-shelf non-private method to learn the parameters of this mixture. We tested this naive method on the simple case $d = 1$ and $k = 2$, where we generated samples from a mixture of standard Gaussians that are separated by $R = 512$. By the Gaussian mechanism, the noise magnitude that we need to add to each point for guaranteeing (ε, δ)-differential privacy, is $\sigma \approx \frac{\Lambda}{\varepsilon}\sqrt{\log(1/\delta)} \gg 1$ for some $\Lambda > R$, meaning that the resulting mixture consists of very close Gaussians. We applied GaussianMixture from the package sklearn.mixture to learn this mixture, but it failed even when we used $100M$ samples, as this method is not intended for learning such close Gaussians.We remark that there are other non-private methods that are designed to learn any mixture of Gaussians (even very weakly separated ones) using enough samples (e.g., [SOAJ14]). The sample complexity and running time of these methods, however, are much worse than ours even asymptotically (e.g., the running time of [SOAJ14] is exponential in k), and moreover, we are not aware of any implementation we could use.[6]

8 Conclusion

We developed an approach to bridge the gap between the theory and practice of differentially private clustering methods. For future, we hope to further optimize the "constants" in the k-tuple clustering algorithms, making the approach practical for instances with lower separation. Tangentially, the inherent limitations of private versus non-private clustering suggest exploring different rigorous notions of privacy in the context of clustering.

Acknowledgements

Edith Cohen is supported by Israel Science Foundation grant no. 1595-19.

[5] We remark that in different settings, such as node, edge or weight-differential privacy, there exist some available implementations (e.g., [PMY+18]).

[6] Asymptotically, [SOAJ14] requires at least $\tilde{\Omega}(dk^9)$ samples, and runs in time $\tilde{\Omega}(n^2d + d^2(k^7 \log d)^{k^2})$. For the setting of learning a mixture of k well-separated Gaussians, the approach of first adding noise to each point and then applying a non-private method such as [SOAJ14], results with much worse parameters than our result, which only requires $\tilde{O}(dk)$ samples and runs in time $\tilde{O}(dk^2n)$.

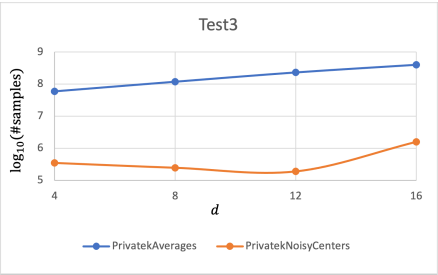

Figure 11: The case $k = 2$, $R = 256\sqrt{d}$, for varies d.

Haim Kaplan is supported by Israel Science Foundation grant no. 1595-19, and the Blavatnik Family Foundation.

Yishay Mansour has received funding from the European Research Council (ERC) under the European Union'sHorizon 2020 research and innovation program (grant agreement No. 882396), by the Israel Science Foundation (grant number 993/17) and the Yandex Initiative for Machine Learning at Tel Aviv University.

Uri Stemmer is partially supported by the Israel Science Foundation (grant 1871/19) and by the Cyber Security Research Center at Ben-Gurion University of the Negev.

References

[ABS10] P. Awasthi, A. Blum, and O. Sheffet, "Stability yields a ptas for k-median and k-means clustering," in *2010 IEEE 51st Annual Symposium on Foundations of Computer Science*, IEEE, 2010, pp. 309–318 (cit. on p. 3).

[ABS12] ——, "Center-based clustering under perturbation stability," *Information Processing Letters*, vol. 112, no. 1-2, pp. 49–54, 2012 (cit. on p. 3).

[AM05] D. Achlioptas and F. McSherry, "On spectral learning of mixtures of distributions," in *Learning Theory, 18th Annual Conference on Learning Theory, COLT 2005*, vol. 3559, 2005, pp. 458–469 (cit. on pp. 29, 33, 34).

[ANFSW19] S. Ahmadian, A. Norouzi-Fard, O. Svensson, and J. Ward, "Better guarantees for k-means and euclidean k-median by primal-dual algorithms," *SIAM Journal on Computing*, no. 0, FOCS17–97, 2019 (cit. on p. 2).

[AV07] D. Arthur and S. Vassilvitskii, "K-means++ the advantages of careful seeding," in *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, 2007, pp. 1027–1035 (cit. on p. 34).

[BBG09] M.-F. Balcan, A. Blum, and A. Gupta, "Approximate clustering without the approximation," in *Proceedings of the twentieth annual ACM-SIAM symposium on Discrete algorithms*, SIAM, 2009, pp. 1068–1077 (cit. on p. 3).

[BDKU20] S. Biswas, Y. Dong, G. Kamath, and J. R. Ullman, "Coinpress: Practical private mean and covariance estimation," in *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS*, 2020 (cit. on pp. 3, 4, 28).

[BDL+17] M.-F. Balcan, T. Dick, Y. Liang, W. Mou, and H. Zhang, "Differentially private clustering in high-dimensional Euclidean spaces," in *Proceedings of the 34th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, vol. 70, 2017, pp. 322–331 (cit. on pp. 1, 4).

[BDMN05a] A. Blum, C. Dwork, F. McSherry, and K. Nissim, "Practical privacy: The SuLQ framework," in *PODS*, C. Li, Ed., ACM, 2005, pp. 128–138 (cit. on p. 1).

[BDMN05b] ——, "Practical privacy: The SuLQ framework," in *PODS*, 2005, pp. 128–138 (cit. on p. 4).

[BKN10] A. Beimel, S. P. Kasiviswanathan, and K. Nissim, "Bounds on the sample complexity for private learning and private data release," in *TCC*, ser. LNCS, vol. 5978, Springer, 2010, pp. 437–454 (cit. on p. 8).

[BKSW21] M. Bun, G. Kamath, T. Steinke, and Z. S. Wu, "Private hypothesis selection," *IEEE Transactions on Information Theory*, 2021 (cit. on p. 3).

[BL12] Y. Bilu and N. Linial, "Are stable instances easy?" *Combinatorics, Probability and Computing*, vol. 21, no. 5, pp. 643–660, 2012 (cit. on p. 3).

[BNSV15] M. Bun, K. Nissim, U. Stemmer, and S. P. Vadhan, "Differentially private release and learning of threshold functions," in *FOCS*, 2015, pp. 634–649 (cit. on p. 8).

[BS16] M. Bun and T. Steinke, "Concentrated differential privacy: Simplifications, extensions, and lower bounds," in *Theory of Cryptography - 14th International Conference, TCC 2016*, vol. 9985, 2016, pp. 635–658 (cit. on p. 4).

[BS19] ——, "Average-case averages: Private algorithms for smooth sensitivity and mean estimation," in *Advances in Neural Information Processing Systems*, 2019, pp. 181–191 (cit. on pp. 3, 4).

[CO13] A. Coja-Oghlan, "Probabilistic combinatorics," 2013. [Online]. Available: https://www.math.uni-frankfurt.de/~acoghlan/probcomb.pdf (cit. on p. 8).

[CSS13] K. Chaudhuri, A. D. Sarwate, and K. Sinha, "A near-optimal algorithm for differentially-private principal components.," *Journal of Machine Learning Research*, vol. 14, 2013 (cit. on p. 4).

[CWZ19] T. T. Cai, Y. Wang, and L. Zhang, "The cost of privacy: Optimal rates of convergence for parameter estimation with differential privacy," *arXiv preprint arXiv:1902.04495*, 2019 (cit. on pp. 3, 4).

[DKM+06] C. Dwork, K. Kenthapadi, F. McSherry, I. Mironov, and M. Naor, "Our data, ourselves: Privacy via distributed noise generation," in *EUROCRYPT*, S. Vaudenay, Ed., ser. Lecture Notes in Computer Science, vol. 4004, Springer, 2006, pp. 486–503 (cit. on p. 6).

[DKS18] I. Diakonikolas, D. M. Kane, and A. Stewart, "List-decodable robust mean estimation and learning mixtures of spherical gaussians," in *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*, 2018, pp. 1047–1060 (cit. on pp. 29, 34).

[DMNS06] C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating noise to sensitivity in private data analysis," in *TCC*, Springer, 2006, pp. 265–284 (cit. on pp. 1, 4, 6).

[DRV10] C. Dwork, G. N. Rothblum, and S. P. Vadhan, "Boosting and differential privacy," in *FOCS*, IEEE Computer Society, 2010, pp. 51–60 (cit. on p. 6).

[DS00] S. Dasgupta and L. J. Schulman, "A two-round variant of EM for gaussian mixtures," in *UAI '00: Proceedings of the 16th Conference in Uncertainty in Artificial Intelligence*, 2000, pp. 152–159 (cit. on p. 29).

[DTTZ14] C. Dwork, K. Talwar, A. Thakurta, and L. Zhang, "Analyze gauss: Optimal bounds for privacy-preserving principal component analysis," in *Proceedings of the 46th Annual ACM Symposium on Theory of Computing*, ser. STOC '14, ACM, 2014, pp. 11–20 (cit. on p. 4).

[FFKN09] D. Feldman, A. Fiat, H. Kaplan, and K. Nissim, "Private coresets," in *Proceedings of the 41st Annual ACM Symposium on Theory of Computing, STOC 2009*, 2009, pp. 361–370 (cit. on p. 1).

[FXZR17] D. Feldman, C. Xiang, R. Zhu, and D. Rus, "Coresets for differentially private k-means clustering and applications to privacy in mobile sensor networks," in *Proceedings of the 16th ACM/IEEE International Conference on Information Processing in Sensor Networks*, 2017, pp. 3–15 (cit. on p. 1).

