Locally Private k-Means Clustering with Constant Multiplicative Approximation and Near-Optimal Additive Error (Full version)

Anamay Chaturvedi* Matthew Jones† Huy L. Nguyễn‡

Abstract

Given a data set of size n in d'-dimensional Euclidean space, the k-means problem asks for a set of k points (called centers) so that the sum of the ℓ_2^2-distances between points of a given data set of size n and the set of k centers is minimized. Recent work on this problem in the locally private setting achieves constant multiplicative approximation with additive error $\tilde{O}(n^{1/2+a} \cdot k \cdot \max\{\sqrt{d}, \sqrt{k}\})$ and proves a lower bound of $\Omega(\sqrt{n})$ on the additive error for any solution with a constant number of rounds. In this work we bridge the gap between the exponents of n in the upper and lower bounds on the additive error with two new algorithms. Given any $\alpha > 0$, our first algorithm achieves a multiplicative approximation guarantee which is at most a $(1+\alpha)$ factor greater than that of any non-private k-means clustering algorithm with $k^{\tilde{O}(1/\alpha^2)}\sqrt{d'n}\,\mathrm{poly}\log n$ additive error. Given any $c > \sqrt{2}$, our second algorithm achieves $O(k^{1+\tilde{O}(1/(2c^2-1))}\sqrt{d'n}\,\mathrm{poly}\log n)$ additive error with constant multiplicative approximation. Both algorithms go beyond the $\Omega(n^{1/2+a})$ factor that occurs in the additive error for arbitrarily small parameters a in previous work, and the second algorithm in particular shows for the first time that it is possible to solve the locally private k-means problem in a constant number of rounds with constant factor multiplicative approximation and polynomial dependence on k in the additive error arbitrarily close to linear.

1 Introduction

Given n points in a d-dimensional Euclidean space, the k-means clustering problem asks for a set of k points S such that the sum of ℓ_2^2-distances from each data point to the closest respective point in S is minimized. Although k-means clustering in the non-private setting is well-studied, over the past few years there have been several developments in the differentially private (DP) setting. Differential privacy [Dwork et al., 2006] provides a framework to characterize the loss in privacy which occurs when sensitive data is processed and the output of this computation is revealed publicly. Although there are different ways to define and capture this loss in privacy, broadly speaking these characterizations tend to be either central or local in nature.

Informally, differential privacy asks for a guarantee that the likelihood of any possible output does not change too much by adding to or dropping from our data set any possible private value from the data universe. In any private algorithm such a guarantee is fulfilled by adding carefully calibrated noise to quantities that are information-theoretically sensitive to the private data in the course of the computation, and under the constraints of being private the goal is to achieve relatively low error. Perfect answers to the algorithmic problem at hand typically violate privacy; as a consequence, the constraints of privacy usually enforce harsher lower bounds on accuracy or utility than those imposed by the limits of time or sample efficient computation.

In local differentially privacy (LDP) the constraints are even more severe; the entity solving the algorithmic problem only gets access to noisy, privatized data. This constraint forces even stronger lower bounds on the

*Khoury College of Computer Sciences, Northeastern University ..... chaturvedi.a@northeastern.edu
†Khoury College of Computer Sciences, Northeastern University ..... jones.m@northeastern.edu
‡Khoury College of Computer Sciences, Northeastern University ..... hu.nguyen@northeastern.edu

Table 1: Comparison with recent LDP algorithms for k-means

Work	Multiplicative Approximation	Additive Error
Nissim and Stemmer [2018]	$O(k)$	$\tilde{O}(n^{2/3+a} \cdot d'^{1/3} \cdot \sqrt{k})$
Kaplan and Stemmer [2018]	$O(1)$	$\tilde{O}(n^{2/3+a} \cdot d'^{1/3} \cdot k^2)$
Stemmer [2020]	$O(1)$	$\tilde{O}(n^{1/2+a} \cdot k \cdot \max\{\sqrt{d'}, \sqrt{k}\})$
This work, algorithm 1	$(1+\alpha)\eta$	$\tilde{O}(n^{1/2} \cdot d'^{1/2} \cdot k^{\tilde{O}(1/\alpha^2)})$
This work, algorithm 4	$O(c^2)$	$\tilde{O}(n^{1/2} \cdot d'^{1/2} \cdot k^{1+O(1/(2c^2-1))})$

The additive error assumes a data set of size n inside a ball with unit radius. The $\tilde{O}$ notation hides dependence on the privacy parameters, the failure probability, and log terms. The user-defined parameter c can take any real value greater than $\sqrt{2}$.

accuracy of locally private protocols; for the k-means clustering problem a lower bound of $\Omega(\sqrt{n})$ is known for the additive error of any interactive constant factor multiplicative approximation algorithm [Stemmer, 2020].

Recent work on LDP k-means The first LDP algorithm for the k-means problem with provable guarantees was given by Nissim and Stemmer [2018] wherein they achieved a multiplicative approximation of $O(k)$ and and an additive error term of $\tilde{O}(n^{2/3+a} \cdot d^{1/3} \cdot \sqrt{k})$. They achieved this result by solving the related 1-cluster problem that asks the solver to privately allocate a small number of centers so that some center in that set covers all data points within a ball of minimal radius; by an observation of Feldman et al. [2017], there is a general algorithm that given access to a private solution for the 1-cluster problem solves the private k-means problem. The exponent of n in the additive error term holds for arbitrarily small a at the cost of looser multiplicative approximation guarantees; this artefact is the consequence of using *locality sensitive hashing* (LSH), something which appears in most later work as well.

Kaplan and Stemmer [2018] gave the first constant factor multiplicative approximation algorithm for this problem within an additive error of $\tilde{O}(n^{2/3+a} \cdot d^{1/3} \cdot k^2)$. They refine the approach of the previous work by specifically targeting the k-means problem but also use LSH functions to detect the accumulation of data. The additive error was further brought down by Stemmer [2020], who achieved an additive error of $\tilde{O}(n^{1/2+a} \cdot k \cdot \max\{\sqrt{d}, \sqrt{k}\}$ and also proved a lower bound of $\Omega(\sqrt{n})$, as mentioned before. Given that all previous works exhibit some trade-off between the exponent of n and the multiplicative approximation, the exponents of $1/2 + a$ and $1/2$ in the upper and lower bounds of Stemmer [2020] is particularly provocative. It naturally leads to the question

Does there exist an LDP k-means clustering algorithm with constant multiplicative approximation and additive error with a $\sqrt{n}$ dependence on the size of the data set?

In the non-private setting it has been seen that the performance of k-means clustering algorithms is usually not very sensitive to the multiplicative approximation guarantee, unless the data set is chosen in a pathological fashion. Experimental work [Balcan et al., 2017, Chaturvedi et al., 2020] on k-means clustering in the related central model of DP shows that the performance of private clustering algorithms seems to be far more sensitive to the additive error, which as we have observed is bound to exist due to the constraints of being private. This highlights the importance of the question of determining the true dependence of the additive error term on the size of the data set.

Technical contributions: In this work we present two algorithms for the k-means problem in the LDP setting, wherein we go beyond the $n^{1/2+a}$ barrier demonstrating that the trade-off can be independent of n for some regimes of k and n. Our first algorithm is a one-round protocol that achieves a $(1+\alpha)$-multiplicative approximation to the cost guarantee of any non-private clustering algorithm that it is given acess to as a subroutine. It achieves additive error $k^{\tilde{O}(1/\alpha^2)}\sqrt{d'n}\,\mathrm{poly}\log n$; we see that the trade-off between the additive and multiplicative approximations in this algorithm has been shifted from n to k. However, the $\tilde{O}$ term in

the exponent of k can hide large constants, which is an undesirable property in a setting where low additive error seems to dictate performance.

Theorem 1.1. *Algorithm 1 is an (ϵ, δ)-locally differentially private algorithm that after one round of interaction with a private distributed data set $D' \subset \mathbb{R}^{d'}$ of size n, outputs a set S' of size k such that for failure probability polynomially small in n,*

$$f_{D'}(S') \leq (1 + O(\alpha))\eta \operatorname{OPT}' + \frac{1}{\epsilon} k^{\tilde{O}(1/\alpha^2)} \sqrt{d' n \log 1/\delta} \operatorname{poly} \log n.$$

We address this deficit with our second algorithm, where we return to an LSH-based approach and drive down the exponent of k to $1 + O(1/(2c^2 - 1))$. Again as this exponent approaches 1 the multiplicative approximation factor blows up but this shows for the first time that it is possible to have constant factor multiplicative approximation k-means clustering algorithms in the LDP setting with additive error that has a truly square-root dependence on the data set size and the ambient dimension and arbitrarily close to linear dependence on the number of cluster centers.

Theorem 1.2. *Algorithm 4 is an (ϵ, δ)-locally differentially private algorithm such that given $c > \sqrt{2}$, after four rounds of interaction with a private distributed data set $D' \subset \mathbb{R}^{d'}$ of size n outputs a set S' of size k such that with probability $1 - \beta$,*

$$f_{D'}(S') = O(\operatorname{OPT}') + O\left(\frac{1}{\epsilon}\sqrt{d' n \ln(n/\delta)}\right)\left(\frac{k \operatorname{poly} \log n}{\beta}\right)^{1+O(1/(2c^2-1))}.$$

It was observed in Stemmer [2020] that one of the main road-blocks in computing solutions with low additive error is figuring out how to generate a relatively small bi-criteria solution to the k-means problem as a first step. A bi-criteria solution relaxes two constraints of the k-means problem; we permit picking more than k centers, and we relax the minimum cost requirement to a multiplicative approximation guarantee. Any such bi-criteria solution can be exploited to construct a proxy data set on which we can apply any non-private k-means clustering algorithm. The fact that the clustering cost of the original data set with respect to the candidate centers used to generate the proxy data set can be exploited to show that k-means solutions for proxy data sets work well for the original data set as well. In order to avoid an exponent of $1/2 + a$ on n, it is necessary to find a bi-criteria solution with $O(\operatorname{poly} k \operatorname{poly} \log n)$ many candidate centers such that the additive error to their respective multiplicative approximations is at most $O(\operatorname{poly} k\sqrt{n} \operatorname{poly} \log n)$ (omitting the dependence on dimension). Both our algorithms achieve their improvements by generating such a small size bi-criteria solutions for the k-means problem.

LDP k-means with arbitrarily tight multiplicative approximation: In our first algorithm, we appeal to recent advances in dimension reduction for k-means clustering Makarychev et al. [2019] which show that Johnson-Lindenstrauss style dimension reduction to $\tilde{O}(\log k/\alpha^2)$ preserves the cost of every k-clustering of a data set within a multiplicative approximation of $(1 \pm \alpha)$. Suppose we decompose the domain in concentric shells depending on their distance from some fixed k cluster centers. By setting geometric thresholds of $1, 1/2, 1/4$ units and so on, the lth ring has the property that every data point in that shell has a clustering cost of $O(1/(2^l)^2)$ units. To cluster the lth ring, we allocate a number of centers by appealing to a grid-based approach following Chaturvedi et al. [2020]; since we were able to reduce dimensions to $\tilde{O}(\log k/\alpha^2)$ we are able to show that allocating $k^{\tilde{O}(1/\alpha^2)} \operatorname{poly} \log n$ centers suffices to ensure that most points in the lth shell are within an $O(\alpha/(2^l)^2)$ distance of some candidate center.

Extending this for every shell with appropriately scaled grids we get the promise that moving each data point to its closest candidate center would lead to net movement of $O(\alpha \operatorname{OPT})$ where OPT is the optimal clustering cost. The rest of the argument follows essentially by applications of the triangle inequality to prove that the dimension reduced and proxy data sets have similar costs for any candidate k-means solutions.

LDP k-means with low additive error: We note that the constant absorbed by the $\tilde{O}$ term in the exponent of k of our first algorithm could be large, which is an undesirable property in a setting where low additive error seems to dictate performance. We address this deficit with our second algorithm, where we return to an LSH-based approach and drive down the exponent of k to $1 + O(1/(2c^2 - 1))$. This shows for the first time that it is possible to have constant factor multiplicative approximation k-means clustering algorithms in the LDP setting with additive error that has a square-root dependence on the data set size and the ambient dimension (up to log factors) and arbitrarily close to linear dependence on the number of cluster centers.

We achieve this improvement by appealing to a construction of Braverman et al. [2017] who impose a randomly-shifted hierarchy of dyadic cells in a dimension reduced space. A tree structure is defined on subsets of the domain $[0, 1)^d$; starting with $[0, 1)^d$ as the root node, we bisect the hypercube along each axis to generate 2^d congruent octants. Each octant is itself a hypercube that we designate a child of the original cell, on proceeding recursively for $\log n$ levels the side length of each cell in the lowest level is $< 1/n$.

The crucial observation made by Braverman et al. [2017] was that after a uniformly random shift of the tiling there are $O(1)$ cells with side-length t units within a distance of t/d units of any point. By applying this observation to an optimal k-means solution, we are able to identify a small number of cells where the data accumulates per level. These cells serve as our domains for LSH functions. The number of data points that accumulate in these cells scales inversely with the side-length of the cells; this ensures that we only allocate centers when such an allocation is certain to be helpful. We are able to allocate a far smaller number of centers to generate our bi-criteria solution than in our first algorithm. Moving the $1/2 + a$-style exponent from n to k is technically involved and we give a more detailed explanation in section 4.

Challenges of the local setting: We recall that in the locally private setting, each agent must add noise to any response they give under the assumption that it is public knowledge that all data lies in a domain of diameter 1. This will require adding a noise vector with length proportional to $1/\epsilon$ to their private data if they were to ϵ-privately release their point directly. The implications of the large noise needed to obfuscate information means that it is impossible to privately derive fine-grained information about where individual points lie.

It follows from these considerations that we must try and get aggregate information about the geometry of the data set indirectly. One way of accomplishing this is to *discretize* the agents' response. Although again the privatized individual responses are highly noisy, since the range of values taken by this discretized response is finite the slight bias towards values which are *heavy-hitters* causes their counts to accumulate and be distinguishable from the counts of false positives. We will appeal to prior work on locally private succinct histogram recovery to recover such heavy hitting values with minimal loss in privacy.

From this perspective, we see that in the first algorithm we achieve our discretization by dividing our space using proximity to grid points, and in the second algorithm we use a two different kinds of discretization; a cell based discretization which is philosophically similar to that of the first, and an LSH-based discretization which gives a geometrically meaningful response not in terms of the ambient space but instead in terms of the rest of the data set.

Reducing round-complexity via HeavySumsOracle: In the course of our algorithms we often encounter a situation where we first identify some subset of the data domain that is advantageous for us to allocate a candidate center in and then we need to compute a vector average over points in that domain. Although naively performing such a computation would require two rounds in the LDP setting, we construct a subroutine that can be run in parallel with the succinct histograms used to identify such regions of the data domain, and can be queried to estimate the vector sums of all points mapping to such domains. Dividing these sums by the histogram counts yields the averages we need. Indeed, our construction is in fact a bit more general, and allows one to recover sums of arbitrary private vector values for all points that map to some heavy hitting value under a completely different value mapping. This construction allows us to compute vector averages over points mapping to heavy LSH buckets as well as vector averages in the original space over all points that map to a certain cluster in the dimension-reduced space; the two value mappings

need not have anything to do with each other.

Concurrent work: In Chang et al. [2021] a one-round protocol for LDP k-means with similar cost guarantees as algorithm 1 is introduced, also surpassing the $n^{1/2+a}$ barrier mentioned above. They operate in the ϵ-DP setting and get a multiplicative approximation of $\eta(1+\alpha)$ where η is the multiplicative approximation guarantee of any given non-private k-means algorithm and an additive error term of $k^{O_\alpha(1)} \cdot \sqrt{nd'} \cdot \text{poly}\log(n)/\epsilon$. They also demonstrate that their protocol can be extended to the shuffle model [Bittau et al., 2017, Cheu et al., 2019, Erlingsson et al., 2019] of differential privacy.

Outline of paper: In section 2 we start by formalizing the problem statement and the definition of LDP that our algorithms must fulfill. We then summarize some notation that eases the description of our analysis and recall private subroutines from previous work. We also introduce the HeavySumsOracle, a one-round protocol that can be run in parallel with a private succinct histogram and privately constructs a data structure that may be queries to recover sums of vector-function values taken by all agents that happen to map to a heavy-hitting value in the succinct histogram. We recall the LSH function definition and prove some fundamental properties of the construction we use in section 4.

In section 3 we introduce our LDP k-means algorithm for arbitrarily tight multiplicative approximation, algorithm 1. We start by establishing the pseudo-code and outlining the main steps, and then give a technical discussion explaining some of the algorithmic choices made as well as sketching why the cost analysis works out. We then give a formal proof of the cost and privacy guarantees. The main result of this section is theorem 1.1.

In section 4, we introduce our LSH-based LDP k-means algorithm, algorithm 4. We start by giving a high level overview of the core ideas and advantages behind our algorithmic choices. We provide the pseudo-code in a modular fashion and analyse the cost guarantees of each subroutine in a separate subsection. The main result of this section is theorem 1.2.

2 Preliminaries

2.1 Problem Definition

We start by formally defining the k-means clustering problem.

Definition 2.1 (Non-private k-means)**.** For any Euclidean space E, let $z : E \times E \to \mathbb{R}$ denote the square of the ℓ_2 metric. Let D' be a data set of n points in $\mathbb{R}^{d'}$ such that $D' \subset B(0,1)$, the d'-dimensional unit ball of radius 1 centered at the origin. The k-means clustering cost $f_{D'}(S)$ of the data set D' for a set S of k points in $B(0,1)$ is defined by the expression

$$f_{D'}(S) = \sum_{p \in D'} z(p, S)$$

where we let $z(p, S) = \min_{q \in S} z(p, q)$. The k-means clustering problems asks one to find a set of k points in $B(0,1)$ such that $f_{D'}(\cdot)$ is minimized.

Remark 2.2. Both algorithms introduced in this work start with a dimension reduction so it will be convenient to let d' denote the dimension of the given ambient space and d denote the dimension of the space that the majority of the computation is done in. Similarly, D' is used to denote the original data set and D is used to denote the image of the data set in the dimension reduced space.

We require that our k-means algorithm also satisfy the constraints of local differential privacy. In this framework, the dataset is distributed among n agents each of whom has a single point of D, and the constraint of being locally differentially private requires that the transcript of any agent's responses is not too sensitive to their private data. This is formalized by appealing to the central model of differential privacy, which in turn is defined as follows:

Definition 2.3 (Differential privacy (DP), Dwork et al. [2006])**.** Two datasets $D_1, D_2 \in \mathcal{X}^n$ are *neighbouring* if they differ in at most one member element, i.e. $|D_1 \triangle D_2| = 1$. An algorithm $A : \mathcal{X} \to \mathcal{Y}$ is said to be (ϵ, δ)-*differentially private* (DP) if for any $S \subset \mathcal{Y}$ and any two neighbouring datasets $D_1, D_2 \in \mathcal{X}$,

$$P(A(D_1) \in S) \leq \exp(\epsilon) P(A(D_2) \in S) + \delta.$$

If $\delta = 0$, we can say that A is ϵ-differentially private.

Given the definition of the central model of differential privacy, local differential privacy is then defined as follows:

Definition 2.4 (Local differential privacy (LDP), Kasiviswanathan et al. [2011])**.** Consider a protocol which interacts with any one agent in some r rounds, and let the response of the agent with private data p be $A(p) = (A_1(p), \ldots, A_r(p))$, where $A_i(p)$ is the response of the agent in the ith round. We say that this protocol is (ϵ, δ)-*locally differentially private* (LDP) if the algorithm that outputs privatized responses for any agent $p \mapsto A(p)$ is (ϵ, δ)-differentially private. Again, if $\delta = 0$, we can say that a protocol is ϵ-locally differentially private.

Remark 2.5 (Notation)**.** We use $\tilde{O}(\cdot)$ to denote that certain terms have been suppressed in the argument. Concretely, in this notation we omit terms that are logarithmic in the multiplicative approximation factor α, the failure probability β and $\log n$. We use the expression $\operatorname{poly}\log n$ to denote terms that are $O(\log^p n)$ for some constant power p.

2.2 Dimension reduction for k-means clustering

In this subsection we recall some results about distance preserving dimension reduction maps that are fundamental to the construction of both algorithms described in this work. We follow the description in Makarychev et al. [2019], where the state of the art for the application of dimension reduction to ℓ_p clustering is stated and proved. We adopt the notation that for any $x, y, \alpha \in \mathbb{R}$, $x \simeq_{1+\alpha} y$ if $\frac{x}{1+\alpha} \leq y \leq (1+\alpha)x$, note that for any x, y, for all sufficiently small α, this is equivalent to $y = (1 \pm O(\alpha))x$.

Lemma 2.6 (Johnson-Lindenstrauss lemma, Johnson and Lindenstrauss [1984])**.** *There is a family of random linear maps $T_{d',d} : R^{d'} \to R^d$ with the property that for every $d' \geq 1$, $\alpha, \beta \in (0, 1/2)$ and all $x \in \mathbb{R}^{d'}$,*

$$P_{T \sim T_{d',d}} \left(\|Tx\| \in \left[\frac{\|x\|}{1+\alpha}, (1+\alpha)\|x\| \right] \right) \geq 1 - \beta,$$

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

This result is often cited in the form that for a data set $D' \subset \mathbb{R}^{d'}$ of size n, in order to preserve all pair-wise distances with probability $1 - \beta$ it suffices to reduce dimensions to $O\left(\log(n/\beta)/\alpha^2\right)$; this version follows directly by scaling the failure probability for the statement above by a factor of $1/n$ and applying the union bound. It is a well-known fact that the k-means clustering function can be written entirely in terms of pair-wise distances between the points in each cluster, i.e. for a k-means solution S that induces a partition $(C_1, \ldots, C_k)$,

$$f_{D'}(S) = \sum_{p \in D'} z(p, S) = \sum_{i=1}^{k} \sum_{p \in C_i} z(p, \mu_i) = \sum_{i=1}^{k} \frac{1}{2|C_i|} \sum_{p,q \in C_i} z(p, q).$$

It follows that preserving ℓ_2 distances within a $(1 \pm \alpha)$ approximation guarantees that the k-means clustering cost for the same cluster sets is preserved within a $(1 \pm O(\alpha))$ factor. This is the formulation that we appeal to for the multi-round clustering algorithm with low additive error.

For the purpose of k-ℓ_p clustering it has been shown that one can reduce dimensions far more aggressively; this line of work culminates in the following near-optimal result of Makarychev et al. [2019]:

Theorem 2.7 (Theorem 1.3 of Makarychev et al. [2019]). *Any family of linear maps $T_{d',d} : R^{d'} \to R^d$ that satisfies the conditions of the JL lemma and is sub-Gaussian tailed has the property that for any clustering $(C_1, \ldots, C_k)$ of D' with probability $1 - \beta$ over the choice of $T \sim T_{d',d}$*

$$\sum_{i=1}^{k} \frac{1}{2|S_i|} \sum_{p,q \in S_i} z(p,q) = \left(\sum_{i=1}^{k} \frac{1}{2|S_i|} \sum_{p,q \in S_i} z(Tp, Tq) \right) \left[\frac{1}{1+\alpha}, (1+\alpha) \right].$$

where $d = O(\log(k/\alpha\beta)/\alpha^2)$.

We recall that a family of linear maps $T_{d',d}$ is called sub-Gaussian-tailed if for every unit vector $x \in \mathbb{R}^{d'}$ and $t \geq 0$,

$$P_{T \sim T_{d',d}}(\|Tx\| \geq 1 + t) \leq \exp\left(-\Omega(t^2 d)\right).$$

For our purposes, we will also need a bound on the lengths of the vectors that holds with probability $1 - \beta$ after map reducing dimensions to $\log(k/\alpha\beta)/\alpha^2$. We can use the fact that the dimension reduction maps are sub-Gaussian-tailed to get the following bound.

Lemma 2.8. *For every point p in a dataset D' of size n, given a sub-Gaussian tailed dimension reducing family of maps $T_{d',d}$, we have that with probability $1 - \beta$, $\|Tp\| \leq O(\alpha\sqrt{\log n/\beta})\|p\|$.*

Proof. For any $p \in D'$ we have that

$$P_{T \sim T_{d',d}}(\|Tp\| \geq (1+t)\|p\|) \leq \exp\left(-\Omega\left(t^2 \frac{\log(k/\alpha\beta)}{\alpha^2}\right)\right).$$

It follows that there is a choice of $t = O(\sqrt{\log n/\beta \cdot \frac{\alpha^2}{\log(k/\alpha\beta)}}) = O(\alpha\sqrt{\log n/\beta})$ such that the bound above is at most β/n. Applying the union bound, the desired inequality follows. □

2.3 Fundamental privacy subroutines

We briefly recall a couple of standard results from the differential privacy literature that are used in the sequel. We rely on the following composition theorem which bounds the loss in privacy of the composition of multiple DP algorithms by appealing to their individual privacy guarantees in a modular fashion.

Theorem 2.9 (Basic Composition, Dwork et al. [2006]). *A mechanism with N adaptive interactions with (ϵ_i, δ_i)-DP mechanisms each for $i \in [N]$ and no other accesses to the database is $(\sum_{i\in[N]} \epsilon_i, \sum_{i\in[N]} \delta_i)$-DP.*

We also use the *Gaussian mechanism* and its privacy guarantee as formalized in the following lemma.

Lemma 2.10 (Gaussian mechanism, Dwork and Roth [2014]). *Given a d-dimensional function $f : \mathcal{X} \to \mathbb{R}^d$ which has ℓ_2-sensitivity $\max_{x,y\in\mathcal{X}} \|f(x) - f(y)\|_2 < \Delta_{f,2}$, randomized response via the Gaussian mechanism which for an agent with private data p returns $f(p) + Y$ for $Y \sim N(0, \frac{c_G^2 \Delta_{f,2}^2}{\epsilon^2} \mathbb{I}_{d\times d})$ is (ϵ, δ)-differentially private for any $c_G^2 > 2\ln(1.25/\delta)$.*

2.4 Bitstogram and the Heavy Sums Oracle

The contents of this subsection are used in the cost analysis for both clustering algorithms. In the sequel we make extensive use of locally private frequency estimation. For private frequency estimation a lower bound of $\Omega_\epsilon(\sqrt{n})$ is known [Chan et al., 2012]. A state of the art construction for this problem is the Bitstogram algorithm Bassily et al. [2020], which is an ϵ-LDP algorithm for the heavy-hitters problem that achieves low error.

Lemma 2.11 (Algorithm Bitstogram, Bassily et al. [2020])**.** *Let V be a finite domain of values, let $f : D' \to V$, and let $n(v)$ denote the frequency with which v occurs in $f(D')$. Let $\epsilon \leq 1$. Algorithm* Bitstogram(f, ϵ, β) *interacts with the set of n users in 1 round and satisfies ϵ-LDP. Further, it returns a list $L = ((v_i, a_i))_i$ of value-frequency pairs with length $\tilde{O}(\sqrt{n})$ such that with probability $1 - \beta$ the following statements hold:*

1. *For every $(v, a) \in L$, $\|a - f(v)\| \leq E$ where $E = O\left(\frac{1}{\epsilon}\sqrt{n \log(n/\beta)}\right)$.*
2. *For every $v \in V$ such that $f(v) \geq M$, $v \in L$, where $M = O\left(\frac{1}{\epsilon}\sqrt{n \log |V|/\beta \log(1/\beta)}\right)$.*

We overload notation to treat the list returned by Bitstogram *returns as either a set of (heavy-hitter, frequency) pairs or a function which may be queried on a value to return either the corresponding frequency if it is a heavy hitter or a value of 0 otherwise. A subscript of M will denote the upper bound on the maximum frequency omitted. We see that whenever $|V| = \Omega(n)$, $M = \Omega(E)$ and* Bitstogram *promises a uniform error bound of M when estimating the frequency of any element in the co-domain for an appropriate choice of constants.*

We introduce an extension of the Bitstogram algorithm called HeavySumsOracle that allows us to query the sums of some vector valued function over the set of elements that map to a queried heavy-hitter value. For a given value-mapping function $f : \mathcal{X} \to \mathcal{V}$ and a vector-valued function $g : \mathcal{X} \to \mathbb{R}^d$ the sum estimation oracle privately returns for every heavy hitter $v \in \mathcal{V}$ the sum of all agents that map to x, i.e. $\sum_{p:f(p)=x} p$. We recall that Bitstogram is a modular algorithm with two subroutines; a frequency oracle that privately estimates the frequency of any value in the data universe, and a succinct histogram construction that constructs the heavy hitters in a bit-wise manner by making relatively few calls to the frequency oracle. The construction of HeavySumsOracle essentially mimics the frequency oracle construction called Hashtogram from Bassily et al. [2020] and can be run in parallel with Bitstogram, allowing us to reduce the round complexity of our protocols. The pseudo-code and proof of lemma 2.12 may be found in appendix B.

Lemma 2.12 (HeavySumsOracle)**.** *Let $f : \mathcal{X} \to \mathcal{V}$, $g : \mathcal{X} \to B(0, \Delta/2) \subset \mathbb{R}^{d'}$ be some functions where g has bounded sensitivity $\Delta_{g,2}$ and let $D' \subset \mathcal{X}$ be a distributed dataset over n users. With probability at least $1 - \beta$, for every $v \in \mathcal{V}$ that occurs in $f(D')$, if $S(v)$ is the value returned by Algorithm 6 then*

$$\left\| S(v) - \sum_{f(y)=v} g(y) \right\| \leq 2\Delta \sqrt{2n \log \frac{d'+1}{\beta}} + \frac{4c_G \Delta_{g,2}}{\epsilon} \sqrt{2d'n \log \frac{4}{\beta}}.$$

Here c_G is the constant derived from the Gaussian mechanism (lemma 2.10), and $\Delta_{g,2}$ is the ℓ_2-sensitivity of g. Note that since $\Delta_{g,2} \leq \Delta$, this also implies (whenever $\epsilon < c_G = \sqrt{2 \ln(1.25/\delta)}$)

$$\left\| S(v) - \sum_{f(y)=v} g(y) \right\| \leq O\left(\frac{c_G \Delta}{\epsilon} \sqrt{d'n \log \frac{1}{\beta}} \right).$$

Further, Algorithm 6 is (ϵ, δ)-LDP.

2.5 Locality Sensitive Hashing

The contents of this subsection are used only for the construction and analysis of the multi-round k-means algorithm with low additive error. We start by recalling the definition of an LSH family. Complete proofs may be found in the appendix.

Definition 2.13 (Locality sensitive hashing (LSH))**.** We say that a family of hash functions $H : \mathbb{R}^d \to B$ for a finite set of buckets B is *locality-sensitive* with parameters (p, q, r, cr) if for every $x, y \in \mathbb{R}^d$ for some $1 \geq p > q \geq 0$, $r > 0$ and $c > 1$

$$P(H(x) = H(y)) \begin{cases} \geq p \text{ if } d(x, y) \leq r \\ \leq q \text{ if } d(x, y) \geq cr. \end{cases}$$

In this work we use an LSH family construction from Andoni and Indyk [2006].