[GKM20] B. Ghazi, R. Kumar, and P. Manurangsi, "Differentially private clustering: Tight approximation ratios," in *NeurIPS*, 2020 (cit. on pp. 1, 4).

[GLM+10] A. Gupta, K. Ligett, F. McSherry, A. Roth, and K. Talwar, "Differentially private combinatorial optimization," in *Proceedings of the Twenty-first Annual ACM-SIAM Symposium on Discrete Algorithms*, ser. SODA '10, 2010, pp. 1106–1125 (cit. on pp. 1, 4).

[HL18a] S. B. Hopkins and J. Li, "Mixture models, robustness, and sum of squares proofs," in *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*, 2018, pp. 1021–1034 (cit. on pp. 29, 34).

[HL18b] Z. Huang and J. Liu, "Optimal differentially private algorithms for k-means clustering," in *Proceedings of the 37th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems*, 2018, pp. 395–408 (cit. on pp. 1, 3, 4).

[HW71] D. L. Hanson and F. T. Wright, "A bound on tail probabilities for quadratic forms in independent random variables," *Annals of Mathematical Statistics*, vol. 42, no. 3, pp. 1079–1083, 1971 (cit. on p. 29).

[JL84] W. Johnson and J. Lindenstrauss, "Extensions of lipschitz maps into a hilbert space," *Contemporary Mathematics*, vol. 26, pp. 189–206, 1984 (cit. on p. 7).

[KK10] A. Kumar and R. Kannan, "Clustering with spectral norm and the k-means algorithm," in *2010 IEEE 51st Annual Symposium on Foundations of Computer Science*, IEEE, 2010, pp. 299–308 (cit. on p. 3).

[KLN+11] S. P. Kasiviswanathan, H. K. Lee, K. Nissim, S. Raskhodnikova, and A. D. Smith, "What can we learn privately?" *SIAM J. Comput.*, vol. 40, no. 3, pp. 793–826, 2011 (cit. on p. 8).

[KLSU19] G. Kamath, J. Li, V. Singhal, and J. Ullman, "Privately learning high-dimensional distributions," in *Conference on Learning Theory, COLT 2019*, vol. 99, 2019, pp. 1853–1902 (cit. on pp. 3, 4, 28, 33, 34).

[KS18] H. Kaplan and U. Stemmer, "Differentially private k-means with constant multiplicative error," in *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018*, 2018, pp. 5436–5446 (cit. on pp. 1, 4).

[KSS18] P. K. Kothari, J. Steinhardt, and D. Steurer, "Robust moment estimation and improved clustering via sum of squares," in *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*, 2018, pp. 1035–1046 (cit. on pp. 29, 34).

[KSSU19] G. Kamath, O. Sheffet, V. Singhal, and J. Ullman, "Differentially private algorithms for learning mixtures of separated gaussians," in *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019*, 2019, pp. 168–180 (cit. on pp. 3, 33, 34).

[KSU20] G. Kamath, V. Singhal, and J. Ullman, "Private mean estimation of heavy-tailed distributions," *arXiv preprint arXiv:2002.09464*, 2020 (cit. on pp. 3, 4).

[KT13] M. Kapralov and K. Talwar, "On differentially private low rank approximation," in *Proceedings of the twenty-fourth annual ACM-SIAM symposium on Discrete algorithms*, 2013, pp. 1395–1414 (cit. on p. 4).

[KV18] V. Karwa and S. Vadhan, "Finite sample differentially private confidence intervals," in *9th Innovations in Theoretical Computer Science Conference (ITCS 2018)*, 2018 (cit. on pp. 3, 4).

[LSW17] E. Lee, M. Schmidt, and J. Wright, "Improved and simplified inapproximability for k-means," *Information Processing Letters*, vol. 120, pp. 40–43, 2017 (cit. on p. 2).

[McS09] F. McSherry, "Privacy integrated queries: An extensible platform for privacy-preserving data analysis," in *Proceedings of the ACM SIGMOD International Conference on Management of Data, SIGMOD 2009*, 2009, pp. 19–30 (cit. on p. 1).

[MT07] F. McSherry and K. Talwar, "Mechanism design via differential privacy," in *FOCS*, IEEE Computer Society, 2007, pp. 94–103 (cit. on pp. 35, 41).

[MTS+12] P. Mohan, A. Thakurta, E. Shi, D. Song, and D. Culler, "Gupt: Privacy preserving data analysis made easy," in *Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data*, ser. SIGMOD '12, 2012, pp. 349–360 (cit. on p. 1).

[NCBN16] R. Nock, R. Canyasse, R. Boreli, and F. Nielsen, "K-variates++: More pluses in the k-means++," in *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016*, 2016, pp. 145–154 (cit. on p. 1).

[Ngu20] H. L. Nguyen, "A note on differentially private clustering with large additive error," *CoRR*, vol. abs/2009.13317, 2020 (cit. on p. 1).

[NRS07] K. Nissim, S. Raskhodnikova, and A. Smith, "Smooth sensitivity and sampling in private data analysis," in *STOC*, ACM, 2007, pp. 75–84 (cit. on pp. 1, 3, 4).

[NS17] K. Nissim and U. Stemmer, "Clustering algorithms for the centralized and local models," *CoRR*, vol. abs/1707.04766, 2017. [Online]. Available: http://arxiv.org/abs/1707.04766 (cit. on p. 4).

[NS18] ——, "Clustering algorithms for the centralized and local models," in *Proceedings of Algorithmic Learning Theory*, ser. Proceedings of Machine Learning Research, vol. 83, 2018, pp. 619–653 (cit. on p. 1).

[NSV16] K. Nissim, U. Stemmer, and S. P. Vadhan, "Locating a small cluster privately," in *Proceedings of the 35th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems, PODS 2016, San Francisco, CA, USA, June 26 - July 01, 2016*, 2016, pp. 413–427 (cit. on pp. 1, 4, 7, 18, 34).

[ORSS12] R. Ostrovsky, Y. Rabani, L. J. Schulman, and C. Swamy, "The effectiveness of lloyd-type methods for the k-means problem," *J. ACM*, vol. 59, no. 6, 28:1–28:22, 2012 (cit. on pp. 3, 21, 26).

[PMY+18] R. Pinot, A. Morvan, F. Yger, C. Gouy-Pailler, and J. Atif, "Graph-based clustering under differential privacy," in *Proceedings of the Thirty-Fourth Conference on Uncertainty in Artificial Intelligence, UAI 2018*, A. Globerson and R. Silva, Eds., 2018, pp. 329–338 (cit. on p. 36).

[RV17] O. Regev and A. Vijayaraghavan, "On learning mixtures of well-separated gaussians," in *2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS)*, IEEE, 2017, pp. 85–96 (cit. on pp. 29, 34).

[SCL+16] D. Su, J. Cao, N. Li, E. Bertino, and H. Jin, "Differentially private k-means clustering," in *Proceedings of the Sixth ACM Conference on Data and Application Security and Privacy*, ser. CODASPY '16, 2016, pp. 26–37 (cit. on p. 1).

[SK01] A. Sanjeev and R. Kannan, "Learning mixtures of arbitrary gaussians," in *Proceedings of the thirty-third annual ACM symposium on Theory of computing*, 2001, pp. 247–257 (cit. on p. 29).

[SOAJ14] A. T. Suresh, A. Orlitsky, J. Acharya, and A. Jafarpour, "Near-optimal-sample estimators for spherical gaussian mixtures," *Advances in Neural Information Processing Systems*, vol. 27, pp. 1395–1403, 2014 (cit. on p. 36).

[SSS20] M. Shechner, O. Sheffet, and U. Stemmer, "Private k-means clustering with stability assumptions," in *The 23rd International Conference on Artificial Intelligence and Statistics, AISTATS 2020*, ser. Proceedings of Machine Learning Research, vol. 108, 2020, pp. 2518–2528 (cit. on pp. 1, 3, 26, 45).

[Ste20] U. Stemmer, "Locally private k-means clustering," in *SODA*, SIAM, 2020 (cit. on p. 1).

[VW04] S. Vempala and G. Wang, "A spectral algorithm for learning mixture models," *Journal of Computer and System Sciences*, vol. 68, no. 4, pp. 841–860, 2004 (cit. on pp. 29, 34).

[WWS15] Y. Wang, Y.-X. Wang, and A. Singh, "Differentially private subspace clustering," in *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1*, ser. NIPS'15, 2015, pp. 1000–1008 (cit. on pp. 1, 3).

A Additional Preliminaries

A.1 Additional Facts About Differential Privacy

A.1.1 The Exponential Mechanism

We next describe the Exponential Mechanism of McSherry and Talwar [MT07]. Let $\mathcal{X}$ be a domain and $\mathcal{H}$ a set of solutions. Given a database $\mathcal{S} \in \mathcal{X}^*$, the Exponential Mechanism privately chooses a "good" solution h out of the possible set of solutions $\mathcal{H}$. This "goodness" is quantified using a quality function that matches solutions to scores.