Theorem 2.14. *For every sufficiently large d and n there exists a family $\mathcal{H}$ of hash functions defined on $\mathbb{R}^d$ such that for a dataset of size n,*

1. *A function from this family can be sampled, stored and computed in time $t^{O(t)} \log n + O(dt)$, where t is a free positive parameter of our choosing.*
2. *The collision probability for two points $u, v \in \mathbb{R}^d$ depends only on the ℓ_2 distance between them, which we henceforth denote by $p(\|u - v\|)$.*
3. *The following inequalities hold:*

$$p(1) \geq \frac{A}{2\sqrt{t}} \frac{1}{(1+\epsilon+8\epsilon^2)^{t/2}}$$
$$\forall c > 1,\ p(c) \leq \frac{2}{(1+c^2\epsilon)^{t/2}}$$

   where A is an absolute constant < 1, and $\epsilon = \Theta(t^{-1/2})$. One can choose $\epsilon = \frac{1}{4\sqrt{t}}$.

4. *The number of buckets N_B an LSH function with parameter t uses is $t^{O(t)} \log n$.*

Note that by scaling the input to the LSH function this gives us constructions for (p, q, r, cr)-sensitive LSH families for arbitrary values of $r > 0$. Due to the occurrence of terms like $t^{O(t)}$ in the collision probabilities and the number of buckets, the performance of an LSH family is very sensitive to the choice of t. In the following lemma we show how to choose a value of t for a desired ratio of $p^2(1)$ to $p(c)$.

Lemma 2.15. *Given a fixed $c > \sqrt{2}$, for any $B > 1$, there is a choice of $t = O\left(\log^2 B\right)$ for the LSH function described in theorem 2.14 such that*

$$\frac{p^2(1)}{p(c)} = \Omega(B),$$
$$p(1) \geq \Omega(B^{-1/c'}/\log B),$$
$$\log N_B = O(\log^2 B \log\log B + \log\log n),$$

where $c' = (c^2/8 - 1/4)$. It will be convenient to note that $1/c' = O(1/(2c^2 - 1))$.

In the construction of the multi-round k-means algorithm with low additive error, we will need to estimate the average of all points that map to a given heavy bucket. Due to the pair-wise nature of the LSH guarantee, the analysis of this requires us to use an arbitrary point from the bucket as a filter to ensure that sufficiently many points close to it and not too many points far from it map to that bucket.

Lemma 2.16. *Let $C \subset D$ be a set of points with diameter r and let the diameter of D be Δ. For any $x_0 \in C$, if $\hat{x_0}$ is the average over all points colliding with x_0 under a $(p(1), p(c), r, rc)$-sensitive LSH function H applied to D, then with probability $p(1)/4$,*

$$\|x_0 - \hat{x}_0\| \leq cr + \frac{8p(c)|D|}{p^2(1)|C|}\Delta,$$

and the number of points of C that collide with x_0 is at least $\frac{p(1)C}{2}$.

Notation	Meaning
$D' \subset \mathbb{R}^{d'}$	Original data set
$Q : \mathbb{R}^{d'} \to \mathbb{R}^d$	mapping from high-dim. to low-dim. space
$D \subset \mathbb{R}^d$	$Q(D')$, dimension reduced data set
G_l	Rectangular grid in dimension reduced space
$G_l(\cdot)$	Mapping from $\mathbb{R}^d$ to coordinate-wise floor in G_l
t_l	Unit length of grid G_l
PH^l	Succinct histogram of number of points mapping to $g \in G_l$ for "heavy" g
Count$(\cdot)$	Count of previously uncovered data points $g \in G_l^*$ serves
G_l^*	Candidate centers picked from grid points in G_l, $G_l^* \subset \mathsf{PH}^l$
N_G	A $k^{\tilde{O}(1/\alpha^2)}$ term used to greedily pick G_l^*
PSO^l	Vector sums of points in original space mapping to $g \in G_l$ for "heavy" g
Sum$(\cdot)$	Sum of previously uncovered points in original space whose image served by $g \in G_l^*$
$G_{l,i}^*$	Points of G_l^* for which $s_i^* \in S^*$ is closest center
M_l^*	Maximal grid points picked from $G_1, \ldots, G_l$
$D^* \subset \mathbb{R}^d$	Proxy data set generated by weighing points in $G_1^*, \ldots, G_L^*$ by points served
S^*	k-means solution derived by clustering D^*
S'	k-means solution for D' output by algorithm 1

Table 2: Summary of notation used in algorithm 1

3 LDP k-means with arbitrarily tight multiplicative approximation

In this section we describe a one-round k-means clustering algorithm and formally analyse its cost and privacy guarantees. We start by describing our algorithm and provide the pseudo-code. We then give an informal description of our methods and a high-level justification for various algorithmic choices. In one line, what we will do is find a small collection of candidate centers for the bi-criteria relaxation to the k-means problem, derive cluster centers for a proxy data set derived by weighing the candidate centers by counts of points served, and use these cluster centers to cluster the original data set.

3.1 Pseudo-code and algorithm description

Step 1 - Initialization and interaction: From line 1 to line 12, we first formalize the publicly available dimension reduction, scaling and projection required to ensure that every point lies inside the domain $B(0,1)$; this is the map Q. We define $L = \lg n$ grids $G_1, \ldots G_L$ where G_l has unit length $t_l = 2^{l-L+1}/\alpha\sqrt{d}$. This definition ensures that the distance from any point in the space to its coordinate-wise floor is at most $\alpha 2^{l-L+1}$ units. The end of Step 1 occurs by L calls to the Bitstogram and HeavySumsOracle routines to privately generate succinct histograms PH^l and sum oracles PSO^l over points mapping to any given grid-point.

Step 2 - Construction of proxy dataset From line 13 to line 30 we iteratively construct the proxy data set by going from low to high threshold and greedily picking some $2N_G \log 1/\alpha$ grid points G_l^* that maximize the Count$(\cdot)$ function. The Count$(\cdot)$ function maintains estimate of the number of previously uncovered data points that would be covered by $g \in G_l$ if picked. We also keep track of the "maximal" grid points in the sets M_l^*; at the beginning of round l, M_{l-1}^* consists of all grid points that have been picked so far that have the property that no grid point which would cover them has yet been picked. This will ensure that when we update the Count$(\cdot)$ function to account for data points that have already been covered, we do not subtract for any one data points multiple times. Along the way we mimic the Count$(\cdot)$ construction by generating a similar Sum$(\cdot)$ mapping that estimates the vector sum of all points in the original space served by $g \in G_l$. This step ends with the construction of the proxy dataset D^* where we repeat each grid point $g \in G_l^*$ with multiplicity equal to the number of data points it served, i.e. Count(g).

Data: Data set $D' \subset \mathbb{R}^{d'}$ distributed over n agents, privacy parameters ϵ, δ, accuracy parameter α, failure probability β

/* Step 1: Initialization and interaction */

$T : \mathbb{R}^{d'} \to \mathbb{R}^d$ dimension reduction for $d = O(\log(k/\alpha\beta)/\alpha^2)$

$S : \mathbb{R}^d \to \mathbb{R}^d$ scaling by a factor $\Omega(1/(\alpha\sqrt{\log n/\beta}))$

$P : \mathbb{R}^d \to B(0,1)$ projection to the unit ball

$Q = P \circ S \circ T : \mathbb{R}^{d'} \to B(0,1) \subset \mathbb{R}^d$; /* Publicly available mapping */

$L = \lg n$ number of grids in dimension reduced space

$t_l = 2^{l-L+1}/\alpha\sqrt{d}$ for $l = 1, \ldots, L$

$G_l = t_l(\mathbb{Z}^d)$ grid with unit length t_l

$G_l : \mathbb{R}^d \to G_l$ map flooring points coordinate-wise to the grid G_l ; /* Overloaded notation */

do in parallel for $l \in [L+1]$**:**

  $\mathsf{PH}^l \leftarrow \mathsf{Bitstogram}(G_l \circ Q, \epsilon, \beta)$; /* Get frequency oracle for number of points snapping to grid point */

  $\mathsf{PSO}^l \leftarrow \mathsf{HeavySumsOracle}(G_l \circ M, p \mapsto p, \epsilon, \beta)$; /* Get sum oracle for points mapping to grid point */

end

/* Step 2: Construction of proxy data set */

$M_0^* \leftarrow \emptyset$; /* Keeps track of "maximal" points in grid */

for $l = 1, \ldots, L$ **do**

  $(\text{Count} : G_l \to \mathbb{R}) \leftarrow \mathsf{PH}^l(\cdot)$

  $(\text{Sum} : G_l \to \mathbb{R}^{d'}) \leftarrow \mathsf{PSO}^l(\cdot)$

  for $g \in M_{l-1}^*$ **do**

    $\text{Count}(G_l(g)) \leftarrow \text{Count}(G_l(g)) - \mathsf{PH}(g)$

    $\text{Sum}(G_l(g)) \leftarrow \text{Sum}(G_l(g)) - \mathsf{PSO}(g)$

  end

  $G_l^* \leftarrow \{(g, \text{Count}(g)) : g \in \lceil 2N_G \log 1/\alpha \rceil \text{ points with largest values of Count in } G_l\}$

  $M_l^* \leftarrow M_{l-1}^*$

  for $g \in M_l^*$ **do**

    if $G_l(g) \in G_l^*$ **then**

      $M_l^* \leftarrow M_l^* \backslash \{g\}$

    end

  end

  $M_l^* \leftarrow M_l^* \cup G_l^*$

end

$D^* \leftarrow \{g \text{ with multiplicity Count}(g) \text{ for } (g, \text{Count}(g)) \in G_1^*, \ldots, G_L^*\}$; /* Proxy data set */

/* Step 3: Cluster center recovery */

$S^* = \{s_1^*, \ldots, s_k^*\} \leftarrow \mathsf{Standard}\ k - \mathsf{Means}_\eta(D^*)$

$G_{l,i}^* \leftarrow \{g \in G_l^* : \arg\min_{s \in S^*} z(g,c) = s_i^*\}$ for each level $l = 1, \ldots, L$ and cluster center $s_i^* \in S$

for $j = 1, \ldots, k$ **do**

  $\text{Sum} \leftarrow \sum_{l=1}^{L} \sum_{g : \in G_l^*(s_j^*)} \text{Sum}(g)$

  $\text{Count} \leftarrow \sum_{l=1}^{L} \sum_{g : \in G_l^*(s_j^*)} \text{Count}(g)$

  $\hat{\mu}_j \leftarrow \frac{\text{Sum}}{\text{Count}}$

end

return $S' = \{\hat{\mu}_1, \ldots, \hat{\mu}_k\}$

Algorithm 1: 1-Round k-means Clustering

Step 3 - Cluster center recovery: From line 31 to line 38 we compute the final cluster centers S' in the original space. We start by first using a non-private k-means algorithm Standard $k-$ Means$_\eta$ with an η-multiplicative approximation guarantee on the privately derived proxy data set D^* to derive cluster centers S^* in the low dimensional space. Then, we iterate over each cluster center $s_i^* \in S^*$ and for every fixed cluster center we use the Sum$(\cdot)$ functions constructed in step 2 to compute the vector sums over all points snapping to grid points $G_{l,i}^*$ which are closer to s_i^* than to any other center in S^*, as well as the count of all such data points (via Count). Our estimate for the true average of this cluster in the original space is then simply $\hat{\mu}_i = \text{Sum/Count}$, and these k estimates $\{\hat{\mu}_1, \ldots, \hat{\mu}_k\}$ form the final output of our algorithm.

3.2 Technical discussion

We recall from the introduction that for the error we are targeting we need to find $O(\text{poly}\, k\, \text{poly}\log n)$ candidate centers with respect to which additive error in a $1+\alpha$ approximation to OPT is $O(\text{poly}\, k\sqrt{n}\, \text{poly}\log n)$. We recall from that discussion that in the LDP setting one approach to get around the large amount of error added is to discretize the response of the agents. A natural way to achieve this is to the domain is via a rectangular d'-dimensional grid of points and ask agents to reveal their closest grid point; the question then becomes how best to exploit this privately derived information for k-means clustering. Previous work on k-means clustering in the central DP setting [Chaturvedi et al., 2020] uses such an approach where in order to get an $O(1)$ multiplicative factor approximation to the optimal clustering cost, a sequence of grids is used where the unit length of the lth grid equals $2^{-l}/\sqrt{d}$.

To analyse this approach, one fixes an arbitrary optimal solution to the k-means problem S_{OPT} and partitions the data set based on how far a point lies from the optimal solution via geometrically increasing thresholds 2^{-l}. Then for any point p which lies at a distance between 2^{-l} and 2^{-l+1}, the closest grid point to p in the grid with unit length $2^{-l}/\sqrt{d}$ is at a distance of at most 2^{-l}, i.e. closer than the optimal solution. One then reverses the argument to observe that if a point lies within a distance of 2^{-l+1} units of S_{OPT}, then by the triangle inequality its closest grid point must lie within a distance of $O(2^{-l+1})$ units of S_{OPT}. The authors then bound the total number of grid points that lie within any collection of k centers to derive the promise that there is a small set of grid points which serve almost all data points which lie at a distance between 2^{-l} and 2^{-l+1} of the candidate centers.

Choice of grid construction: As in this work we are targeting a $(1+\alpha)$ multiplicative approximation, we scale the grid unit lengths by a factor of α to get the promise that if a point lies within a distance of 2^{-l} of S_{OPT}, it lies within a distance of $O(\alpha 2^{-l})$ of some grid point. Since we can only identify grid points whose counts are at least $\sqrt{n}$, we can afford to miss at most $O(\text{poly}\, k\, \text{poly}\log n)$ many such grid points serving the dataset across all levels for an additive error of $O(\text{poly}\, k\sqrt{n}\, \text{poly}\log n)$. It follows that we will need an $O(\text{poly}\, k\, \text{poly}\log n)$ bound on the number of grid points that lie close to the optimal centers. We will address this point further ahead in this discussion.

One technicality suppressed so far is that we must have a finite (in fact $O(\text{poly}\, k\, \text{poly}\log n)$) sequence of grids and thresholds for the set of candidate centers accrued across grids to be finite. We observe that if the smallest threshold is $1/n$, then the discretization error for all points which lie within $1/n$ of S_{OPT} is absorbed by an additive $O(1)$ term instead of the $O(1)$ multiplicative approximation factor; this in conjunction with the fact that the diameter of the domain is $O(1)$ shows that a set of $O(\log n)$-many thresholds suffices.

Returning to the identification of grid points close to S_{OPT} in the grid, we observe that there is an issue with this approach; the choice of S_{OPT} was arbitrary and different choices can possibly lead to very different sets of grid points close to S_{OPT}. It is not immediately clear what is a good way to pick grid points when we are oblivious of any choice of S_{OPT} using only the grid points histogram data.

Greedy maximum coverage: Reasoning along the lines of Jones et al. [2020] for the k-medians problem shows that a choice of grid points that greedily maximizes how many data points are covered by including these grid points among the candidate centers ensures that the clustering cost of the data set with respect to this set of grid points is at most $O(\text{OPT})$. Since the number of grid points is larger than k, and the cost

is a constant factor multiplicative approximation to OPT, this set of grid-points chosen across grids is a solution to the bi-criteria relaxation of the k-means solution (modulo some additive error).

A closer look at the argument in Jones et al. [2020] shows that the greedy picks must maximize coverage only over yet-uncovered points, when proceeding from low to high thresholds. In the centrally private setting one can dynamically update the coverage of candidate grid points by directly accessing the data set and marking points off as they are covered, but this is not possible in the local setting. We get around this hurdle by two tools; one, ensuring a *consistency* across grids in the sense that if two points map to the same grid point in a low-level grid then they also map to the same grid point in all higher-level grids; and two; keeping track of all grid points picked so far such that they are *maximal* in the sense that no grid point that they themselves snap to in a coarser grid has been picked. We will then be able to evaluate the count of yet uncovered data points covered by any candidate grid point by simply subtracting the histogram counts of all maximal grid points picked so far snapping to that candidate grid points from the histogram count of that candidate grid point.