Definition A.1. *(Quality function) A quality function is a function $q\colon \mathcal{X}^* \times \mathcal{H} \mapsto \mathbb{R}$ that maps a database $\mathcal{S} \in \mathcal{X}^*$ and a solution $h \in \mathcal{H}$ to a real number, identified as the score of the solution h w.r.t the database $\mathcal{S}$.*

Given a quality function q and a database $\mathcal{S}$, the goal is to chooses a solution h approximately maximizing $q(\mathcal{S}, h)$. The Exponential Mechanism chooses a solution probabilistically, where the probability mass that is assigned to each solution h increases exponentially with its quality $q(\mathcal{S}, h)$:

Definition A.2. *(The Exponential Mechanism) Given input parameter ε, finite solution set $\mathcal{H}$, database $\mathcal{S} \in \mathcal{X}^m$, and a sensitivity 1 quality function q, choose randomly $h \in \mathcal{H}$ with probability proportional to $\exp(\varepsilon \cdot q(\mathcal{S}, h)/2)$.*

Proposition A.3. *(Properties of the Exponential Mechanism) (i) The Exponential Mechanism is ε-differentially private. (ii) Let $\hat{e} := \max_{f \in \mathcal{H}}\{q(\mathcal{S}, f)\}$ and $\Delta > 0$. The Exponential Mechanism outputs a solution h such that $q(\mathcal{S}, h) \leq \hat{e} - \Delta$ with probability at most $|\mathcal{H}| \cdot \exp(-\varepsilon\Delta/2)$.*

A.1.2 Private Interior Point and Bounding Segment in $\mathbb{R}$

Proposition A.4 (Finding an Interior Point in $\mathbb{R}$)**.** *Let $\varepsilon \in (0, 1)$, $\Lambda > 0$ and $g \in [0, \Lambda]$. There exists an efficient ε-differentially private algorithm that takes an n-size database $\mathcal{S}$ of numbers in the segment $[-\Lambda, \Lambda]$ and outputs a number $z \in [-\Lambda, \Lambda]$ that with probability $1 - 2(\Lambda/g + 1) \cdot \exp(-\varepsilon n/4)$ it holds that $z \in [\min(\mathcal{S}) - g, \max(\mathcal{S}) + g]$. The algorithm runs in time $\tilde{O}(n)$ (ignoring $\log\left(\frac{n\Lambda}{g}\right)$ factors).*

Proof. Define the grid $G = \{-\Lambda, -\Lambda + g, \ldots, -\Lambda + \left\lceil \frac{2\Lambda}{g} \right\rceil \cdot g\}$, and for every $x \in G$ let $\text{left}(x) = -\Lambda + \left\lfloor \frac{x+\Lambda}{g} \right\rfloor \cdot g$ (i.e., the closest grid point to x from the left side) and $\text{right}(x) = -\Lambda + \left\lceil \frac{x+\Lambda}{g} \right\rceil \cdot g$ (i.e., the closest grid point to x from the right side). Now, apply the exponential mechanism (A.2) with the quality function

$$\forall y \in G\colon \quad q(\mathcal{S}, y) = \min\{|\{x \in \mathcal{S}\colon \text{left}(x) \leq y\}|, |\{x \in \mathcal{S}\colon \text{right}(x) \geq y\}|\}$$

For the utility analysis, let m be the median of $\mathcal{S}$, and note that $q(\mathcal{S}, \text{left}(m)), q(\mathcal{S}, \text{right}(m)) \geq n/2$. Therefore, by Proposition A.3, with probability $\geq 1 - |G| \cdot \exp(-\varepsilon n/4) \geq 1 - 2(\Lambda/g + 1) \cdot \exp(-\varepsilon n/4)$, the mechanism outputs a point z with $q(S, z) > 0$, which yields in particular that $z \in [\min(\mathcal{S}) - g, \max(\mathcal{S}) + g]$.

For the running time analysis, we implement the sampling as follows: For $x \in \mathcal{S}$ we let $A_x = \{\text{left}(x) - g, \text{left}(x), \text{right}(x), \text{right}(x) + g\}$, and let $A = \cup_{x \in \mathcal{S}} A_x$. Note that for every consecutive grid points $y, y' = y + g \in G$ with $q(\mathcal{S}, y) \neq q(\mathcal{S}, y')$, it holds that $y, y' \in A$: If $q(\mathcal{S}, y) > q(\mathcal{S}, y')$, there must exist $x \in \mathcal{S}$ such that $x \in (y - g, y]$, yielding that $y \in [x, x + g) \implies y = \text{right}(x), y' = \text{right}(x) + g$. Otherwise (i.e., $q(\mathcal{S}, y) < q(\mathcal{S}, y')$), there must exist $x \in \mathcal{S}$ such that $x \in [y', y' + g)$, yielding that $y' \in (x - g, x] \implies y' = \text{left}(x), y = \text{left}(x) - g$.

Then, we sort A in time $\tilde{O}(n)$, and let $a_1 \leq \ldots \leq a_m$ be the sorted elements in A (recall that $m = |A| \leq 4n$). For each $i \in [m + 1]$, we compute $w(\mathcal{S}, a_i) = q(\mathcal{S}, a_i) \cdot |G \cap (a_{i-1}, a_i]|$ (i.e., $w(\mathcal{S}, a_i)$ is the the original quality of a_i times the number of grid points in $(a_{i-1}, a_i]$, where $a_0 = -\Lambda - g$ and $a_{m+1} = \Lambda + g$), and choose a value a_i with probability $\propto w(\mathcal{S}, a_i)$. Note that the computation of each $w(\mathcal{S}, a_i)$ can be done in time $\tilde{O}(1)$ using simple binary searches over the (sorted) multisets $\mathcal{S}_{\text{left}} = \cup_{x \in \mathcal{S}}\{\text{left}(x)\}$ and $\mathcal{S}_{\text{right}} = \cup_{x \in \mathcal{S}}\{\text{right}(x)\}$ (a "multiset" union, that includes duplications).

Finally, given the chosen a_i from the mechanism, it is left to sample a uniform point in $G \cap (a_{i-1}, a_i]$ (since we know, by the property of A, that all the point there have the same value of $q(\mathcal{S}, \cdot)$). This can be easily implemented in time $O(\log|G|) = \tilde{O}(1)$. □

Proposition A.5 (Finding a Bounding Segment of Points in $\mathbb{R}$)**.** *Let $\beta, \varepsilon \in (0,1)$, $\Lambda > 0$ and $g \in [0, \Lambda]$. There exists an efficient ε-differentially private algorithm that takes an n-size database $\mathcal{S}$ of numbers in the segment $[-\Lambda, \Lambda]$ and outputs a segment $[x, y]$ such that with probability at least $1 - \beta$ the following holds:*

- $|\mathcal{S} \cap [x, y]| \geq n - \frac{8}{\varepsilon} \log\left(\frac{4\Lambda}{g\beta}\right) - 2$ *(i.e., the segment contain most of the points in $\mathcal{S}$), and*
- $y - x \leq \max(\mathcal{S}) - \min(\mathcal{S}) + 4g$.

The algorithm runs in time $\tilde{O}(n)$ (ignoring $\log\left(\frac{n\Delta}{\varepsilon\beta g}\right)$ factors).

Proof. In the following assume that $n \geq \frac{8}{\varepsilon} \log\left(\frac{4\Lambda}{g\beta}\right) + 2$ (otherwise the proof trivially holds for any segment $[x, x]$). Let $\mathcal{S}_0$ be the smallest $\frac{4}{\varepsilon} \log\left(\frac{4\Lambda}{g\beta}\right) + 1$ points in $\mathcal{S}$, and let $\mathcal{S}_1$ be the largest $\frac{4}{\varepsilon} \log\left(\frac{4\Lambda}{g\beta}\right) + 1$ points in $\mathcal{S}$. For each $b \in \{0,1\}$ apply Proposition A.4 (interior point) on $\mathcal{S}_b$ for finding a number $z_b \in [-\Lambda, \Lambda]$ that belongs to $[\min(\mathcal{S}_b) - g, \max(\mathcal{S}_b) + g]$ with probability at least $1 - 2(\Lambda/g + 1) \cdot \exp(-\varepsilon|\mathcal{S}_b|/4) \geq 1 - \beta/2$. Therefore, by setting $x = z_0 - g$ and $y = z_1 + g$ we get that with probability $1 - \beta$ it holds that: (1) $[\max(\mathcal{S}_0), \min(\mathcal{S}_1)] \subseteq [x, y]$ and that (2) $[x, y] \subseteq [\min(\mathcal{S}_0) - 2g, \max(\mathcal{S}_1) + 2g] = [\min(\mathcal{S}) - 2g, \max(\mathcal{S}) + 2g]$. By (1) we get that all points in $\mathcal{S}$ expect (at most) $(|\mathcal{S}_0| - 1) + (|\mathcal{S}_1| - 1) \leq \frac{8}{\varepsilon} \log\left(\frac{4\Lambda}{g\beta}\right)$ are inside $[x, y]$, and by (2) we get that $y - x \geq \max(\mathcal{S}) - \min(\mathcal{S}) + 4g$, as required.

For the running time analysis, note that by sorting $\mathcal{S}$ we can determine $\mathcal{S}_0$ and $\mathcal{S}_1$ in time $\tilde{O}(n)$, and the cost of executing the algorithm from Proposition A.4 on each $\mathcal{S}_b$ is $\tilde{O}\left(\frac{1}{\varepsilon}\right) = \tilde{O}(n)$. □

A.1.3 Estimating the Average of Points

Proposition A.6 (Estimating the Average of Bounded Points in $\mathbb{R}$)**.** *Let $\beta, \varepsilon, \delta \in (0,1)$, $\Lambda > 0$ and $r_{\min} \in [0, \Lambda]$. There exists an efficient (ε, δ)-differentially private algorithm that takes an n-size database $\mathcal{S}$ of numbers in the segment $[-\Lambda, \Lambda]$ and satisfy the following utility guarantee: If $n \geq \frac{16}{\varepsilon} \log\left(\frac{4\Lambda}{r_{\min}\beta}\right) + 4$, then with probability $1 - \beta$, the algorithm outputs a number $\hat{a} \in \mathbb{R}$ such that*

$$|\hat{a} - \mathrm{Avg}(\mathcal{S})| \leq O\left(\frac{\max\{r, r_{\min}\}}{\varepsilon n}\left(\sqrt{\log(1/\delta)\log(1/\beta)} + \log\left(\frac{\Lambda}{r_{\min}\beta}\right)\right)\right),$$

where $r = \max(\mathcal{S}) - \min(\mathcal{S})$. The algorithm runs in time $\tilde{O}(n)$ (ignoring $\log\left(\frac{n\Delta}{r_{\min}\varepsilon\beta}\right)$ factors).

Proof. The algorithm does the following: (1) Privately find a bounding segment $[x, y]$ using Proposition A.5 with parameters $\beta/2, \varepsilon/2, g = r_{\min}, \Lambda$, let $\hat{r} = y - x$ and let $\mathcal{S}' = \mathcal{S} \cap [x, y]$ (2) Use the (1-dimensional) Gaussian mechanism (Theorem 2.19) with $\lambda = \frac{\hat{r}}{|\mathcal{S}'|}$ and parameters $\beta/2, \varepsilon/2, \delta$ for computing a noisy average $\hat{a}$ of $\mathcal{S}'$ (see Observation 2.20). By the properties of the Gaussian mechanism (see Remark 2.21) along with basic composition it holds that the above algorithm is (ε, δ)-differentially private. For the utility analysis, note that with probability $1 - \beta$, the segment

$[x, y]$ satisfies the conditions of Proposition A.5 and the noise added to the average in the second step is at most $O\left(\frac{\hat{r}}{\varepsilon|\mathcal{S}'|}\sqrt{\log(1/\delta)\log(1/\beta)}\right)$. In the rest of the analysis we assume that this event occurs. Now, by definition of r, it holds that

$$\left|\mathrm{Avg}(S) - \mathrm{Avg}(\mathcal{S}')\right| \leq \frac{r|\mathcal{S} \setminus \mathcal{S}'|}{n} \leq \frac{8r}{\varepsilon n}\log\left(\frac{2\Lambda}{r_{\min}\beta}\right) + \frac{2r}{n}$$

Moreover, it holds that

$$\left|\hat{a} - \mathrm{Avg}(\mathcal{S}')\right| \leq O\left(\frac{\hat{r}}{\varepsilon|\mathcal{S}'|}\sqrt{\log(1/\delta)\log(1/\beta)}\right) \leq O\left(\frac{\max\{r, r_{\min}\}}{\varepsilon n}\sqrt{\log(1/\delta)\log(1/\beta)}\right),$$

where the second inequality holds since $\hat{r} \leq r + 4r_{\min}$ and $|\mathcal{S}'| \geq n/2$ by the assumption on n. The proof now follow by the above two inequalities.

For the running time analysis, step (1) takes $\tilde{O}(n)$ time (Proposition A.5). Step (2) that executes the Gaussian Mechanism, takes $\tilde{O}(n)$ time for computing the average, and $\tilde{O}(1)$ for sampling a number from a single one-dimensional. □

Proposition A.7 (Estimating the Average of Bounded Points in $\mathbb{R}^d$ (Restatement of Proposition 2.23))**.** *Let $\varepsilon \in (0,1)$, $d, \Lambda > 0$ and let $r_{\min} \in [0, \Lambda]$. There exists an efficient (ε, δ)-differentially private algorithm that takes an n-size database $\mathcal{S}$ of points inside the ball $B(\mathbf{0}, \Lambda)$ in $\mathbb{R}^d$ and satisfy the following utility guarantee: Let $r > 0$ be the minimal radius of a d-dimensional ball that contains all points in $\mathcal{S}$. Then with probability $1 - \beta$, the algorithm outputs $\hat{\mathbf{a}} \in \mathbb{R}^d$ such that*

$$\|\hat{\mathbf{a}} - \mathrm{Avg}(\mathcal{S})\| \leq O\left(\max\{r, r_{\min}\} \cdot \frac{d\sqrt{\log(1/\delta)}}{\varepsilon n}\left(\sqrt{\log(d/\delta)\log(d/\beta)} + \log\left(\frac{\Lambda d}{r_{\min}\beta}\right)\right)\right).$$

The algorithm runs in time $\tilde{O}(dn)$ (ignoring logarithmic factors).

Proof. The algorithm does the following: For each $i \in [d]$, let $\mathcal{S}_i = \{x_i \colon (x_1, \ldots, x_d) \in \mathcal{S}\}$ and compute an estimation $\hat{a}_i$ of $\mathrm{Avg}(\mathcal{S}_i)$ (in time $\tilde{O}(n)$) by applying Proposition A.6 with parameters $r_{\min}, \Lambda$, $\tilde{\varepsilon} = \frac{\varepsilon}{2\sqrt{2d\log(2/\delta)}}$, $\tilde{\delta} = \frac{\delta}{d}$, $\tilde{\beta} = \frac{\beta}{d}$. Finally, output $\hat{\mathbf{a}} = (\hat{a}_1, \ldots, \hat{a}_d)$. It is clear by advanced composition (Theorem 2.11) that the algorithm is (ε, δ)-differentially private. For the utility guarantee, note that with probability at least $1 - \beta$, for every $i \in [d]$ it holds that

$$\begin{aligned}
|\hat{a}_i - \mathrm{Avg}(\mathcal{S}_i)| &\leq O\left(\frac{r}{\tilde{\varepsilon}n}\left(\sqrt{\log(1/\tilde{\delta})\log(1/\tilde{\beta})} + \log\left(\frac{\Lambda}{r_{\min}\tilde{\beta}}\right)\right)\right) \\
&= O\left(\frac{r\sqrt{d\log(1/\delta)}}{\varepsilon n}\left(\sqrt{\log(d/\delta)\log(d/\beta)} + \log\left(\frac{\Lambda d}{r_{\min}\beta}\right)\right)\right),
\end{aligned}$$

and hence

$$\begin{aligned}
\|\hat{\mathbf{a}} - \mathrm{Avg}(\mathcal{S})\| &= \sqrt{\sum_{i=1}^{d}(\hat{a}_i - \mathrm{Avg}(\mathcal{S}_i))^2} \\
&\leq O\left(\frac{rd\sqrt{\log(1/\delta)}}{\varepsilon n}\left(\sqrt{\log(d/\delta)\log(d/\beta)} + \log\left(\frac{\Lambda d}{r_{\min}\beta}\right)\right)\right)
\end{aligned}$$

□

Remark A.8. *The above two algorithms guarantee differential-privacy whenever two neighboring databases have equal size. However, they can be easily extended to a more general case in which the privacy guarantee also holds in cases of addition and deletion of a point, by extending the Gaussian mechanism used in Proposition A.6 (see Remark 2.21) with essentially the same noise magnitude.*

B Missing Proofs

B.1 Proving Proposition 5.1

In this section we prove Proposition 5.1, restated below

Proposition B.1 (Restatement of Proposition 5.1)**.** *Let $\mathcal{P}$ be a multiset of n points in $\mathcal{B}(\mathbf{0},\Lambda)\subseteq\mathbb{R}^d$ and let $\mathcal{A}$ be an ω-approximation algorithm for k-means. Consider the following random execution: (1) Construct a multiset $\mathcal{S}$ of s i.i.d. samples from $\mathcal{P}$, (2) Compute $\tilde{C}=\mathcal{A}(\mathcal{S},k)$. Then for every $\beta>0$, with probability $1-\beta$ it holds that*

$$\mathrm{COST}_{\mathcal{P}}(\tilde{C})\leq\omega\cdot\mathrm{OPT}_k(\mathcal{P})+\xi(s,\beta),$$

where $\xi(s,\beta)=4\Big(M(s,\beta)+\sqrt{M(s,\beta)\cdot\omega\mathrm{OPT}_k(\mathcal{P})}\Big)$ for $M(s,\beta):=25\Lambda^2kd\log\Big(\frac{2nd}{\beta}\Big)\cdot\frac{n}{s}$.

In the following, fix values of s and β, let $\xi=\xi(s,\beta)$ and $M=M(s,\beta)$. The following event and claims are with respect to the random process in Proposition 5.1.

Claim B.2 (Event E [SSS20])**.** *Let E be the event that for every $C\in B(0,\Lambda)^k$, we have that*

$$\Big|\frac{n}{s}\cdot\mathrm{COST}_{\mathcal{S}}(C)-\mathrm{COST}_{\mathcal{P}}(C)\Big|\leq\sqrt{M\cdot\mathrm{COST}_{\mathcal{P}}(C)}:=\Delta(C)$$

Then it holds that $\Pr[E]\geq1-\beta$.

We next prove some useful facts that holds when event E occurs.

Claim B.3. *Conditioned on event E, it holds that*

$$\mathrm{COST}_{\mathcal{P}}(\tilde{C})\leq\omega\cdot\mathrm{OPT}_k(\mathcal{P})+\Delta(C^*_{\mathcal{P}})+\Delta(\tilde{C}),$$

letting $\tilde{C}$ be the set from Proposition 5.1, and letting $C^*_{\mathcal{P}}$ be the optimal k-means of $\mathcal{P}$.