We will ensure consistency by mapping each point to its *coordinate-wise floor* in the d-dimensional grid instead of its closest point; this makes no significant different in the arguments made so far as the floor always lies within a distance of 2^{-l} in a grid with unit length $2^{-l}/\sqrt{d}$.

Dimension reduction for bounded candidate centers: We now discuss how to get the $O(\mathrm{poly}\, k\, \mathrm{poly}\log n)$ bound on the number of grid points within the aforementioned threshold distance of S_{OPT}. For reasons of time and space efficiency, in Chaturvedi et al. [2020] the authors needed to bound the number of grid points close to any choice of S_{OPT} by $O(\mathrm{poly}(n))$. They showed that by dimension reduction to $O(\log n/\alpha^2)$ many dimensions, there are at most $O(n^{1/\alpha^2})$ many grid points within a distance of r of any optimal center for a grid with unit length $\alpha r/\sqrt{d}$. They then appeal to the well-known Johnson-Lindenstrauss lemma that shows that there is a choice of $O(\log n/\alpha^2)$ many dimensions so that the ℓ_2 distance between all pairs of data points in a data set of size n is preserved within a multiplicative factor of $(1 \pm \alpha)$. It is relatively easy to show that the k-means clustering cost for any choice of clusters is also preserved within a factor of $(1 \pm \alpha)$.

A recent work by Makarychev et al. [2019] generalized the Johnson-Lindenstrauss guarantee for k-means clustering by showing that in fact performing dimension reduction to $\log(k/(\alpha\beta))/\alpha^2$-dimensions ensures that with probability $1 - \beta$ the cost of every clustering solution is preserved within a multiplicative cost of $(1 \pm \alpha)$. By tracing the argument of Chaturvedi et al. [2020] for upper bounding the number of grid points close to any optimal center with this stronger bound on the dimensionality of the dimension-reduced space leads to a $k^{\tilde{O}(1/\alpha^2)}$ bound on the number of grid points close to S_{OPT}. For any fixed approximation factor $(1 + \alpha)$, this immediately gives us the desired $O(\mathrm{poly}\, k\, \mathrm{poly}\log n)$ bound on the number of grid points close to S_{OPT} as well as the $O(\mathrm{poly}\, k\, \mathrm{poly}\log n)$ bound on the number of candidate centers picked.

Proxy data set construction: To recap, the set of candidate centers derived to construct the proxy data set has the property that for all but $O(\mathrm{poly}\, k\, \mathrm{poly}\log n)$ many data points, there is a candidate center at a distance of $O(\alpha)$ times the distance between a data point and the optimal centers. We construct the proxy data set by repeating each candidate center with an estimate of the number of points it covers in this manner. This can be seen as essentially *moving* each data point to the candidate center that covers it; in sum what we have shown is that the net movement is $O(\alpha\, \mathrm{OPT})$. We can then show by the triangle inequality that the k-means clustering functions of the original and the proxy data set are within a $(1+O(\alpha))$ multiplicative approximation factor and $O(\mathrm{poly}\, k\, \mathrm{poly}\log n)$ additive error. It will follow that the optimal clustering cost for the proxy data set is a $(1+\alpha)$ factor more than the optimal cost for the original data set (modulo additive error), and therefore that any clustering solution derived by a non-private k-means clustering algorithm with multiplicative approximation factor η has net clustering cost at most $(1 + \alpha)\eta$. Using the relation between the k-means clustering functions this time in reverse, we get that the privately derived cluster centers for the proxy data set serve as cluster centers for the original data set with cost $(1 + O(\alpha))\eta$.

Undoing the dimension reduction: We have privately derived k cluster centers in the dimension reduced

space that serve the data set D with clustering cost $(1+O(\alpha))\eta\,\mathrm{OPT}$ and additive error $O(\mathrm{poly}\,k\,\mathrm{poly}\log n)$. This implicitly clusters the original data set with a similar error guarantee by mapping each data point to a cluster corresponding to the center in the low-dimensional solution that its image is closest to. To compute the centers of these clusters in the original space, we use the *sum oracles* derived from calls to HeavySumsOracle to recover the vector sums of all data points in the original space that lie in these implicitly defined clusters. Dividing these sums by the counts derived during our proxy data set construction gives us good estimates to the cluster-wise centers.

3.3 Formal cost and privacy analysis

Proof outline: We begin by relating the optimal clustering cost in the original space OPT', and the clustering cost in the dimension reduced space OPT (lemma 3.2). We then formally derive some properties of the grids G_l, the maximal points identified at the end of round l i.e. M_l^*, and the accuracy of the Count map used in step 2 to choose grid points as candidate cluster centers (lemma 3.3 to lemma 3.7). Since the error bound for the Sum map is practically identical to that of the Count map, we prove that in immediate succession.

The core of our cost analysis for the bi-criteria solution is showing that the clustering cost of the data set with respect to many greedy choices of candidate centers is competitive with the optimal clustering (definition 3.9 and lemma 3.10). These results allow us to show in that the k-means clustering functions for the dimension reduced data set D and the proxy data set D^* are close in ℓ_1 norm (lemma 3.12). Lemma 3.12 is then exploited in turn to show that the output of the non-private clustering algorithm works well for the original dimension reduced data set (corollary 3.13).

Finally, starting from definition 3.14, we start the work of recovering cluster center in the original space. We begin in the definition by formalizing the actual clustering of the dimension reduced data set that results from identifying each data point with the first grid point that serves it in some grid. Then we show that the output of the algorithm works well for this actual clustering and leads to a $(1+O(\alpha))\eta$ factor multiplicative approximation (lemma 3.15 to lemma 3.17). This section culminates in the main result theorem 1.1 which accounts for all privacy loss which occurs across all calls to Bitstogram and HeavySumsOracle and after scaling the privacy parameters in the calls to these routines formalizes the final cost guarantee of algorithm 1.

Definition 3.1. We recall some notation used in the algorithm description and introduce some definitions that help with the cost analysis for this algorithm.

- There is a sensitive dataset $D' \subset B(0,1)$ distributed among n users, exactly one point per user. We denote the cost of the optimal k-means solution by OPT'.
- Let $Q : \mathbb{R}^{d'} \to \mathbb{R}^d$ be a publicly available function that maps the data domain to $B(0,1)$ in the dimension reduced space $\mathbb{R}^d$. It is computed by first computing the output of the dimension reduction map T, followed by a scaling S by $1/\alpha\sqrt{\log n}$ units (which ensures that with high probability all points lie inside the unit ball in the dimension reduced space, followed by a projection P to the unit ball to deal with any outliers.
- We denote the dimension-reduced data set $Q(D')$ by D. We denote its optimal clustering cost by OPT. We fix any optimal k-means solution S_{OPT} for D with clustering cost OPT.
- Let $L = \lceil \lg n \rceil$ denote the number of levels.
- Let $r_l = 2^l/2^{L-1}$ for $l = 1, \ldots, L$ denote the ℓ_2^2 distances which we use as thresholds to partition D depending on how far points lie from S_{OPT}. Note that $r_1 < 1/n$ and $r_L = 2$. Further, we set $r_0 = 0$. With this notation we see that $D \subset B(0,1) \subset B(p, r_L)$ for any $p \in B(0,1)$.
- Let $o_l := \{p \in D : z(p, S_{\mathrm{OPT}}) \in [r_l, r_{l+1})\}$ for $l = 1, \ldots, L$ denote the thresholded partitions of D. Note that with our choice of r_l this definition implies $D = \sqcup_{i=l}^{L} o_l$.

- Let $t_l = \alpha r_l/\sqrt{d}$ denote the unit length of the grid G_l for $l = 1, \ldots, L$. Let G_l be the axis aligned grid of unit length t_l units centered at the origin in $B(0,1)$, i.e. $G_l := B(0,1) \cap (t_l \mathbb{Z}^d)$. We overload notation and let $G_l(\cdot) : \mathbb{R}^d \to G_l$ map each point to its floor in G_l, i.e. p maps to $t_l(\lfloor p_1/t_l \rfloor, \ldots, \lfloor p_d/t_l \rfloor)$. Note that these multidimensional floor maps are consistent in the sense that for any $m > i$, for the jth coordinate we have

$$\begin{aligned} G_m \circ G_l(p)_j &= t_m \lfloor t_l \lfloor p_j/t_l \rfloor / t_m \rfloor \\ &= t_m \lfloor 2^{l-m} \lfloor p_j/t_l \rfloor \rfloor \\ &= t_m \lfloor 2^{l-m} p_j/t_l \rfloor \\ &= t_m \lfloor p_j/t_m \rfloor \\ &= G_m(p)_j \end{aligned}$$

  so putting all coordinates together $G_m \circ G_i(p)_j = G_m(p)$. Note that $t_m \lfloor 2^{l-m} \lfloor p_j/t_l \rfloor \rfloor = t_m \lfloor 2^{l-m} p_j/t_l \rfloor$ because $1/2^{l-m} \in \mathbb{Z}$.

- We assume each grid point is implicitly tagged with the index of its parent grid point. We will abuse notation and drop indices for the succinct histograms PH^l and PSO^l where they may be deduced from the grid point for which the frequency or sum is being queried.

Lemma 3.2 (Accounting for dimension reduction)**.** *With probability $1-\beta$, we have that for every clustering $(D'_1, \ldots, D'_k)$ of D',*

$$\sum_{i \in k} \sum_{p \in D'_i} s\left(p, \frac{\sum_{q \in D'_i} q}{|D'_i|}\right) \simeq_{1+\alpha} (\alpha \log n/\beta) \sum_{i \in k} \sum_{p \in Q(D'_i)} s\left(Q(p), \frac{\sum_{q \in D'_i} M(q)}{|D'_i|}\right).$$

As a direct corollary $\mathrm{OPT}' \simeq_{1+\alpha} (\alpha \log n/\beta)\, \mathrm{OPT}$.

Proof. We write $Q = P \circ S \circ T$, where T is the dimension reduction to $O(\log(k/\alpha\beta)/\alpha^2)$, S is the scaling by a factor of $\Omega(1/\alpha\sqrt{n/\beta})$, and P is projection to the unit ball. Given any clustering $(D'_1, \ldots, D'_k)$ of D', by theorem 2.7 we have that

$$\sum_{i \in k} \sum_{p \in D'_i} s\left(p, \frac{\sum_{q \in D'_i} q}{|D'_i|}\right) \simeq_{1+\alpha} \sum_{i \in k} \sum_{p \in D'_i} s\left(T(p), \frac{\sum_{q \in D'_i} T(q)}{|D_i|}\right).$$

The scaling map changes all ℓ_2-distances by precisely the scaling factor, so we also have that

$$\sum_{i \in k} \sum_{p \in D'_i} s\left(T(p), \frac{\sum_{q \in D'_i} T(q)}{|D'_i|}\right) = (\alpha \log n/\beta) \sum_{i \in k} \sum_{p \in S \circ T D'_i} s\left(S \circ T(p), \frac{\sum_{q \in D'_i} S \circ T(q)}{|D'_i|}\right).$$

Finally, since with probability $1-\beta$ all points lie in the unit ball after scaling by a factor of $1/\alpha\sqrt{\log n/\beta}$, the projection map does not move any point and hence the same clustering cost holds for $P \circ S \circ T(D') = Q(D')$. □

In the following lemma we derive a bound on the discretization error and use that to derive a promise that in every level l if we snap o_l to the grid then we get at most $k^{\tilde{O}(1/\alpha^2)}$ many grid points.

Lemma 3.3 (Properties of grids G_l)**.** *For all $l = 1, \ldots, L$, the following bound statements hold for each grid G_l:*

1. *For any $p \in B(0,1)$, $\|p - G_l(p)\| \le \alpha 2^{-l} = t_l\sqrt{d} = \alpha r_l$.*
2. *$\left|G_l(\cup_{j=1}^{l} o_j)\right| = k^{O(1/\alpha^2)}$.*

Proof. 1. By definition $G_l(p) = t_l(\lfloor p_1/t_l \rfloor, \ldots, \lfloor p_d/t_l \rfloor)$. Since $|p_j/t_l - \lfloor p_j/t_l \rfloor| \leq 1$, it follows that $|p_j - G_l(p)_j| \leq t_l \Rightarrow \|p - G_l(p)\| \leq t_l\sqrt{d} = \alpha r_l$.

2. Let $p \in (\cup_{j=1}^{l} o_j)$. By definition of $z(\cdot,\cdot)$, $z(G_l(p), S_{\text{OPT}}) \leq z(G_l(p), \arg\min_{c \in S_{\text{OPT}}} z(p,c))$. Then, since $z(p, G_l(p)) \leq \alpha^2 r_l^2 = O(\alpha^2 r_l)$ and $r_l = O(z(p, \arg\min_{c \in S_{\text{OPT}}} z(p,c)))$, by the weak triangle inequality $z(G_l(p), S_{\text{OPT}}) \leq (1 + O(\alpha))r_l$.

   Since we have shown $G_l(\cup_{j=1}^{l} o_j) \subset \{g \in G_l : z(g, S_{\text{OPT}}) < (1 + O(\alpha))r_l\}$, it will suffice to bound the size of the latter set. Fix any $s \in S_{\text{OPT}}$. If $g \in G_l$ is such that $z(g,s) \leq (1 + O(\alpha))r_l$ then by the weak triangle inequality $z(G_l(s), g) \leq (1 + O(\alpha))r_l$. By translating $G_l(s)$ and G_l so that $G_l(s)$ lies at the origin and scaling the space up by a factor of $1/t_l$ so that G_l maps onto $\mathbb{Z}^d$, we see that there is an injection from $\{g \in G_l : z(g,s) \leq (1 + O(\alpha))r_l)\}$ into $V = \{j \in \mathbb{Z}^d : z(j,0) \leq d/\alpha^2 + O(1)\}$.

   Expanding the definition of V, we get

$$v \in V \Rightarrow \sum_{i \in [d]} v_i^2 \leq d/\alpha^2 + O(1).$$

   It follows that the number of unsigned $v \in V$ is at most the number of ways of partitioning $d/\alpha^2 + O(1)$ balls into $d+1$ distinguishable bins. Then,

$$\begin{aligned} |V| &= \binom{d/\alpha^2 + O(1)}{d+1} \\ &< \left(\frac{e \cdot (d/\alpha^2 + O(1))}{d+1}\right)^{d+1} \\ &= k^{\tilde{O}(1/\alpha^2)} \end{aligned}$$

   where we use that $d = O\left(\log(k/(\alpha\beta))/\alpha^2\right)$. Hence, $|\{g \in G_l : z(g, S_{\text{OPT}}) < (1 + O(\alpha))r_l\}| = k \cdot 2^d \cdot k^{\tilde{O}(1/\alpha^2)} = k^{\tilde{O}(1/\alpha^2)}$.

□

Definition 3.4. We make a couple of definitions to ease our analysis from this point.

1. Let N_G be a uniform upper bound on the the sizes of the sets $G_l(\cup_{j=1}^{l} o_j)$. It follows from lemma 3.3 that we can choose a value of $N_G = k^{\tilde{O}(1/\alpha^2)}$.

2. We define a sequence of subsets a_l inductively. Let $a_1 = \{p \in D : G_1(p) \in G_1^*\}$ and let $a_l = \{p \in D : G_l(p) \in G_l^*\} \setminus \left(\cup_{j=1}^{l-1} a_j\right)$. Informally, a_l consists of those points which were not explicitly covered at a distance of αr_j for any $j < l$ but are successfully covered by some $g \in G_l^*$ at an ℓ_2 distance of αr_l, since its floor in the grid was added to G_l^*.

3. M_l^* is the set of grid points constructed iteratively by adding all grid points picked in round l from G_l to grid points picked in previous rounds and then removing all grid points picked in previous rounds which snap to any grid point picked in round l. Intuitively, we can think of this set as the set of "maximal" grid points that have been picked so far. Keeping track of this set will allow us to avoid over-counting data points being covered at different levels and keep private estimation error terms small.

Lemma 3.5 (Properties of maximal grid point sets M_l^*)**.** *The following properties hold for the sets M_l^* for $l = 1, \ldots, L$.*

1. *If $p \in a_j$ for some $j \leq l$ then $\exists! k \in \{j, \ldots, l\}$ such that $G_k(p) \in M_l^*$.*
2. *$|M_l^*| = lN_G$*

Proof. 1. Given that $p \in a_j$, by construction of M_j^*, $G_j(p) \in M_j^*$. If for some $j' > j$ there is some $g \in G_{j'}^*$ such that $G_{j'}(G_j(p)) = g$ and $G_j(p)$ is removed from $M_{j'}^*$ then since $G_{j'}(G_j(p)) = G_{j'}(p)$ and $G_{j'}(p) = g$ is included in $M_{j'}^*$ proceeding inductively it follows that $G_k(p) \in M$ for some $k \in \{j, \ldots, l\}$.

To see that this value of k is unique suppose to the contrary that $G_{k_1}(p)$ and $G_{k_2}(p)$ both lie in M_l^*. Assume without loss of generality that $k_1 < k_2$. Then since $G_{k_2}(G_{k_1}(p)) = G_{k_2}(p)$ we see that $G_{k_1}(p) \notin M_{k_2}^*$ and therefore $G_{k_1}(p) \notin M_l^*$.

2. We see that by construction in every loop $|M_l^*| \leq \left|M_{l-1}^*\right| + N_G$. The bound follows directly.

□

In order to proceed with the cost analysis, we derive bounds on the estimation error for the point histograms PH^l and point sum oracles PSO^l. We will avoid substituting for these error terms until we have reached the end of this analysis but it will be useful to keep in mind that, as the lemma shows, they are roughly $O(\frac{1}{\epsilon}\sqrt{n}\log n)$. These bounds are essentially corollaries of the Bitstogram and HeavySumsOracle error bounds.

Lemma 3.6 (Private estimation error bounds)**.** *With probability $1-\beta$, for all $l = 1, \ldots, L$, suppressing terms logarithmic in $1/\alpha$, $1/\beta$ and $\log n$, the following guarantees hold.*

1. *For every $g \in G_l$, $\left|\mathsf{PH}^i(g) - G_i^{-1}(g)\right| \leq \mathsf{PH}_E := \tilde{O}\left(\frac{1}{\epsilon\alpha}\sqrt{n\log^3 n}\right)$.*
2. *For every $g \in \mathsf{PH}^l$, $\mathsf{PSO}_E \leq \tilde{O}\left(\frac{c_G}{\epsilon}\sqrt{d'n}\right)$.*

Proof. 1. We simplify the Bitstogram guarantee and use PH_M as a uniform upper bound for both PH_E and PH_M. In other words, since PH_M is larger than PH_E, which we show below using the bound on $\log(|G_l|)$, every heavy hitter in PH is already estimated within an error of PH_M. If a value does not occur in PH, it must be the case that its frequency is less than PH_M, so we estimate the frequency of any omitted element by 0 and use the upper bound for PH_M as a uniform bound for the frequency estimates of $g \in G_l$. Similarly, we bound PSO_E by PSO_M.

To derive the expression for the bound we need to bound from above the sizes of the grids G_l. The domain $B(0,1)$ is contained inside the unit cube with side-length 2 units centered at the origin. The length of each axis that lies within this unit cube is 2. For every $g \in G_l$, g_j for every coordinate j can take at most $2/t_l = 2^{L-l}\sqrt{d}/\alpha$ many values. Since the number of dimension is $\tilde{O}((\log k)/\alpha^2)$, it follows that $|G_l| = (2^{L-l}\sqrt{d}/\alpha)^{\tilde{O}((\log k)/\alpha^2)} \Rightarrow \log(|G_l| \cdot 2L/\beta) < \tilde{O}(((\log k)/\alpha^2)\log(n/\alpha)) + \log(2L/\beta)$. Substituting, for any $l = 1, \ldots, L$, $\left|\mathsf{PH}^l(g) - G_l^{-1}(g)\right| \leq \tilde{O}\left(\frac{1}{\epsilon}\sqrt{n((\log k)/\alpha^2)\log(n)\log(2L/\beta)}\right) = \tilde{O}\left(\frac{1}{\epsilon\alpha}\sqrt{n\log^3 n}\right)$ with probability $1 - \beta/2L$.

2. We recall that the diameter of the data domain is $O(1)$. Scaling the failure probability by $1/2L$ so that we may apply the union bound and absorbing the resulting $\sqrt{\log\log n/\beta}$ term in the $\tilde{O}$ notation, the HeavySumsOracle guarantee gives us that $\|\mathsf{PSO}_E\| = \tilde{O}\left(\frac{c_G}{\epsilon}\sqrt{d'n}\right)$.

□

The significance of the following lemma is that $\mathrm{Count}(g)$ is a good estimate of the number of previously uncovered data points covered by a grid point $g \in G_l$ for any $l = 1, \ldots, L$ within a distance of αr_l.

Lemma 3.7. *For $l = 1, \ldots, L$ and any $g \in \mathsf{PH}^l$,*

$$\left|\mathrm{Count}(g) - \left|\{p \in D : G_i(p) = g\} \setminus \left(\cup_{j\in[i]} a_i\right)\right|\right| \leq (l \cdot N_G)\mathsf{PH}_E$$

Proof. By construction of Count, we can write

$$\text{Count}(g) = \mathsf{PH}^l(g) - \sum_{\substack{g' \in M^*_{i-1} \\ G_l(g)=g}} \mathsf{PH}(g'). \tag{1}$$

Similarly, by definition we can write

$$a_l = \{p \in D : G_l(p) = g\} \setminus \left(\cup_{j=1}^{l-1} a_j\right).$$

By lemma 3.5 we see that the sets $\{p \in D : G_j(p) = g', j < l\}$ for $g' \in M^*_{l-1}$ form a partition of $\cup_{j=1}^{l-1} a_j$. Similarly, the sets $\{p \in D : G_j(p) = g' \text{ for some } j\}$ for $g' \in M^*_l$ such that $G_l(g') = g$ form a partition of $\{p \in D : G_l(p) = g\} \cap \left(\cup_{j=1}^{l-1} a_i\right)$. This implies

$$\begin{aligned}
\left|\{p \in D : G_l(p) = g\} \cap \left(\cup_{j=1}^{l-1} a_j\right)\right| &= \sum_{\substack{g' \in M^*_{l-1}, \\ G_l(g')=g}} |\{p \in D : G_j(p) = g' \text{ for some } j\}| \\
\Rightarrow \left|\{p \in D : G_l(p) = g\} \setminus \left(\cup_{j=1}^{l-1} a_i\right)\right| & \\
= |\{p \in D : G_l(p) = g\}| &- \sum_{\substack{g' \in M^*_{l-1}, \\ G_l(g')=g}} |\{p \in D : G_j(p) = g' \text{ for some } j\}|. \qquad (2)
\end{aligned}$$

By the Bitstogram guarantee we have that for every $g' \in \mathsf{PH}^j$, $\left|\mathsf{PH}^j(g') - |\{p \in D : G_j(p) = g'\}|\right| \leq \mathsf{PH}_M$. Subtracting eq. (2) from eq. (1) and using the error bound derived from the Bitstogram guarantee (lemma 3.6) we get

$$\begin{aligned}
\text{Count}(g) - \left|\{p \in D : G_l(p) = g\} \setminus \left(\cup_{j=1}^{l-1} a_j\right)\right| &\leq \mathsf{PH}_M + \sum_{g' \in M^*_{l-1}, G_l(g)=g} \mathsf{PH}_M \\
&\leq O\left(\left|M^*_{l-1}\right| \mathsf{PH}_M\right) \\
&\leq l N_G \mathsf{PH}_M.
\end{aligned}$$

□

The following lemma is used only for the cluster center recovery in the original space but we state and prove it here due to its similarity to lemma 3.7.

Lemma 3.8. *For any $g \in G^*_l$,*

$$\left\| \text{Sum}(g) - \sum_{\substack{G_l(Q(p)) \in G_l^{-1}(g) \\ Q(p) \notin (\cup_{j=1}^{l-1} a_j)}} p \right\| \leq l N_G \mathsf{PSO}_M.$$

Proof. The proof is essentially identical to that of lemma 3.7, but we reproduce the calculations for completeness. In the level l we can write by construction that

$$\text{Sum}(g) = \mathsf{PSO}^i(g) - \sum_{g' \in M^*_{l-1}, G_i(g')=g} \mathsf{PSO}(g').$$

By lemma 3.5 we see that $\{p : Q(p) \in D : G_j(Q(p)) = g', j < l\}$ for $g' \in M^*_{l-1}$ is a partition of $\cup_{j=1}^{l-1} a_j$. We can write these sets more succinctly as $\{p : Q(p) \in G^{-1}(g')\}$ where we can drop the index of the G^{-1} as it depends upon and can be inferred by the argument g'. Continuing, we also have that the sets

$\{p : T(p) \in G^{-1}(g')\}$ for $g' \in M^*_{i-1}, G_l(g') = g$ form a partition of $\{p : T(p) \in G_l^{-1}(g)\} \cap \left(\cup_{j=1}^{l-1} a_j\right)$. This implies

$$\sum_{\substack{Q(p)\in G_l^{-1}(g)\\ \cap\left(\cup_{j=1}^{l-1} a_j\right)}} p = \sum_{\substack{g'\in M^*_{l-1}\\ G_l(g')=g}} \sum_{Q(p)\in G^{-1}(g')} p$$

$$\Rightarrow \sum_{\substack{Q(p)\in G^{-1}(g)\\ \setminus\left(\cup_{j=1}^{l-1} a_i\right)}} p = \sum_{Q(p)\in G_l^{-1}(g)} p - \sum_{\substack{g'\in M^*_{l-1}\\ G_l(g')=g}} \sum_{Q(p)\in G^{-1}(g')} p$$

By the HeavySumsOracle guarantee we have that $\left|\mathsf{PSO}(g') - \sum_{Q(p)\in G^{-1}(g')} p\right| \le \mathsf{PSO}_E \le \mathsf{PSO}_M$. Subtracting the second equation from the first and using the error bound derived from the HeavySumsOracle guarantee (lemma 3.6) we get

$$\left\|\mathrm{Sum}(g) - \sum_{\substack{Q(p)\in G^{-1}(g)\\ \setminus\left(\cup_{j=1}^{l-1} a_j\right)}} p\right\| \le \mathsf{PSO}_M + \sum_{\substack{g'\in M^*_{l-1}\\ G_l(g')=g}} \mathsf{PSO}_M$$

$$\le lN_G\mathsf{PSO}_M.$$

□

Definition 3.9. We let $O_l = \sum_{j=l}^{L} |o_j|$ and $A_l = \sum_{j=l}^{L} |a_j|$. Note that with these definitions,

$$\sum_{l=1}^{L} A_l(r_l - r_{l-1}) = \sum_{l=1}^{L-1} (A_l - A_{l+1})r_l - r_0 A_1$$

$$= \sum_{l=1}^{L} |a_l| r_l. \tag{3}$$

This relation will be useful to us in the cost analysis. Further, we observe that $A_l - O_{l+1} = (n - O_{l+1}) - (n - A_l) = \sum_{j=1}^{l} |o_j| - \sum_{j=1}^{l-1} |a_j|$. Since the o_j are disjoint, it follows that $A_l - O_{l-1}$ is a lower bound for $\left|\cup_{j=1}^{l} o_j \setminus \cup_{j=1}^{l-1} a_j\right|$, the size of the set of points covered by S_{OPT} within a distance of r_l but still uncovered at the beginning of the lth round, i.e. before a_l is picked.

Lemma 3.10. *For $l = 1, \ldots, L$ and for $err = 4lN_G^2\mathsf{PH}_M$,*

$$|a_l| \ge (1 - \alpha)(A_l - O_{l+1}) - err.$$

Proof. From lemma 3.3 we know that $\left|G_l(\cup_{j=1}^{l} o_j))\right| \le N_G$. For all $g \in G_l$, let $\mathrm{Cover}(g) = \{p \in D : G_l(p) = g\} \setminus \cup_{j=1}^{l-1} a_j$, i.e. the set of yet uncovered data points that would be served by g if g were picked. We note that the sets $\mathrm{Cover}(g)$ as defined are disjoint for distinct $g \in G_l$. Let $G_l^\dagger = (g_1^\dagger, \ldots, g_{N_G}^\dagger)$ be the N_G many grid points g with the greatest values of $|\mathrm{Cover}(g)|$ sorted in decreasing order. Then by the observations in definition 3.9 it follows that

$$\left|\{p \in D : G_l(p) \in G_l^\dagger\} \setminus \cup_{j=1}^{l-1} a_j\right| \ge \left|o_l \setminus \cup_{j=1}^{l-1} a_j\right|$$

$$\Rightarrow \sum_{j\in N_G} \left|\mathrm{Cover}(g_j^\dagger)\right| \ge A_l - O_{l+1}.$$

In algorithm 1, we pick grid points greedily via the privatized counts $\mathrm{Count}(g)$. By lemma 3.7 we know that for all $g \in \mathsf{PH}^l$,

$$|\mathrm{Count}(g) - |\mathrm{Cover}(g)|| \le lN_G\mathsf{PH}_M.$$

It follows that if g_j^* is our jth greedy pick maximizing $\text{Count}(g)$ and the maximum value of $|\text{Cover}(g)|$ over all unpicked grid points is $4lN_G\mathsf{PH}_M$, then $\left|\text{Cover}(g_j^*)\right| \geq 4lN_G\mathsf{PH}_M \geq (1/2)\max_{g\in G_i}|\text{Cover}(g)|$.

Let m be the largest index such that $\left|\text{Cover}(g_m^\dagger)\right| \geq 4lN_G\mathsf{PH}_M$. In the context of lemma D.2, we let $Z = \{\text{Cover}(g_j^\dagger) : j \leq m\}$, a family of sets guaranteed to cover $U = \cup_{z\in Z} z$, and $\mathcal{S}$ the family of all possible sets we can pick from $\{\text{Cover}(g) : g \in G_i\}$. Then, since in the jth round each greedy pick g_j^* covers at least half of the maximum that any pick could cover, we see that $(2|N_G|\log(1/\alpha)+1)$ greedy picks cover $(1-\alpha)\left|\cup_{j=1}^m \text{Cover}(g_j^*)\right| \geq (1-\alpha)\left(\left[\sum_{j\in N_G}\left|\text{Cover}(g_j^\dagger)\right|\right] - 4lN_G^2\mathsf{PH}_M\right) \geq (1-\alpha)|o_l| - 4lN_G^2\mathsf{PH}_M$ points, which is what we wanted to show. We use $err = 4lN_G^2\mathsf{PH}_M$ as shorthand going forward. □

Lemma 3.11. *We can relate the optimal clustering cost* OPT *to the sizes of the sets a_l and o_l via the following bounds.*

1. $\sum_{l=0}^{L}|o_l|r_l = O(\text{ OPT }) + O(1)$.
2. $\sum_{l=1}^{L}|a_l|r_l \leq (1+O(\alpha))\sum_{l=1}^{L}|o_l|r_l + O(err)$.
3. $\sum_{l=1}^{L}|a_l|r_l \leq O(\text{ OPT }) + O(err)$.