Proof. Let $C^*_{\mathcal{S}}$ be the optimal k-means of $\mathcal{S}$. By the assumption on the algorithm $\mathcal{A}$, the set $\tilde{C}$ satisfies $\mathrm{COST}_{\mathcal{S}}(\tilde{C})\leq\omega\cdot\mathrm{OPT}_k(\mathcal{S})$. The proof follows by the following calculation

$$\begin{aligned}
\mathrm{COST}_{\mathcal{P}}(\tilde{C})&\leq\frac{n}{s}\cdot\mathrm{COST}_{\mathcal{S}}(\tilde{C})+\Delta(\tilde{C})\\
&\leq\omega\cdot\frac{n}{s}\cdot\mathrm{COST}_{\mathcal{S}}(C^*_{\mathcal{S}})+\Delta(\tilde{C})\\
&\leq\omega\cdot\frac{n}{s}\cdot\mathrm{COST}_{\mathcal{S}}(C^*_{\mathcal{P}})+\Delta(\tilde{C})\\
&\leq\omega\cdot\frac{n}{s}\cdot\Big(\frac{m}{n}\cdot\mathrm{COST}_{\mathcal{P}}(C^*_{\mathcal{P}})+\frac{s}{n}\cdot\Delta(C^*_{\mathcal{P}})\Big)+\Delta(\tilde{C})\\
&=\omega\cdot\mathrm{OPT}_k(\mathcal{P})+\Delta(C^*_{\mathcal{P}})+\Delta(\tilde{C}),
\end{aligned}$$

where the third inequality holds by event E, □

We now prove a corollary of Claim B.3.

Corollary B.4. *Conditioned on event E, it holds that*

$$\Delta(\tilde{C}) \leq 2\Big(M + \sqrt{M\omega\mathrm{OPT}_k(\mathcal{P})}\Big)$$

Proof. Let $x = \Delta(\tilde{C}) = \sqrt{M \cdot \mathrm{COST}_{\mathcal{P}}(\tilde{C})}$. By Claim B.3, it holds that

$$\frac{x^2}{M} - x \leq \omega \cdot \mathrm{OPT}_k(\mathcal{P}) + \sqrt{M \cdot \mathrm{OPT}_k(\mathcal{P})}.$$

Since $x \geq 0$, we conclude that

$$\begin{aligned} x &\leq \frac{1}{2} \cdot \left(M + \sqrt{M^2 + 4M\omega\mathrm{OPT}_k(\mathcal{P}) + 4M^{1.5}\sqrt{\mathrm{OPT}_k(\mathcal{P})}}\right) \\ &\leq M + \sqrt{M\omega\mathrm{OPT}_k(\mathcal{P})} + M^{0.75} \cdot \mathrm{OPT}_k(\mathcal{P})^{1/4} \qquad (13) \\ &\leq 2\Big(M + \sqrt{M\omega\mathrm{OPT}_k(\mathcal{P})}\Big), \end{aligned}$$

where the second inequality holds by the fact that $\sqrt{a+b} \leq \sqrt{a} + \sqrt{b}$ for $a, b \geq 0$, and the last inequality holds since the third term in (13) is either smaller than the first term, or smaller than the second one (recall that $M \geq 1$). □

The proof of Proposition 5.1 now immediately follows by Claim B.3 and Corollary B.4.

B.2 Proving Proposition 5.2

Proposition B.5 (Restatement of Proposition 5.2)**.** *Let $k \in \mathbb{N}$ and $\gamma \in [0, 1/8]$. Let $\mathcal{P} \in (\mathbb{R}^d)^*$, let $C = \{\mathbf{c}_1, \ldots, \mathbf{c}_k\}$ and $C' = \{\mathbf{c}'_1, \ldots, \mathbf{c}'_k\}$ be two k-tuples of centers in $\mathbb{R}^d$ such that for every $i \in [k]$ it holds that $\|\mathbf{c}'_i - \mathbf{c}_i\| \leq \gamma \cdot D_i$, where $D_i = \min_{j \neq i}\|\mathbf{c}_i - \mathbf{c}_j\|$. In addition, for every $i \in [k]$ let $\mathcal{P}_i$ be the multiset of all points in $\mathcal{P}$ that $\mathbf{c}'_i$ is closest to them in C'. Then*

$$\sum_{i=1}^{k} \mathrm{OPT}_1(\mathcal{P}_i) \leq (1 + 32\gamma)\mathrm{COST}_{\mathcal{P}}(C).$$

Proof. In the following, for $\mathbf{x} \in \mathcal{P}$ let $i_{\mathbf{x}} = \mathrm{argmin}_i\{\|\mathbf{x} - \mathbf{c}_i\|\}$ (i.e., the index of the closest center to $\mathbf{x}$ in C), and let $j_{\mathbf{x}} = \mathrm{argmin}_j\{\left\|\mathbf{x} - \mathbf{c}'_j\right\|\}$ (i.e., the index of the closest center to $\mathbf{x}$ in C'). It holds that

$$\begin{aligned} \sum_{i=1}^{k} \mathrm{OPT}_1(\mathcal{P}_i) &\leq \sum_{i=1}^{k} \sum_{\mathbf{x} \in \mathcal{P}_i} \|\mathbf{x} - \mathbf{c}_i\|^2 \\ &= \sum_{\mathbf{x} \in \mathcal{P}} \|\mathbf{x} - \mathbf{c}_{j_{\mathbf{x}}}\|^2 \\ &= \sum_{\mathbf{x} \in \mathcal{P}} \|\mathbf{x} - \mathbf{c}_{i_{\mathbf{x}}}\|^2 + \sum_{\mathbf{x} \in \mathcal{P}} \Big(\|\mathbf{x} - \mathbf{c}_{j_{\mathbf{x}}}\|^2 - \|\mathbf{x} - \mathbf{c}_{i_{\mathbf{x}}}\|^2\Big) \\ &= \mathrm{COST}_{\mathcal{P}}(C) + \sum_{\mathbf{x} \in \mathcal{P}} \Big(\|\mathbf{x} - \mathbf{c}_{j_{\mathbf{x}}}\|^2 - \|\mathbf{x} - \mathbf{c}_{i_{\mathbf{x}}}\|^2\Big) \end{aligned}$$

In the following, fix $\mathbf{x} \in \mathcal{P}$. We now bound

$$\|\mathbf{x} - \mathbf{c}_{j_\mathbf{x}}\|^2 - \|\mathbf{x} - \mathbf{c}_{i_\mathbf{x}}\|^2 = (\|\mathbf{x} - \mathbf{c}_{j_\mathbf{x}}\| - \|\mathbf{x} - \mathbf{c}_{i_\mathbf{x}}\|)(\|\mathbf{x} - \mathbf{c}_{j_\mathbf{x}}\| + \|\mathbf{x} - \mathbf{c}_{i_\mathbf{x}}\|)$$

First, since $\left\|\mathbf{x} - \mathbf{c}'_{j_\mathbf{x}}\right\| \leq \left\|\mathbf{x} - \mathbf{c}'_{i_\mathbf{x}}\right\|$ it holds that

$$\|\mathbf{x} - \mathbf{c}_{j_\mathbf{x}}\| \leq \left\|\mathbf{x} - \mathbf{c}'_{j_\mathbf{x}}\right\| + \left\|\mathbf{c}'_{j_\mathbf{x}} - \mathbf{c}_{j_\mathbf{x}}\right\| \leq \left\|\mathbf{x} - \mathbf{c}'_{i_\mathbf{x}}\right\| + \gamma\|\mathbf{c}_{i_\mathbf{x}} - \mathbf{c}_{j_\mathbf{x}}\| \tag{14}$$

Second,

$$\|\mathbf{x} - \mathbf{c}_{i_\mathbf{x}}\| \geq \left\|\mathbf{x} - \mathbf{c}'_{i_\mathbf{x}}\right\| - \left\|\mathbf{c}'_{i_\mathbf{x}} - \mathbf{c}_{i_\mathbf{x}}\right\| \geq \left\|\mathbf{x} - \mathbf{c}'_{i_\mathbf{x}}\right\| - \gamma\|\mathbf{c}_{i_\mathbf{x}} - \mathbf{c}_{j_\mathbf{x}}\|$$

Therefore

$$\|\mathbf{x} - \mathbf{c}_{j_\mathbf{x}}\| - \|\mathbf{x} - \mathbf{c}_{i_\mathbf{x}}\| \leq 2\gamma\|\mathbf{c}_{i_\mathbf{x}} - \mathbf{c}_{j_\mathbf{x}}\|$$

Now, $\|\mathbf{x} - \mathbf{c}_{i_\mathbf{x}}\| \leq \|\mathbf{x} - \mathbf{c}_{j_\mathbf{x}}\|$ and therefore

$$\begin{aligned}
\|\mathbf{x} - \mathbf{c}_{j_\mathbf{x}}\| + \|\mathbf{x} - \mathbf{c}_{i_\mathbf{x}}\| &\leq 2\|\mathbf{x} - \mathbf{c}_{j_\mathbf{x}}\| \\
&\leq 2\left\|\mathbf{x} - \mathbf{c}'_{i_\mathbf{x}}\right\| + 2\gamma\|\mathbf{c}_{i_\mathbf{x}} - \mathbf{c}_{j_\mathbf{x}}\| \\
&\leq 2\left(\|\mathbf{x} - \mathbf{c}_{i_\mathbf{x}}\| + \left\|\mathbf{c}'_{i_\mathbf{x}} - \mathbf{c}_{i_\mathbf{x}}\right\|\right) + 2\gamma\|\mathbf{c}_{i_\mathbf{x}} - \mathbf{c}_{j_\mathbf{x}}\| \\
&\leq 2\|\mathbf{x} - \mathbf{c}_{i_\mathbf{x}}\| + 4\gamma\|\mathbf{c}_{i_\mathbf{x}} - \mathbf{c}_{j_\mathbf{x}}\|,
\end{aligned}$$

where the second inequality holds by Equation (14).