Proof. 1. Since $r_{l+1} = 2r_l$ for $l > 0$, and $r_1 \leq 1/n$,

$$\begin{aligned}\sum_{l=1}^{L}|o_l|r_l &= \sum_{l=1}^{L}|o_l|4r_{l-1} + |o_0|r_1\\ &\leq 2\sum_{l=1}^{L}|o_l|r_{l-1} + 1\\ &\leq 2\sum_{l=1}^{L}\sum_{p\in o_l} z(p, S_{\text{OPT}}) + 1\\ &\leq 2f_D(S_{\text{OPT}}) + 1.\end{aligned}$$

2. By lemma 3.10,

$$\begin{aligned}|a_l| &\geq (1-\alpha)(A_l - O_{l+1}) - err\\ &\geq (1-\alpha)(|a_l| + A_{l+1} - O_{l+1}) - err\\ &\geq \left(\frac{1-\alpha}{\alpha}\right)(A_{l+1} - O_{l+1}) - \frac{err}{\alpha}\\ \Rightarrow A_{l+1} &\leq \frac{\alpha|a_l|}{1-\alpha} + O_{l+1} + \frac{err}{1-\alpha}\\ &= O(\alpha)|a_l| + O_{l+1} + O(err).\end{aligned}$$

Continuing from eq. (3) and using the convention that a_0 begin undefined is empty,

$$\begin{aligned}\sum_{l=1}^{L}|a_l|r_l &= \sum_{l=1}^{L}A_l(r_l - r_{l-1})\\ &\leq \sum_{l=1}^{L}\left(O(\alpha)a_{l-1} + O_l + O(err)\right)(r_l - r_{l-1})\\ &\leq \sum_{l=1}^{L}O(\alpha)|a_{l-1}|(r_l - r_{l-1}) + \sum_{l=1}^{L}O_l(r_l - r_{l-1}) + O(err)(r_{L+1} - r_0)\end{aligned}$$

$$\leq O(\alpha)\sum_{l=1}^{L}|a_{l-1}|r_{l-1}+\sum_{l=1}^{L}|o_l|r_l+O(err)$$

$$\Rightarrow (1-O(\alpha))\sum_{l=1}^{L}|a_l|r_l\leq\sum_{l=1}^{L}|o_l|r_l+O(err)$$

$$\Rightarrow \sum_{l=1}^{L}|a_l|r_l\leq(1+O(\alpha))\sum_{l=1}^{L}|o_l|r_l+O(err).$$

3. This is a direct consequence of the first and second results of this lemma.

□

Lemma 3.12. *The k-means clustering functions for the dimension reduced dataset D and the proxy dataset D^* are close in ℓ_1 norm. Concretely, for any finite set C,*

$$f_{D^*}(C)\leq(1+O(\alpha))f_D(C)+O(\alpha\,\mathrm{OPT}\,)+O(\alpha err)+O(L^2N_G^2\mathsf{PH}_M),$$
$$f_D(C)\leq(1+O(\alpha))f_{D^*}(C)+O(\alpha\,\mathrm{OPT}\,)+O(\alpha err)+O(L^2N_G^2\mathsf{PH}_M).$$

Proof. We can write $D^*=\sqcup_{l=1}^{L}\sqcup_{g\in G_l^*}\{g \text{ with multiplicity } \mathrm{Count}(g)\}$. Then it follows that

$$\begin{aligned}
f_{D^*}(C)&=\sum_{l=1}^{L}\sum_{g\in G_l^*}\mathrm{Count}(g)f_g(C)\\
&=\sum_{l=1}^{L}\sum_{g\in G_l^*}\mathrm{Count}(g)z(g,C)\\
&\leq\sum_{l=1}^{L}\sum_{g\in G_l^*}\left(\left|a_l\cap G_l^{-1}(g)\right|+lN_G\mathsf{PH}_M\right)z(g,C)\\
&\leq\left(\sum_{l=1}^{L}\sum_{p\in a_l}z(G_l(p),C)\right)+O(L^2N_G^2\mathsf{PH}_M)\\
&\leq\left(\sum_{l=1}^{L}\sum_{p\in a_l}z(G_l(p),\arg\min_{s\in C}z(p,s))\right)+O(L^2N_G^2\mathsf{PH}_M)
\end{aligned}$$

To bound $z(G_l(p),\arg\min_{s\in C}z(p,s))$, we use the AM-GM inequality in conjunction with the triangle inequality for the ℓ_2 norm as follows:

$$\begin{aligned}
z(G_l(p),\arg\min_{s\in C}z(p,s))&\leq(\sqrt{z(G_l(p),p)}+\sqrt{z(p,C)})^2\\
&\leq\alpha^2r_l^2+2\alpha r_l\sqrt{z(p,C)}+z(p,C)\\
&\leq\alpha^2r_l^2+2r_l(\sqrt{\alpha}\cdot\sqrt{\alpha z(p,C)})+z(p,C)\\
&\leq\alpha^2r_l^2+\alpha r_l+\alpha r_lz(p,C)+z(p,C)\\
&\leq O(\alpha r_l)+(1+O(\alpha))z(p,C)
\end{aligned}$$

Applying this bound for every point $p\in a_l$ for all $l=1,\ldots,L$ we get

$$\begin{aligned}
f_{D'}(C)&=(1+O(\alpha))f_D(C)+O(\alpha)\sum_{l=1}^{L}|a_l|r_l+O(L^2N_G^2\mathsf{PH}_M)\\
&=(1+O(\alpha))f_D(C)+O(\alpha\,\mathrm{OPT}\,)+O(\alpha err)+O(L^2N_G^2\mathsf{PH}_M).
\end{aligned}$$

Similarly,

$$\begin{aligned}
f_D(C) &= \sum_{p\in D} z(p,C) \\
&= \sum_{l=1}^{L}\sum_{p\in a_l} z(p, \arg\min_{s\in C} z(G_l(p),s)) \\
&= \sum_{l=1}^{L}\sum_{p\in a_l} (1+O(\alpha))z(p,C) + O(\alpha)r_l \\
&\le \left[\sum_{l=1}^{L}\sum_{p\in a_l} (1+O(\alpha))z(G_l(p),C)\right] + \left[\sum_{l=1}^{L} O(\alpha)|a_l|r_l\right] \\
&\le \left[\sum_{l=1}^{L}\sum_{g\in G_l^*} (\mathrm{Count}(g) + lN_G\mathsf{PH}_M)(1+O(\alpha))z(G_l(p),C)\right] + O(\alpha\,\mathrm{OPT}) + O(\alpha err) \\
&\le \left[\sum_{l=1}^{L}\sum_{g\in G_l^*} \mathrm{Count}(g) z(G_l(p),C)\right] + O(L^2N_G^2\mathsf{PH}_M) + O(\alpha\,\mathrm{OPT}) + O(\alpha err) \\
&\le f_{D'}(C) + O(L^2N_G^2\mathsf{PH}_M) + O(\alpha\,\mathrm{OPT}) + O(\alpha err).
\end{aligned}$$

□

Corollary 3.13. *As a direct consequence of lemma 3.12, it follows that*

$$f_D(S^*) \le (1+O(\alpha))\eta\,\mathrm{OPT} + O(\alpha err) + O(L^2N_G^2\mathsf{PH}_M),$$

where we absorb the η factor in the additive error terms in the big-Oh notation and an $O(\alpha\,\mathrm{OPT})$ term in the first term.

We now want to recover the cluster centers of the clusters derived from the low-dimension space by using the PSO derived from calls to $\mathsf{HeavySumsOracle}$. Since we identify points by their images in level-wise grids, we incur additional discretization error that must be accounted for. Concretely, the clustering we actually derive is not $p \mapsto \arg\min_{s'\in S'} z(T(p),s')$ but instead given by the following definition.

Definition 3.14. 1. Let $G'_* : \mathbb{R}^{d'} \to \left(\cup_{l=1}^{L} G_l\right)$ denote $G_l \circ Q(p')$ where l is the minimum index such that $G_l \circ Q(p') \in G_l^*$. We then define a clustering of D' via the solution S^* by letting $D'(s_i^*) = \{p' \in D' : \arg\min_{s\in S^*} z(s, G'_*(p')) = s_i^*\}$. Alternatively, we can first define $G_{l,i}^* = \{g \in G_l^* : \arg\min_{s\in S^*} z(g,s) = s_i^*\}$, then let $D'_l(s_i^*) = \{p' \in D' : G_l \circ Q \in G_{l,i}^* \text{ and } G_j \circ Q \notin G_j^* \text{ for } j < l\}$ and then let $D'(s_i^*) = \cup_{l\in[L]} D'_l(s_i^*)$; these two formulations are equivalent.

2. We see that with these definitions $a_l = Q(\sqcup_{s_i^*\in S^*} D'_l(s_i^*))$. Further, this also defines a clustering of D by identifying each point in D' with its dimension-reduced image in D, with clustering cost

$$\sum_{l=1}^{L}\sum_{p\in a_l} z(G_l(p), S^*) = \sum_{p'\in D'} z(G'_*(p'), S^*).$$

Lemma 3.15. *For the privately derived cluster centers S^* in the dimension reduced space, we have the following bound for the clustering of D as defined in definition 3.14.*

$$\sum_{l=1}^{L}\sum_{p\in a_l} z(G_l(p), S^*) = \eta(1+O(\alpha))\,\mathrm{OPT} + O(\alpha err) + O(N_G^2\mathsf{PH}_M \log^2 n).$$

As a direct corollary,

$$\sum_{p' \in D'} z(G'_*(p), S^*) = \eta(1 + O(\alpha))\,\mathrm{OPT} + O(\alpha err) + O(N_G^2 \mathsf{PH}_M \log^2 n).$$

Proof. We want to understand the increase in clustering cost due to discretization.

$$\sum_{l=1}^{L} \sum_{p \in a_l} z(G_l(p), S^*) - z(p, S^*) \le \sum_{l=1}^{L} \sum_{p \in a_l} z(G_l(p), \arg\min_{s \in S^*} z(p, s)) - z(p, S^*).$$

Here we use the same trick of applying the ℓ_2-triangle inequality in conjunction with the A.M.-G.M. inequality as in the proof of lemma 3.12 and bound $z(G_l(p), \arg\min_{s \in S^*} z(p, s))$ from above by $O(\alpha r_i) + (1 + O(\alpha))z(p, S^*)$. Continuing,

$$\begin{aligned}
\sum_{l \in 1}^{L} \sum_{p \in a_l} z(G_l(p), S^*) - z(p, S^*) &\le \sum_{i=1}^{L} \sum_{p \in a_l} O(\alpha r_l) + \sum_{i=1}^{L} \sum_{p \in a_l} O(\alpha) z(p, C^*) \\
&\le O(\alpha) \sum_{l=1}^{L} |a_l| r_l + \sum_{i=1}^{L} \sum_{p \in a_l} O(\alpha) z(p, S^*) \\
&\le O(\alpha\,\mathrm{OPT}) + O(\alpha err) + O(\alpha) f_D(S^*).
\end{aligned}$$

Since we use a non-private clustering algorithm with multiplicative approximation factor η, we can substitute for $f_D(S^*)$ by $\eta\,\mathrm{OPT}$, and rearranging terms we get the stated bound. □

We now use the error bounds for the sum oracle and the succinct histogram to recover the cluster centers of the cluster as defined in definition 3.14.

Lemma 3.16. *For every cluster center $s_i^* \in S^*$, we have the following estimation error bound for the cluster centers of the clusters derived in the original space.*

$$\left\| \frac{\sum_{l=1}^{L} \sum_{g \in G_{l,i}^*} \mathrm{Sum}(g)}{\sum_{l=1}^{L} \sum_{g \in G_{l,i}^*} \mathrm{Count}(g)} - \frac{\sum_{p' \in D'(s_i^*)} p'}{|D'(s_i^*)|} \right\| \le \frac{2L^2 N_G^2}{|D'(s_i^*)|} \left(\left\| \frac{\sum_{p' \in D'(s_i^*)} p'}{|D'(s_i^*)|} \right\| \mathsf{PH}_M + \mathsf{PSO}_M \right).$$

Proof. The proof of this result is essentially the same as that of lemma B.1, with the additional complication that we must account for the error accrued when summing over queries for multiple heavy values.

$$\begin{aligned}
&\frac{\sum_{l=1}^{L} \sum_{g \in G_{l,i}^*} \mathrm{Sum}(g)}{\sum_{l=1}^{L} \sum_{g \in G_{l,i}^*} \mathrm{Count}(g)} - \frac{\sum_{p' \in D'(s_i^*)} p'}{|D'(s_i^*)|} \\
=&\frac{\sum_{l=1}^{L} \sum_{g \in G_{l,i}^*} \mathrm{Sum}(g)}{\sum_{l=1}^{L} \sum_{g \in G_{l,i}^*} \mathrm{Count}(g)} - \frac{\sum_{p' \in D'(s_i^*)} p'}{\sum_{l=1}^{L} \sum_{g \in G_{l,i}^*} \mathrm{Count}(g)} + \frac{\sum_{p' \in D'(s_i^*)} p'}{\sum_{l=1}^{L} \sum_{g \in G_{l,i}^*} \mathrm{Count}(g)} - \frac{\sum_{p' \in D'(s_i^*)} p'}{|D'(s_i^*)|} \\
=&\frac{\sum_{l=1}^{L} \sum_{g \in G_{l,i}^*} \mathrm{Sum}(g) - \sum_{p' \in D'(s_i^*)} p'}{\sum_{l=1}^{L} \sum_{g \in G_{l,i}^*} \mathrm{Count}(g)} + \frac{\sum_{p' \in D'(s_i^*)} p'}{|D'(s_i^*)|} \frac{|D'(s_i^*)| - \sum_{l=1}^{L} \sum_{g \in G_{l,i}^*} \mathrm{Count}(g)}{\sum_{l=1}^{L} \sum_{g \in G_{l,i}^*} \mathrm{Count}(g)} \\
\Rightarrow&\left\| \frac{\sum_{l=1}^{L} \sum_{g \in G_{l,i}^*} \mathrm{Sum}(g)}{\sum_{l=1}^{L} \sum_{g \in G_{l,i}^*} \mathrm{Count}(g)} - \frac{\sum_{p' \in D'(s_i^*)} p'}{|D'(s_i^*)|} \right\| \\
\le&\frac{L \cdot N_G \cdot LN_G \mathsf{PSO}_M}{|D'(s_i^*)| - L \cdot N_G \cdot LN_G \mathsf{PH}_M} + \frac{\sum_{p' \in D'(s_i^*)} p'}{|D'(s_i^*)|} \cdot \frac{L \cdot N_G \cdot LN_G \mathsf{PH}_M}{|D'(s_i^*)| - L \cdot N_G \cdot LN_G \mathsf{PH}_M}
\end{aligned}$$

So for all clusters $D'(s_i^*)$ such that with at least $2L^2N_G^2\mathsf{PH}_M$ many points, we can bound the ℓ_2 estimation error by

$$\frac{2L^2N_G^2\mathsf{PSO}_M}{|D'(s_i^*)|} + \left\|\frac{\sum_{p'\in D'(s_i^*)} p'}{|D'(s_i^*)|}\right\| \cdot \frac{2L^2N_G^2\mathsf{PH}_M}{|D'(s_i^*)|}$$
$$= \frac{2L^2N_G^2}{|D'(s_i^*)|}\left(\left\|\frac{\sum_{p'\in D'(s_i^*)} p'}{|D'(s_i^*)|}\right\|\mathsf{PH}_M + \mathsf{PSO}_M\right).$$

□

Now we can derive the cost bound for the private clustering solution derived in the original space.

Lemma 3.17.

$$f_{D'}(S') \le (1+O(\alpha))\eta\,\mathrm{OPT}' + O(\alpha^2 err\log n/\beta) + O(\alpha L^2N_G^2\mathsf{PH}_M\log n/\beta) + O(kL^2N_G^2\mathsf{PSO}_M).$$

Proof. We are interested in bounding the clustering cost of D' with respect to the clusters $(D'(s_1^*),\ldots,D'(s_k^*))$. In lemma 3.15 we bounded the cost of the dimension reduced image of this clustering $(D(s_1),\ldots,D(s_k)) = (Q(D'(s_1^*)),\ldots,Q(D'(s_k^*)))$. From lemma 3.2 we recall that for any clustering $(D_1',\ldots,D_k')$ of D we have that

$$\sum_{i\in k}\sum_{p\in D_i'} s\left(p, \frac{\sum_{q\in D_i'} q}{|D_i'|}\right) \simeq_{1+\alpha} (\alpha\log n/\beta)\sum_{i\in k}\sum_{p\in Q(D_i')} s\left(Q(p), \frac{\sum_{q\in D_i'} M(q)}{|D_i'|}\right).$$

If we let $D_i' = D'(s_i^*)$, and denote the (unknown) true cluster centers of the clusters in the original space by $\mu_i = \frac{\sum_{p'\in D'(s_i^*)} p'}{|D'(s_i^*)|}$ for $i = 1,\ldots,k$ then we get

$$f_{D'}(\{\mu_1,\ldots,\mu_k\}) \simeq_{1+\alpha} (\alpha\log n/\beta) f_D(\{s_1^*,\ldots,s_k^*\})$$
$$\simeq_{1+\alpha} \alpha\log n/\beta(1+O(\alpha))\eta\,\mathrm{OPT} + O(\alpha^2 err\log n/\beta) + O(\alpha L^2N_G^2\mathsf{PH}_M\log n/\beta).$$

Then, since $\mathrm{OPT}' \simeq_{1+\alpha} \alpha\log n/\beta\,\mathrm{OPT}$, we can write

$$f_{D'}(\{\mu_1,\ldots,\mu_k\}) \simeq_{1+O(\alpha)} (1+O(\alpha))\eta\,\mathrm{OPT}' + O(\alpha^2 err\log n/\beta) + O(\alpha L^2N_G^2\mathsf{PH}_M\log n/\beta).$$

We have estimates

$$\hat{\mu_i} = \frac{\sum_{l=1}^{L}\sum_{g\in G_i^*(s_i^*)}\mathrm{Sum}(g)}{\sum_{l=1}^{L}\sum_{g\in G_i^*(s_i^*)}\mathrm{Count}(g)}$$

for the true cluster centers μ_i for $i = 1,\ldots,k$. From lemma D.3, in order to bound the additive error incurred due to the estimation error, i.e. $f_{D'(s_i^*)}(\{\hat{\mu}_i\}) - f_{D'(s_i^*)}(\{\mu_i\})$, it will suffice to bound $|D'(s_i)^*|\|\mu_i - \hat{\mu}_i\|^2$. Lemma 3.16 bounds the estimation error $\|\hat{\mu} - \mu\|$. Putting everything together, we get

$$f_{D'(s_i^*)}(\{\hat{\mu}_1,\ldots,\hat{\mu}_k\}) - f_{D'(s_i^*)}(\{\mu_i\}) \le |D'(s_i^*)|\left(\frac{2L^2N_G^2}{|D'(s_i^*)|}\left(\left\|\frac{\sum_{p'\in D'(s_i^*)} p'}{|D'(s_i^*)|}\right\|\mathsf{PH}_M + \mathsf{PSO}_M\right)\right)^2$$
$$\le \frac{8L^4N_G^4\mathsf{PH}_M^2}{|D'(s_i^*)|} + \frac{8L^4N_G^4\mathsf{PSO}_M^2}{|D'(s_i^*)|}$$
$$\le \frac{L^4N_G^4}{|D'(s_i^*)|}O(\mathsf{PSO}_M^2).$$

For each $s_i^* \in S^*$, if $D'(s_i^*) \geq L^2 N_G^2 \mathsf{PSO}_M$, then the first factor is $O(L^2 N_G^2 \mathsf{PSO}_M)$. On the other hand, if $D(s_i^*) < L^2 N_G^2 \mathsf{PSO}_M$ then the clustering cost for $D'(s_i^*)$ i.e. $f_{D'(s_i)}(\{\hat{\mu}_1, \ldots, \hat{\mu}_k\})$ is unconditionally less than $L^2 N_G^2 \mathsf{PSO}_M$ as the diameter of the data domain is $O(1)$. It follows that the additive error over all k clusters is at most $O(kL^2 N_G^2 \mathsf{PSO})$. Since $f_{D'(s_i^*)}(\{\mu_i\}) = f_{D'(s_i^*)}(\{\hat{\mu}_1, \ldots, \hat{\mu}_k\})$, putting everything together we get that

$$\begin{aligned} f_{D'}(\{\hat{\mu}_1, \ldots, \hat{\mu}_k\}) &\leq f_{D'}(\{\mu_1, \ldots, \mu_k\}) + O(kL^2 N_G^2 \mathsf{PSO}_M) \\ &\leq (1 + O(\alpha))\eta \operatorname{OPT}' + O(\alpha^2 err \log n/\beta) + O(\alpha L^2 N_G^2 \mathsf{PH}_M \log n/\beta) + O(kL^2 N_G^2 \mathsf{PSO}_M). \end{aligned}$$

□

Theorem 1.1. *Algorithm 1 is an (ϵ, δ)-locally differentially private algorithm that after one round of interaction with a private distributed data set $D' \subset \mathbb{R}^{d'}$ of size n, outputs a set S' of size k such that for failure probability polynomially small in n,*

$$f_{D'}(S') \leq (1 + O(\alpha))\eta \operatorname{OPT}' + \frac{1}{\epsilon} k^{\tilde{O}(1/\alpha^2)} \sqrt{d' n \log 1/\delta} \operatorname{poly} \log n.$$

Proof. We make $2L = 2 \log n$ calls (in parallel) to Bitstogram and HeavySumsOracle. From their respective privacy guarantees, we know that each call is (ϵ, δ)-differentially private. By simple composition of privacy, it follows that the net privacy loss is $(2(\log n)\epsilon, 2(\log n)\delta)$. To ensure net (ϵ, δ) privacy loss, we must scale the respective privacy parameters by a factor of $1/(2 \log n)$; with this scaling we have $\mathsf{PH}_M = \tilde{O}\left(\frac{1}{\epsilon\alpha}\sqrt{n \log^5 n}\right)$ and $\mathsf{PSO}_M = \tilde{O}\left(\frac{c_G}{\epsilon}\sqrt{d' n \log^2 n}\right)$. We recall that $N_G = k^{\tilde{O}(1/\alpha^2)}$. Substituting all these bounds in the guarantee of lemma 3.17 we get

$$\begin{aligned} f_{D'}(\{\hat{\mu}_1, \ldots, \hat{\mu}_k\}) &\leq (1 + O(\alpha))\eta \operatorname{OPT}' + O(\alpha^2 err \log n/\beta) + O(\alpha L^2 N_G^2 \mathsf{PH}_M \log n/\beta) + O(kL^2 N_G^2 \mathsf{PSO}_M) \\ &\leq (1 + O(\alpha))\eta \operatorname{OPT}' + \frac{1}{\epsilon} k^{\tilde{O}(1/\alpha^2)} \sqrt{d' n \log 1/\delta} \operatorname{poly} \log n. \end{aligned}$$

To simplify the error term in the above expression we assume without loss that $k \geq 2$, as $k = 1$ is a degenerate case i.e. mean estimation of vectors in d' dimensional space. We then absorb all constants in the $\tilde{O}$ term in the exponent of the k to state a simplified bound. □

4 LDP k-means with low additive error

In this section we describe our second algorithm that, given a constant $c > \sqrt{2}$, can achieve a constant factor multiplicative approximation and $O(k^{O(1/(2c^2-1))}\sqrt{nd'} \operatorname{poly} \log n)$ additive error. Our algorithm is described in a modular fashion, and one may refer to the respective section for the pseudo-code and an informal walk-through of how the algorithm proceeds. We begin with a technical discussion to help explain some of the algorithmic choices made along the way.

4.1 Technical discussion

We recall from the introduction that any differentially private solution for k-means clustering in the local setting has to somehow indirectly access the aggregate geometry of the data set because of the high magnitude of the noise that is added to maintain privacy. We then discussed how discretizing the response function that is sensitive to the location of each point allows us to do precisely this and understand the geometry of the data set in sum. The one-round clustering algorithm uses a grid-based discretization of the domain to elicit a discrete response. For our four-round algorithm, we will use a combination of a cell-based discretization (which is similar in essence to the grid-based discretization used before) in combination with LSH functions.

Notation	Meaning
$D' \subset \mathbb{R}^{d'}$	Original data set
$Q : \mathbb{R}^{d'} \to \mathbb{R}^d$	mapping from high-dim. to low-dim. space
$D \subset \mathbb{R}^d$	$Q(D')$, dimension reduced data set
L	Number of cell grid levels
$\mathcal{C}_l$	Grid of cells in dimension reduced space for $l \in [L]$
t_l	Side-length of any cell in $\mathcal{C}_l$ (equals 2^{-l})
$\mathcal{C}_l(\cdot)$	Mapping from $\mathbb{R}^d$ to unique containing cell $\mathcal{C}_l$
$\text{Anc}^* : \mathcal{C}_l \to \mathcal{C}_{l-(3/2)\lg d}$	Mapping from cells to the set of their ancestors j with side-length $d^{3/2}t_l$
CH^l	Succinct histogram of number of points mapping to $C \in \mathcal{C}_l$ for "heavy" C
F	Number of geometrically varying guesses for true optimal clusternig cost OPT
$\mathcal{H}_l^f$	Heavy cells identified CH^l where guess for OPT $= k\sqrt{n} \cdot 2^f$, $f \in [F]$
$\mathcal{L}_l^f$	Cells which are not heavy
$\mathcal{M}_l^f$	Light children of heavy cells
M	Number of distance scales with which LSH functions applied
$r_{l,1}, \ldots, r_{l,M}$	Scales at which LSH functions are used to allocate cluster centers for points in $\mathcal{M}_l$
R	Number of repetitions of LSH subroutine to boost success probability
Λ_l^f	Synthetic space of heavy cells in Anc^* level
$\Lambda_l^f(\cdot)$	Mapping from $\mathbb{R}^d$ to synthetic space
$H_{l,m,r,f}(\cdot)$	$(p(1), p(c), r_{l,m}, cr_{l,m})$-sensitive hash function with domain Λ_l^f for points in $\mathcal{M}_l^f$
$\mathsf{BH}_{l,m,r,f}$	Histogram of number of points per hash bucket
$\mathsf{BSO}_{l,m,r,f}$	Vector sums of points in original space mapping to heavy buckets
$\hat{b}$	Average vector mapping to bucket $b \in \mathsf{BH}_{l,m,r,f}$
$\Pi_l(\hat{b})$	projection of $\hat{b}$ to $\Lambda_l^f(\cup_{C \in \mathcal{H}_l^f} C)$
S_l	Candidate centers allocated in one level for some guess of OPT
$S_{\mathcal{H}}$	Candidate centers allocated at the center of heavy cells for some guess of OPT
S	k-means bi-criteria solution

Table 3: Summary of notation used in algorithm 4

Dyadic hierarchy of cells: In Braverman et al. [2017], the authors describe a way of decomposing the data domain in a way that helps identify regions of the domain where data accumulates. Given a rectangular domain $[0,1)^d$, they construct a dyadic 2^d-ary tree of cells, where each rectangular cell is sub-divided into 2^d child cells by bisecting the cell along each axis. The cell at the top of the hierarchy with side-length one unit is simply the whole domain, and it has 2^d children with side-length half units that collectively again cover the whole domain. Each child cell is recursively divided in the same manner, and in level l the side length of each cell is $t_l = 2^{-l}$ units. The idea is that although each point in the domain is covered by each level of cells, the further down the hierarchy one goes the finer is the resolution at which the domain is discretized and the smaller is the diameter of the bounding box at a level. $L = \log n$ levels of the grid will suffice to discretize the domain to a sufficiently fine degree so as to capture clusters at all relevant scales; the cost of clustering cluster with radius smaller than $O(1/n)$ will be dominated by the additive error terms that any private k-means clustering algorithm must have. This entire construction can be done after an application of the JL transform for dimension reduction which ensures that $d = O(\log n)$. We will see during the course of our discussion why this is crucial for our cost analysis.

The authors of Braverman et al. [2017] then observe that if we *randomly shift* this hierarchy of cells, then one can show that with probability $1 - \beta$, for any point in the domain there are at most $O(1/\beta)$ many cells with side-length t_l within an ℓ_2 distance of t_l/d units of that point. Applying this on any choice of optimal centers S_{OPT} means that there are $O(k/\beta)$ many cells close to S_{OPT} at any level. How can we exploit this to capture the aggregate geometry of the data set?

Guessing the optimal cost: Suppose that we knew what the optimal cost OPT were. If this were the case, then we can bound the number of cells further than t_l/d units away from any choice of optimal centers S_{OPT} that carry significantly many data points. Concretely, all data points in cells further than t_l/d units away from S_{OPT} must have a clustering cost of at least t_l^2/d^2. On the other hand, their clustering cost with respect to S_{OPT} cannot exceed the total clustering cost OPT, which means there cannot be more than $\mathrm{OPT}\, d^2/t_l^2$ many such points. Tracing a similar argument with cells, we compute a threshold depending on the level's side length t_l such that there cannot be more then $O(kL/\beta)$ many cells that have more than the number of points in the threshold and lie further than t_l/d units away from S_{OPT}. To see why the bound has changed from $O(k/\beta)$ to $O(kL/\beta)$, note that we scale the failure probability by a factor of $1/L$ so that it apply across all L levels with probability $1 - \beta$. Coupled with the guarantee that there cannot be more than $O(kL/\beta)$ many cells closer than t_l/d to S_{OPT} we get that regardless of where they lie in the domain there are at most $O(kL/\beta)$ *heavy* cells in any level, i.e. cells that beat the threshold T_l for their level.

In the top cell, this threshold is lower than n, so the top cell is always marked heavy. In the bottom level, this threshold exceeds n, so all bottom cells are marked *light* (i.e. not heavy). Between these two extremes the threshold increases monotonically as t_l decreases, which means that there is a unique level for every point where the cell it belongs to transitions from being heavy to light. There is a small technicality here that since we can only identify cell counts via noisy privatized responses we can inadvertently mark heavy cells light and light cells heavy. In practice we will appeal to the locally private histogram construction Bitstogram of Bassily et al. [2020] to estimate the data point counts of cells. The issue of incorrect labelling of cells as heavy or light is readily resolved by requiring that heavy cells have only heavy ancestors, and using the accuracy guarantees of Bitstogram to bound the consequences of such errors in our cost analysis.

In sum, under the promise that OPT is known, we have identified regions of the domain at different scales where the data set accumulates beyond some thresholds. Since we are targeting an additive error of $\tilde{O}(k\sqrt{n})$, we let OPT vary in factors of 2 from $k\sqrt{n}$ to n and simply run the algorithm with varying values of OPT at different scales to ensure that the promise holds for at least some run. This leads to an inflation in our additive error on the order of $\log n$ as the number of candidate centers grows by this factor.

We recall that in the introduction we mentioned that when finding a bi-criteria solution, to get $O(\mathrm{poly}\, k\sqrt{n}\, \mathrm{poly}\log n)$ error we would like to find $O(\mathrm{poly}\, k\, \mathrm{poly}\log n)$ many candidate centers with respect to which the data set has a clustering cost within a constant factor multiplicative approximation to OPT and additive error at most $O(\mathrm{poly}\, k\sqrt{n}\, \mathrm{poly}\log n)$. It is in fact the case that if we can limit the exponent of k in both the number of centers allocated and the additive error incurred, then we will have at most that same exponent

in the final error term. Keeping this in mind we observe that we have partitioned the data set across $O(k \log^2 n/\beta)$ many heavy cells. If we can allocate some $O(\text{poly} \log n)$ centers in each cell such that the additive error with respect to these centers is $O(k\sqrt{n}\,\text{poly} \log n)$, then we would achieve $O(k\sqrt{n}\,\text{poly} \log n)$ error in sum. Although we do not achieve exactly this term, the reason we are able to get arbitrarily close to it is because of the relatively small number of cells within which we have partitioned the data set. We turn to using LSH functions to allocate candidate centers in a cell-wise fashion.