We now like to bound $\|\mathbf{c}_{i_\mathbf{x}} - \mathbf{c}_{j_\mathbf{x}}\|$ as a function of $\|\mathbf{x} - \mathbf{c}_{i_\mathbf{x}}\|$. We first bound $\|\mathbf{c}_{i_\mathbf{x}} - \mathbf{c}_{j_\mathbf{x}}\|$ as a function of $\left\|\mathbf{x} - \mathbf{c}'_{i_\mathbf{x}}\right\|$.

$$\begin{aligned}
2\left\|\mathbf{x} - \mathbf{c}'_{i_\mathbf{x}}\right\| &\geq \left\|\mathbf{x} - \mathbf{c}'_{i_\mathbf{x}}\right\| + \left\|\mathbf{x} - \mathbf{c}'_{j_\mathbf{x}}\right\| \\
&\geq \left\|\mathbf{c}'_{i_\mathbf{x}} - \mathbf{c}'_{j_\mathbf{x}}\right\| \\
&\geq \|\mathbf{c}_{i_\mathbf{x}} - \mathbf{c}_{j_\mathbf{x}}\| - \left\|\mathbf{c}_{i_\mathbf{x}} - \mathbf{c}'_{i_\mathbf{x}}\right\| - \left\|\mathbf{c}_{j_\mathbf{x}} - \mathbf{c}'_{j_\mathbf{x}}\right\| \\
&\geq (1 - 2\gamma)\|\mathbf{c}_{i_\mathbf{x}} - \mathbf{c}_{j_\mathbf{x}}\|,
\end{aligned} \tag{15}$$

In addition

$$\left\|\mathbf{x} - \mathbf{c}'_{i_\mathbf{x}}\right\| \leq \|\mathbf{x} - \mathbf{c}_{i_\mathbf{x}}\| + \left\|\mathbf{c}_{i_\mathbf{x}} - \mathbf{c}'_{i_\mathbf{x}}\right\| \leq \|\mathbf{x} - \mathbf{c}_{i_\mathbf{x}}\| + \gamma\|\mathbf{c}_{i_\mathbf{x}} - \mathbf{c}_{j_\mathbf{x}}\|$$

Therefore,

$$2\|\mathbf{x} - \mathbf{c}_{i_\mathbf{x}}\| \geq (1 - 4\gamma)\|\mathbf{c}_{i_\mathbf{x}} - \mathbf{c}_{j_\mathbf{x}}\|$$

We have that

$$\begin{aligned}
\|\mathbf{x} - \mathbf{c}_{j_\mathbf{x}}\|^2 - \|\mathbf{x} - \mathbf{c}_{i_\mathbf{x}}\|^2 &= (\|\mathbf{x} - \mathbf{c}_{j_\mathbf{x}}\| - \|\mathbf{x} - \mathbf{c}_{i_\mathbf{x}}\|)(\|\mathbf{x} - \mathbf{c}_{j_\mathbf{x}}\| + \|\mathbf{x} - \mathbf{c}_{i_\mathbf{x}}\|) \\
&\leq (2\gamma\|\mathbf{c}_{i_\mathbf{x}} - \mathbf{c}_{j_\mathbf{x}}\|)(2\|\mathbf{x} - \mathbf{c}_{i_\mathbf{x}}\| + 4\gamma\|\mathbf{c}_{i_\mathbf{x}} - \mathbf{c}_{j_\mathbf{x}}\|) \\
&\leq \left(\frac{4\gamma}{1 - 4\gamma}\|\mathbf{x} - \mathbf{c}_{i_\mathbf{x}}\|\right)\left((2 + \frac{8\gamma}{1 - 4\gamma})\|\mathbf{x} - \mathbf{c}_{i_\mathbf{x}}\|\right) \\
&\leq 32\gamma\|\mathbf{x} - \mathbf{c}_{i_\mathbf{x}}\|^2,
\end{aligned}$$

where the least inequality holds since $\gamma \leq 1/8$. Now we can get the bound on the summation:

$$\sum_{\mathbf{x}\in\mathcal{P}}\left(\|\mathbf{x} - \mathbf{c}_{j_\mathbf{x}}\|^2 - \|\mathbf{x} - \mathbf{c}_{i_\mathbf{x}}\|^2\right) \leq \sum_{\mathbf{x}\in\mathcal{P}} 32\gamma\|\mathbf{x} - \mathbf{c}_{i_\mathbf{x}}\|^2 \leq 32\gamma\mathrm{COST}_\mathcal{P}(C)$$

□

B.3 Proving Theorem 6.11

In this section we prove the utility guarantee of PrivatekGaussians. We first by proving the following proposition that states the following: Assume that $\mathbf{X} \sim \mathcal{N}(\mu, \Sigma)$ with $\|\Sigma\| = \sigma^2$, and let $\mathbf{y}, \mathbf{z} \in \mathbb{R}^d$ such that (1) $\|\mathbf{y} - \mu\|$ is "large enough" (larger than $\Omega\left(\sigma\sqrt{\log(1/\beta)}\right)$) , and (2) $\|\mathbf{z} - \mu\|$ is "small enough". Then with probability $1 - \beta$ (over $\mathbf{X}$) it holds that $\|\mathbf{X} - \mathbf{z}\| < \|\mathbf{X} - \mathbf{y}\|$. Note that such an argument is trivial when $\|\mathbf{y} - \mu\|$ is at least $\Omega(\sigma\sqrt{d\log(1/\beta)})$, but using a standard projection argument, we can avoid the dependency in d.

Proposition B.6. *Let $\mathbf{X} \sim \mathcal{N}(\mathbf{0}, \Sigma)$ where $\|\Sigma\| = \sigma^2$, let $\mathbf{y} \in \mathbb{R}^d$ with $\|\mathbf{y}\| \geq 2(1+\gamma)\sqrt{2\log\left(\frac{1}{\beta}\right)} \cdot \sigma$ for some $\gamma > 0$, and let $\mathbf{z} \in \mathbb{R}^d$ with $\|\mathbf{z}\| \leq \frac{\gamma}{3(1+\gamma)}\|\mathbf{y}\|$. Then with probability $1 - \beta$ (over the choice of $\mathbf{X}$), it holds that $\|\mathbf{X} - \mathbf{z}\| < \|\mathbf{X} - \mathbf{y}\|$.*

Proof. Let $\mathbf{W} = \mathbf{z} + \frac{\langle \mathbf{X}-\mathbf{z}, \mathbf{y}-\mathbf{z}\rangle}{\|\mathbf{y}-\mathbf{z}\|^2}(\mathbf{y} - \mathbf{z})$ be the projection of $\mathbf{X}$ onto the line between $\mathbf{y}$ and $\mathbf{z}$. In the following we bound the probability that $\frac{\langle \mathbf{X}-\mathbf{z}, \mathbf{y}-\mathbf{z}\rangle}{\|\mathbf{y}-\mathbf{z}\|^2} < \frac{1}{2}$, which implies that $\|\mathbf{W} - \mathbf{z}\| < \|\mathbf{W} - \mathbf{y}\|$, and therefore, $\|\mathbf{X} - \mathbf{z}\| < \|\mathbf{X} - \mathbf{y}\|$. Note that $\langle \mathbf{X}, \mathbf{y} - \mathbf{z}\rangle$ is distributed according to the (one dimensional) Gaussian $\mathcal{N}(\mathbf{0}, (\mathbf{y} - \mathbf{z})^T\Sigma(\mathbf{y} - \mathbf{z}))$ and it holds that $(\mathbf{y} - \mathbf{z})^T\Sigma(\mathbf{y} - \mathbf{z}) \leq \sigma^2\|\mathbf{y} - \mathbf{z}\|$. Therefore, by Fact 6.6 we obtain that with probability $1 - \beta$ it holds that $\langle \mathbf{X}, \mathbf{y} - \mathbf{z}\rangle < \sigma\|\mathbf{y} - \mathbf{z}\|\sqrt{2\log(1/\beta)}$, and in the following we continue with the analysis assuming that this occurs. The proposition now follows by the following calculation.

$$\begin{aligned}
\frac{\langle \mathbf{X}-\mathbf{z}, \mathbf{y}-\mathbf{z}\rangle}{\|\mathbf{y}-\mathbf{z}\|^2} &= \frac{\langle \mathbf{X}, \mathbf{y}-\mathbf{z}\rangle - \langle \mathbf{z}, \mathbf{y}-\mathbf{z}\rangle}{\|\mathbf{y}-\mathbf{z}\|^2} \\
&< \frac{\sigma\|\mathbf{y}-\mathbf{z}\|\sqrt{2\log(1/\beta)} + \|\mathbf{z}\|\|\mathbf{y}-\mathbf{z}\|}{\|\mathbf{y}-\mathbf{z}\|^2} \\
&\leq \frac{\sigma\sqrt{2\log(1/\beta)}}{\left(1 - \frac{\gamma}{3(1+\gamma)}\right)\|y\|} + \frac{\frac{\gamma}{3(1+\gamma)}}{1 - \frac{\gamma}{3(1+\gamma)}} \\
&\leq \frac{1}{2(1+\gamma)\left(1 - \frac{\gamma}{3(1+\gamma)}\right)} + \frac{\frac{\gamma}{3(1+\gamma)}}{1 - \frac{\gamma}{3(1+\gamma)}} \\
&= \frac{1 + \frac{2\gamma}{3}}{2(1+\gamma)\frac{3+2\gamma}{3(1+\gamma)}} \\
&= \frac{1}{2},
\end{aligned}$$

where in the second inequality holds since $\|\mathbf{y} - \mathbf{z}\| \geq \|\mathbf{y}\| - \|\mathbf{z}\| \geq \left(1 - \frac{\gamma}{3(1+\gamma)}\right)\|y\|$, and the third inequality holds by the assumption on $\|\mathbf{y}\|$. □

In addition, we use the following fact.

Fact B.7. *Let $\mathcal{D} = \sum_{i=1}^k w_i\mathcal{D}_i$ be a mixture of the k distributions $\mathcal{D}_1, \ldots, \mathcal{D}_k$, and let $\mathcal{D}' = \sum_{i=1}^k w'_i\mathcal{D}'_i$ be a mixture of the k distributions $\mathcal{D}'_1, \ldots, \mathcal{D}'_k$. Assume that for every $i \in [k]$ it holds that $\mathrm{d_{TV}}(\mathcal{D}_i, \mathcal{D}'_i) \leq \frac{\alpha}{2}$ and $|w_i - w'_i| \leq \frac{\alpha}{k}$. Then $\mathrm{d_{TV}}(\mathcal{D}, \mathcal{D}) \leq \alpha$.*

We now ready to prove Theorem 6.11, stated for convenient below.

Theorem B.8 (Restatement of Theorem 6.11)**.** *Let $n, d, k, R, \sigma_{\max}, \sigma_{\min}, w_{\min}, \gamma > 0$, let $\alpha, \beta, \varepsilon, \delta \in (0,1)$, let $t = t(n, d, k, \beta, \gamma, \varepsilon, \delta, R, \sigma_{\max}, \sigma_{\min})$ be the value from Definition 6.9, and let $\mathcal{D} = \{(\mu_1, \Sigma_1, w_1), \ldots, (\mu_k, \Sigma_k, w_k)\}$ be an $(R, \sigma_{\max}, \sigma_{\min}, w_{\min})$-bounded $(1+\gamma)h$-separated mixture of k Gaussians in $\mathbb{R}^d$, for $h \geq 2\sqrt{2\log\left(\frac{8n}{\beta}\right)}$. In addition, let $\mathcal{A}$ be a (non-private) $\left(\lfloor \frac{n}{t} \rfloor, \frac{\beta}{8t}\right)$-labeling algorithm for $\mathcal{D}$ (Definition 6.5), and let $\mathcal{A}'$ be a private algorithm for learning a (single) bounded Gaussian with sample complexity υ (Definition 6.2). Assume that*

$$n \geq \max\left\{\frac{900t\left(d + 2\log\left(\frac{16kt}{\beta}\right)\right)}{\min\{450, \gamma^2 h^2\} \cdot w_{\min}} + t, \quad \frac{2\upsilon}{w_{\min}}, \quad \frac{4k^2}{\varepsilon\alpha} \cdot \log\left(\frac{8k}{\beta}\right)\right\}$$

where $\upsilon = \upsilon\left(d, \frac{\varepsilon}{2}, \frac{\delta}{2}, \frac{\alpha}{2}, \frac{\beta}{8k}, R, \sigma_{\max}, \sigma_{\min}\right)$. Then with probability $1-\beta$, when sampling a database $\mathcal{P}$ of $2n$ i.i.d. samples from $\mathcal{D}$, Algorithm PrivatekGaussians *on inputs $\mathcal{P}, k, \alpha, \beta, \varepsilon, \delta, \gamma, R, \sigma_{\max}, \sigma_{\min}, \mathcal{A}, \mathcal{A}'$ outputs $\hat{\mathcal{D}}$ such that $\mathrm{d_{TV}}(\mathcal{D}, \hat{\mathcal{D}}) \leq \alpha$.*

Proof. Let $E_1 = \bigwedge_{j\in[t], i\in[k]} E_1^{t,i}$ where $E_1^{j,i}$ is the event that the a-size set $\mathcal{S}_j$ in Step 1a of CollectEmpiricalMeans contains at least $\frac{w_i s}{2}$ samples from the i'th Gaussian. Note that for every $j \in [t]$ and $i \in [k]$, it holds that

$$\begin{aligned}\Pr\left[E_1^{j,i}\right] &= \Pr\left[\mathrm{Bin}(s, w_i) \geq \frac{sw_i}{2}\right] \\ &\geq 1 - \Pr\left[\mathrm{Bin}(s, w_{\min}) < \frac{sw_{\min}}{2}\right] \\ &\geq 1 - e^{-\frac{w_{\min} s}{4}}\end{aligned}$$

where the last inequality holds by Fact 2.27. Therefore, we obtain that $\Pr\left[E_1^{j,i}\right] \geq 1 - \frac{\beta}{8kt}$ whenever $s \geq \frac{4}{w_{\min}} \log\left(\frac{8kt}{\beta}\right)$. In particular, since $s = \lfloor \frac{n}{t} \rfloor$, the above holds whenever $n \geq \frac{4t}{w_{\min}} \log\left(\frac{8kt}{\beta}\right) + t$. Therefore, by the assumption on n and the union bound, we obtain that

$$\Pr[E_1] \geq 1 - \frac{\beta}{8} \tag{16}$$

In the following, assume that event E_1 occurs. For $j \in [t]$ and $i \in [k]$ let $\hat{\mathcal{S}}_j^i$ be all the points in $\mathcal{S}_j$ that have been drawn from the i'th Gaussian $\mathcal{N}(\mu_i, \Sigma_i)$, and let $\hat{\mu}_{j,i} = \mathrm{Avg}\left(\hat{\mathcal{S}}_j^i\right)$. Let $E_2 = \bigwedge_{j\in[t], i\in[k]} E_2^{j,i}$, where $E_2^{j,i}$ is the event that $\|\hat{\mu}_{j,i} - \mu_i\| \leq \frac{\gamma h}{16} \cdot \sigma_i$. Since $\hat{\mu}_{j,i}$ is the average of at least $\frac{w_i s}{2}$ samples from the Gaussian $\mathcal{N}(\mu_i, \Sigma_i)$, we obtain by Fact 6.8 that with probability $1 - \frac{\beta}{8kt}$ it holds that

$$\|\hat{\mu}_{j,i} - \mu_i\| \leq \frac{\sqrt{2d} + 2\sqrt{\log\left(\frac{8kt}{\beta}\right)}}{\sqrt{w_i s}} \cdot \sigma_i \leq \frac{\gamma h}{16} \cdot \sigma_i, \tag{17}$$

where the last inequality holds whenever $s \geq \frac{900\left(d+2\log\left(\frac{8kt}{\beta}\right)\right)}{w_i\gamma^2h^2}$. Since $s = \left\lfloor \frac{n}{t} \right\rfloor$, we obtain that Equation (17) holds whenever

$$n \geq \frac{900t\left(d+2\log\left(\frac{8kt}{\beta}\right)\right)}{\gamma^2h^2w_{\min}} + t$$

which holds by the assumption on n. Therefore, event $E_2^{j,i}$ occurs with probability at least $1-\frac{\beta}{8kt}$, and we conclude by the union bound that

$$\Pr[E_2 \mid E_1] \geq 1 - \frac{\beta}{8} \tag{18}$$

Let $E_3 = \bigwedge_{j=1}^{t} E_3^j$, where E_3^j is the event that the resulting labeling function L_j in Step 1b of the j'th iteration in CollectEmpiricalMeans satisfies:

$$\forall \mathbf{x}, \mathbf{x}' \in \mathcal{S}_j : \quad \mathbf{x}, \mathbf{x}' \text{ were drawn from the same Gaussian} \iff L_j(\mathbf{x}) = L_j(\mathbf{x}').$$

Since $\mathcal{A}$ is a $(s = \left\lfloor \frac{n}{t} \right\rfloor, \frac{\beta}{8t})$-labeling algorithm for $\mathcal{D}$, it holds that $\Pr\left[E_3^j\right] \geq 1 - \frac{\beta}{8t}$ for every $j \in [t]$, and we deduce by the union bound that

$$\Pr[E_3] \geq 1 - \frac{\beta}{8} \tag{19}$$

In the rest of the analysis we assume that event $E_1 \wedge E_2 \wedge E_3$ occurs. This means that for every $j \in [t]$ there exists a permutation π_j over $[k]$ such that for each $i \in [k]$, the set of all points in $\mathcal{S}_j$ that have been drawn from the i'th Gaussian (which we denoted by $\mathcal{S}_j^i$) equals to $\{\mathbf{x} \in \mathcal{S}_j \colon L_j(\mathbf{x}) = \pi_j(i)\}$, and assume without loss of generality that for all $j \in [t]$, π_j is the identity (i.e., $\pi_j(i) = i$). Therefore, for all $j \in [t]$ and $i \in [k]$ it holds that $\hat{\mu}_{j,i} = \bar{\mu}_{j,i}$, where $\bar{\mu}_{j,i}$ is the empirical mean from Step 1c. Namely, we obtained that

$$\forall j \in [t], i \in [k] : \quad \|\bar{\mu}_{j,i} - \mu_i\| \leq \frac{\gamma h}{16} \cdot \sigma_i, \tag{20}$$

and in particular, it holds that

$$\forall j \in [t], i \in [k] : \quad \|\bar{\mu}_{j,i}\| \leq \|\mu_{j,i}\| + \frac{\gamma h}{16} \cdot \sigma_i \leq \Lambda \tag{21}$$

Therefore, we deduce that $\mathcal{T}$ from Step 7 of $\widetilde{\text{PrivatekGaussians}}$ is contained in $(B(\mathbf{0},\Lambda)^k)^*$, and is partitioned by the Δ-far balls $\mathcal{B} = \{B_i(\mu_i, r_i = \frac{\gamma h}{16} \cdot \sigma_i)\}_{i=1}^k$ (Definition 3.2) for $\Delta = 16$, where Partition($\mathcal{T}$) is exactly $\{\mathcal{P}_1 = \{\bar{\mu}_{j,1}\}_{j=1}^t, \ldots, \mathcal{P}_k = \{\bar{\mu}_{j,k}\}_{j=1}^t\}$ (note that the balls are indeed Δ-far by the separation assumption that $\|\mu_i - \mu_j\| \geq (1+\gamma)h\max\{\sigma_i, \sigma_j\}$). Therefore, by the utility guarantee of PrivatekAverages (Theorem 4.12) we obtain that with probability $1 - \frac{\beta}{8}$:

$$
\begin{aligned}
&\forall i \in [k]: \quad \|\hat{\mathbf{a}}_i - \mathrm{Avg}(\mathcal{P}_i)\| \\
&\leq \max\{r_i, \tilde{r}_{\min}\} \cdot \frac{\lambda dk\ell\sqrt{\log\left(\frac{k\ell}{\tilde{\delta}}\right)}}{\tilde{\varepsilon}\tilde{n}}\left(\sqrt{\log\left(\frac{dk\ell}{\tilde{\delta}}\right)\log\left(\frac{dk\ell}{\tilde{\beta}}\right)} + \log\left(\frac{\Lambda dk}{\tilde{r}_{\min}\tilde{\delta}}\right)\right) \\
&\leq \frac{\gamma h}{16} \cdot \sigma_i \cdot \frac{\lambda dk\ell\sqrt{\log\left(\frac{k\ell}{\delta}\right)}}{\varepsilon t}\left(\sqrt{\log\left(\frac{dk\ell}{\delta}\right)\log\left(\frac{4dk\ell}{\beta}\right)} + \log\left(\frac{dk(16R + \gamma h\sigma_{\max})}{\gamma\delta h\sigma_{\min}}\right)\right) \\
&\leq \frac{\gamma h}{16} \cdot \sigma_i, \qquad (22)
\end{aligned}
$$

where last inequality holds by the assumption on t (Definition 6.9). In the following, we denote by E_4 the event that Equation (22) occurs, where recall that we proved that

$$\Pr[E_4 \mid E_1 \wedge E_2 \wedge E_3] \geq 1 - \frac{\beta}{8} \qquad (23)$$

In the following, we also assume that event E_4 occurs. Recall that by Equation (20), for each $j \in [t]$ and $i \in [k]$ it holds that

$$\|\mathrm{Avg}(\mathcal{P}_i) - \mu_i\| \leq \frac{1}{t}\|\bar{\mu}_{j,i} - \mu_i\| \leq \frac{\gamma h}{16} \cdot \sigma_i, \qquad (24)$$

and we deduce by Equations (22) and (24) that for all $i \in [k]$ it holds that

$$\|\hat{\mathbf{a}}_i - \mu_i\| \leq \frac{2\gamma h}{16} \cdot \sigma_i. \qquad (25)$$

Therefore, for all $i \neq j$ it holds that

$$
\begin{aligned}
\|\hat{\mathbf{a}}_j - \mu_i\| &\geq \|\mu_i - \mu_j\| - \|\hat{\mathbf{a}}_j - \mu_j\| \qquad (26)\\
&\geq \left((1+\gamma) - \frac{2\gamma}{16}\right) \cdot h \cdot \max\{\sigma_i, \sigma_j\} \\
&= \left(1 + \frac{13\gamma}{16}\right) \cdot h \cdot \max\{\sigma_i, \sigma_j\}
\end{aligned}
$$

where the last inequality holds by the separation assumption along with Equation (24). Hence, we obtain that for each $i \neq j$ it holds that

$$\|\hat{\mathbf{a}}_i - \mu_i\| \leq \frac{\frac{2\gamma h}{16}}{\frac{16+13\gamma}{16} \cdot h} \cdot \|\hat{\mathbf{a}}_j - \mu_i\| \leq \frac{\gamma'}{3(1+\gamma')}\|\hat{\mathbf{a}}_j - \mu_i\| \qquad (27)$$

where $\gamma' = \frac{13\gamma}{16}$ (the last inequality holds for every $\gamma > 0$). Since $h \geq 2\sqrt{2\log\left(\frac{1}{\beta'}\right)}$ for $\beta' = \frac{\beta}{8n}$, then by Proposition B.6 along with Equations (26) and (27), for every $i \neq j$, when sampling a point $\mathbf{x}$ from the i'th Gaussian $\mathcal{N}(\mu_i, \sigma_i)$, then with probability $1 - \frac{\beta}{8n}$ it holds that $\|\mathbf{x} - \hat{\mathbf{a}}_i\| < \|\mathbf{x} - \hat{\mathbf{a}}_j\|$. Therefore, let E_5 be the event that for all $i \in [k]$ and all $\mathbf{x} \in \mathcal{P}''$ that have been sampled from the

i'th Gaussian $\mathcal{N}(\mu_i, \Sigma_i)$, it holds that $\hat{\mathbf{a}}_i$ is the closest point to each of them among $\{\hat{\mathbf{a}}_1, \ldots, \hat{\mathbf{a}}_k\}$. Then by the union bound it holds that

$$\Pr[E_5 \mid E_1 \wedge E_2 \wedge E_3 \wedge E_4] \geq 1 - \frac{\beta}{8} \tag{28}$$

In the following we also assume that event E_5 occurs. Let $E_6 = \bigwedge_{i\in[k]} E_6^i$ where E_6^i is the event that $\mathcal{P}''$ contains at least $\frac{w_i n}{2}$ samples from the i'th Gaussian (namely, $|\mathcal{P}_i''| \geq \frac{w_i n}{4}$). Similar calculation to bounding $\Pr[E_1]$, it holds that

$$\Pr[E_6 \mid E_1 \wedge \ldots \wedge E_5] \geq 1 - \frac{\beta}{8} \tag{29}$$

provided that $n \geq \frac{4}{w_{\min}} \log\left(\frac{8k}{\beta}\right)$, which holds by the assumption on n.

In the following we assume that event E_6 occurs, and let $E_7 = \wedge_{i=1}^k E_7^i$, where E_7^i is the event that the output $(\hat{\mu}_i, \hat{\Sigma}_i)$ of the private algorithm $\mathcal{A}'$ in Step 8b of the i'th iteration satisfies $\mathrm{d_{TV}}(\mathcal{N}(\mu_i, \Sigma_i), \mathcal{N}(\hat{\mu}_i, \hat{\Sigma}_i)) \leq \frac{\alpha}{2}$. By the assumption on algorithm $\mathcal{A}'$, we obtain that $\Pr\left[E_7^i\right] \geq 1 - \frac{\beta}{8k}$ whenever $|\mathcal{P}_i''| \geq s$, which holds when $n \geq \frac{2s}{w_i}$. Therefore, since $n \geq \frac{2s}{w_{\min}}$ by assumption, we obtain by the union bound that

$$\Pr[E_7 \mid E_1 \wedge \ldots \wedge E_6] \geq 1 - \frac{\beta}{8}. \tag{30}$$

In the following, for $i \in [k]$ let L_i be the value of the Laplace noise in Step 8c of the i'th iteration, let E_8^i be the event that $|L_i| \leq \frac{2}{\varepsilon} \log\left(\frac{16k}{\beta}\right)$, and let $E_8 = \wedge_{i=1}^k E_8^i$. By Fact 2.13, for any fixing of $i \in [k]$ it holds that $\Pr\left[E_8^i\right] \geq 1 - \frac{\beta}{8k}$, and therefore, by the union bound it holds that

$$\Pr[E_8] \geq 1 - \frac{\beta}{8}. \tag{31}$$

In the following we also assume that E_8 occurs. It is left to show that when event $E_1 \wedge \ldots \wedge E_8$ occurs, for every $i \in [k]$ it holds that

$$\forall i \in [k]: \quad |\hat{w}_i - w_i| \leq \frac{\alpha}{k}. \tag{32}$$

Indeed, given Equation (32) and event E_7, we deduce by Fact B.7 that $\mathrm{d_{TV}}(\mathcal{D}, \hat{\mathcal{D}}) \leq \alpha$, which holds with probability at least $\Pr[E_1 \wedge \ldots \wedge E_8] \geq 1 - \beta$ (holds by Equation (16) to Equation (31)).

We now prove that Equation (32) holds when $E_1 \wedge \ldots \wedge E_8$ occurs. Fix $i \in [k]$, let $L = \sum_{j=1}^k L_j$, and compute

$$\begin{aligned}
|\hat{w}_i - w_i| &= \left|\frac{\hat{n}_i}{\hat{n}} - \frac{n_i}{n}\right| = \left|\frac{n_i + L_i}{n + L} - \frac{n_i}{n}\right| \\
&= \left|\frac{nL_i - n_i L}{n(n+L)}\right| = \left|\frac{(n - n_i)L_i - n_i \sum_{j\neq i} L_j}{n(n+L)}\right| \\
&\leq \frac{\frac{2k}{\varepsilon}\log\left(\frac{8k}{\beta}\right)}{n - \frac{2k}{\varepsilon}\log\left(\frac{8k}{\beta}\right)} \\
&\leq \frac{\alpha}{k},
\end{aligned}$$

where the first inequality holds by event E_8, and the last one holds whenever $n \geq \frac{4k^2}{\varepsilon\alpha} \cdot \log\left(\frac{8k}{\beta}\right)$, which holds by the assumption on n.

□