The $n^{1/2+a}$ barrier: We recall that a (p, q, r, cr) LSH function has the property that if two points are within a distance of r units, they must collide with probability at least p, and if two points are further than cr units, then they cannot collide with probability more than q. By applying LSH functions on the data domain and appealing to locally private succinct histograms, we can recover all heavy LSH buckets; the idea then is that any sufficiently large cluster with radius less than r units must populate one of these heavy buckets with a lot of points, possibly with some false positives. We estimate the point average over each heavy bucket to get a point that is no more than cr units away from the cluster, and serves as a cluster center with a constant factor approximation to the true radius.

We now describe why prior work taking this approach suffer an $O(n^{1/2+a}\,\text{poly} \log n)$ dependence on n in the additive error. When dealing with LSH functions one technicality that has to be dealt with is that the LSH guarantee holds only in a pair-wise fashion, i.e. you only get bounds on the likelihood of points colliding a pair of points at a time. Fixing some cluster C with radius r, this leads us to use some arbitrary fixed point from C as a filter, using the LSH guarantee to argue that (1) "most" points which collide with it under the LSH function with parameters (p, q, r, cr) must lie at a distance of at most cr units and (2) for every cluster, at least a p fraction of points from that cluster must collide with it. What we would like to be the case is that the average over all points colliding with our filter lie at a distance of $O(cr)$ from the filter; since the filter itself lies in the cluster, by the triangle inequality the average can then serve as a candidate center for the cluster with an $O(c)$ constant factor approximation to the radius.

Let Δ be the diameter of the data domain. The distance of the weighted mean of all points colliding with the filter under the LSH function, from that filter, can roughly be bounded from above by

$$cr \cdot |\{\text{points from } C \text{ colliding with filter}\}| + \Delta \cdot |\{\text{points further than } cr \text{ units from filter}\}|$$

We are bounding the impact of points from outside the cluster by the diameter of the domain, and dealing with the arbitrarily many points that lie between a distance of r and cr units by simply inflating the distance considered "close" to cr units so they can be dropped from consideration without giving us an unfair advantage (notice that they can only pull this average towards cr units). It is easy to see by linearity of expectation that the expected number of points from the cluster that collide with the filter is at least $p|C|$, and the number of points from further than cr units that collide with the filter is at most $q|D|$ (again using the worst case as an upper bound).

To get this weighted mean to be of the order of cr, we tune the LSH parameters to get the collision probability ratios to fulfill

$$cr \geq \frac{q|D|\Delta}{p|C|}.$$

One can see this as a tug of war between false positives which in expectation increase with the side of the data set and whose impact is exacerbated by the diameter of the data domain and "true" cluster points whose impact can be as low as cr units and whose number scales with the size of the cluster C. Rearranging terms gives us

$$\frac{p}{q} \geq \frac{|D|}{|C|}\frac{\Delta}{cr}.$$

It follows that if one needs this procedure to work for clusters C with as few as $\sqrt{n}$ many points, as well as for cluster radii that are a $\text{poly}(n)$ factor smaller than Δ, then since $|D| = n$, one would need the ratio between the collision probabilities of near and far points to beat a $\text{poly}(n)$ term.

It is an intrinsic property of LSH functions that tuning parameters to increase the ratio between p and q causes both p and q to fall individually. This is an issue because we also need sufficiently many points from the cluster to accumulate in a bucket to ensure we can distinguish the heavy bucket from random noise; in expectation the number of true points accumulating in a bucket number drops with p. It is in fact the case that p scales with $n^{-\Theta(1/c)}$ which leads us to try and boost the success probability with $n^{\Theta(1/c)}$ many independent runs. Since we cannot test which runs are successful and which are not, we are forced to include all bucket averages generated along the way as candidate centers; this $n^{\Theta(1/c)}$ factor in the number of centers is what leads to the greater than $1/2$ exponent of n that is incurred in previous work applying LSH functions for clustering as discussed in the introduction. We can push this exponent arbitrarily close to $1/2$ by letting $c \to \infty$, but naturally this causes the multiplicative approximation guarantee to blow up.

Even if we were to somehow reduce the number of possible false positives (i.e. the size of the data set D that lies in the LSH domain) from n to something that scales with the cluster size, there is still the issue that Δ/cr could again be $\text{poly}(n)$. We must find a way to both limit the sizes of the subset of the data that participate in the LSH procedure as well as the diameter of the data domain within which that subset can lie. We describe how we achieve exactly this in the sequel.

If we apply this LSH subroutine heavy cell by heavy cell, then the impact of any point from more than $O(cr)$ units can be at most the diameter of the cell, i.e. $2^{-l}\sqrt{d}$ in the lth level, which resolves the $n^{1/2+a}$ issue for all LSH scales which are

$$\Omega\left(\frac{2^{-l}}{\text{poly}\log n}\right).$$

However, there are still two issues to be resolved. We have yet to bound the size of the data subset lying in the LSH domain, as we discussed is necessary. Further, if the lowest LSH scale is still $2^{-l}/\sqrt{n}$ (for example), then the ratio of collision probabilities still has a factor of n, which will lead to an exponent of n greater than $1/2$, as described above. In order to get a truly $O(\sqrt{n}\,\text{poly}\log n)$ term, we need to increase the smallest cut-off distance for the set of LSH scale parameters.

Limiting the sequence of LSH scale parameters: We first take a small detour and describe how a finite sequence of scale parameters is chosen for cluster radii when identifying a bi-criteria solution. The analysis fixes some arbitrary optimal clustering solution S_{OPT} and decomposes the data set using concentric rings around S_{OPT} at geometrically varying thresholds. More concretely, each partition of the data set consists of points which lie between 2^{-l} and 2^{-l+1} units for $l = 1, 2, \ldots$. The goal then is to allocate cluster centers so that for each partition we can derive the promise that most points are covered by some candidate center at a distance of $O(2^{-l})$. Since the optimal clustering distance was at least 2^{-l} units per partition, this would give us a bi-criteria solution with an $O(\,\text{OPT}\,)$ cost.

One typically tries to identify these partitions and allocate centers separately for each partition, but doing so requires that there be a finite (and in fact small) set of distance thresholds and partitions. One way of accomplishing this is to cut off the sequence of thresholds at $\log n$ and instead of promising a constant multiplicative approximation to the optimal clustering distance for points which lie closer than $2^{-\log n} = 1/n$ units to OPT, one observes that as long as there is a candidate center at a distance of $O(1/n)$, the net clustering cost for the at most n such points there could be is $O(n \cdot 1/n^2) = o(1)$. The cost of clustering such points is then treated as a small additive error term in the constant factor approximation guarantee.

When using LSH at a sequence of geometrically varying scales, one runs into a similar issue of needing to identify a lower bound for the smallest distance at which we allocate candidate centers. If the smallest such scale is t units, then as there could be as many as n points within this distance we will need $t^2 n$ units to be dominated by $O(k\sqrt{n})$, which would require t to scale with $O(1/\sqrt{n})$ in the case where k is small. As discussed, we need to avoid a $1/\,\text{poly}(n)$ scaling factor for the lowest threshold t so as to avoid an exponent of n greater than $1/2$; it follows that the only way to do this is to reduce the size of the data subset on which LSH being applied. Essentially, this issue has been reduced to other condition which we needed to fulfill; that of bounding the size of the data subset participating in the LSH subroutine.

Bounding the subset of D participating in LSH subroutines: We see that simply using LSH on heavy cells does not work as is since there could again be arbitrarily many points in a heavy cell; all we have is a lower bound on the number of points it contains. To derive an upper bound we instead focus on the *light children of heavy cells*. By virtue of being light, they have fewer points than the threshold mentioned before; we will be able to show that in level l where the side length $t_l = 2^{-l}$ the total number of points which lie in such cells is $O(d^2 \operatorname{OPT}/t_l^2)$. From the previous discussion, this will allow us to set the lower LSH scale parameter $t = O(t_l/(d\sqrt{L}))$ and incur only $O(\operatorname{OPT}/L)$ additional error per level, leading to an additional $O(\operatorname{OPT})$ cost across all levels. Observe that the lowest LSH scale parameter is essentially $t_l/\operatorname{poly}\log n$, which implies a $\operatorname{poly}\log n$ ratio between the diameter of the cell to the scale parameter, which is exactly what we wanted. The additional $O(\operatorname{OPT})$ additive error term is readily absorbed in our multiplicative approximation factor (as opposed to a small additive error as is usually the case). Since the dimension d and the number of levels in our cell hierarchy are both $O(\log n)$, this means that we have successfully avoided an exponent greater than $1/2$ on the factor of n in the additive error.

However, there is a different sort of issue in the dyadic hierarchy approach that we have not yet addressed; for any level the collection of light children of heavy cells partitions the data in arbitrary ways. It need not be the case that a cluster will lie entirely inside the domain of a single LSH function when making calls to the LSH subroutine. How do we account for the division of clusters across data partitions and cells?

Clusters and cluster sections: Let us denote the partition of the data set D that lies in heavy cells in level $l-1$ but light cells in level l by D_l. With this notation it follows from our observations regarding the existence of a unique level for each data point such that its containing cell is light for the first time when going down levels that $D_0, \ldots, D_{L-1}$ form a partition of D. For any fixed optimal clustering solution, we see that each cluster too can be partitioned across all levels D_l. Based on the discussion above, we would ideally like to use LSH functions on $O(kL/\beta)$ many cells in level $l-1$ and elicit a response only from D_l to ensure that the diameter of the bounding box is not too high and the number of points participating in the LSH subroutine is not too many. This implies that we only need to allocate cluster competitively with respect to the sections of the optimal clusters that lie in heavy cells. However, this could lead to $O(k^2L/\beta)$ many cluster sections per level, which would lead to a candidate center set of size at least $\Omega(k^2 \operatorname{poly}\log n)$, leading to $\Omega(k^2\sqrt{n}\operatorname{poly}\log n)$ error down the line. In order to try and reduce the exponent of k in the number of cluster centers allocated, we make three technical algorithmic choices.

Firstly, we allocate a candidate center at the center of every heavy cells (which would be at most $O(kL^2/\beta)$ many more candidate centers). This gives us the guarantee that every point in the data set partition D_l has a candidate center at a distance of $2^{-l}\sqrt{d}$. Secondly, we go up a few levels and apply LSH functions to the ancestors of these heavy cells of interest which have side-length $d^{3/2}2^{-l}$. The consequence of these two modifications is that we only need to allocate cluster centers within a distance of $2^{-l}\sqrt{d}$ units of any point of D_l, and that since there are only $O(L/\beta)$ many cells with side-length $d^{3/2}2^{-l}$ within a distance of $2^{-l}\sqrt{d}$ units of an optimal center, there are only $O(kL/\beta)$ many cluster sections we must account for.

Thirdly, in order to avoid dealing with the worst case $O(k)$ many cluster sections for every heavy cell when calling the LSH subroutine heavy cell by heavy cell, we construct a synthetic space out of the union of all heavy cells in a level and apply the LSH subroutine on this entire space. We will be able to extend the ℓ_2 metric in a natural way to work across cells, ensure that the cells are far enough apart in this distance measure so that bucket averages that land up "between" cells end up in the correct cell after projection, and that the diameter of this synthetic space is still small enough to keep the improvements we have derived so far.

There is one final technical point which must be addressed; we need to identify a lower bound on the cluster section size to ensure that the ratio of the participating subset of the data to the size of the cluster section does not grow to $\operatorname{poly}(n)$, which would lose us the advances we have made. Since there are at most $O(kL/\beta)$ many such cluster sections in a level, we simply set the threshold to be $\operatorname{OPT} \cdot \frac{\beta}{kL} \cdot \frac{1}{L} \cdot \frac{1}{dt_l^2}$. Why does this work? We recall that we allocated a cluster center at the center of every heavy cell, that ensures that any cluster section has a candidate center at a distance of $\sqrt{d}t_l$, so for a cluster section below the threshold the cluster cost can be at most $\operatorname{OPT} \cdot \frac{\beta}{kL} \cdot \frac{1}{L}$. Then, since there are at most $O(kL/\beta)$ many such

cluster sections, the net clustering cost for any one level across all cluster sections is $\mathrm{OPT} \cdot \frac{1}{L}$. Summing this up over all L levels leads to an additional OPT term which again we can absorb into our constant factor approximation.

We can summarize this lower bound on the cluster section size as $O(\frac{\mathrm{OPT}}{kt_l^2 \operatorname{poly}\log n})$. We recall that the size of the participating data set D_l was at most $O(\mathrm{OPT}/t_l^2)$, which implies a ratio of $k \operatorname{poly}\log n$. A dependence on n in the ratio that the LSH collision probabilities have to beat has been replaced by a dependence on k, leading to a $O\left(k^{1+O(1/2c^2-1)}\sqrt{n}\operatorname{poly}\log n\right)$ bound on the number of candidate centers allocated.

Constructing the proxy data set and undoing dimension reduction: In the one-round algorithm, we constructed a set of candidate centers and undid the dimension reduction in essentially one round of interaction. However, doing everything in one round increases the exponent of k; this was not apparent in that analysis unless studies it carefully since the big-Oh term in the power of k in the number of candidate centers dominated any similar order increases (such as being squared) in the big-Oh notation. Since our goal in this section is to keep the error as low as possible, we avoid reducing the round complexity and instead use two rounds of interaction; one to construct the proxy data set, and one to recover the cluster centers in the original space.

The construction of the proxy data set is relatively straightforward; we release the collection of candidate centers found and invite agents to privately reveal which candidate center is closest to them. Again by an appeal to Bitstogram, we estimate the number of data points a candidate center serves and construct a proxy data set by repeating each candidate center with multiplicity equal to its respective estimate. We then apply the non-private clustering algorithm of our choice on the privately generated proxy data set to get cluster centers $S^* = \{s_1^*, \ldots, s_k^*\}$ in the dimension reduced space.

In the final round of interaction we reveal the set S^*, and we ask agents to privately reveal a k-tuple of d'-dimensional vector where the ith vector equals its true location if s_i^* is its closest cluster center in the dimension reduced space, and is otherwise the 0 vector. In the same round of interaction, we ask them to reveal which is the center closest to them. We then simply compute the noisy sum for each of the k coordinates and divide that by the noisy count of the number of points mapping to the center corresponding to that coordinate; we will be able to show that this estimate for the cluster center in the original space works well in its place for a k-means clustering solution.

Outline: We divide the description and technical analysis of this algorithm into 4 parts. In subsection 4.2 we formally describe the dyadic hierarchy of cells needed to construct the algorithm. In subsection 4.3, we use the identification of heavy and light cells in the previous subsection to partition the data set level-by-level. Fixing any level, we prove that for any fixed optimal clustering, with probability $1-\beta$ we allocate candidate centers for most points in the partition corresponding to that level at an ℓ_2^2 distance at most $O(c^2)$ times their distance from the optimal centers. In subsection 4.4 we use the guarantees of subsection 4.3 to show that the sum-of-squares cost of clustering the dimension reduced dataset via the candidate centers is $O(\mathrm{OPT})$ modulo some additive error. We go on to show by applications of the weak triangle inequality that the clustering functions of the proxy dataset and the dimension reduced dataset are close in ℓ_2^2 distance up to an $O(\mathrm{OPT})$ additive error. Then we bound the cost of the original dataset with respect to cluster centers derived via the dimension reduced clustering and account for the privacy loss to derive our net cost guarantee.

4.2 The cell grids and their hierarchy

In this subsection we formally define the cell grid hierarchy used to allocate candidate centers in the next section and describe an algorithm that uses succinct histogram of cell counts to tag cells as being either *heavy* or *light*. Apart from the definitions made, the main results of this subsection are lemma 4.8 which guarantees lower and upper bounds for the number of data points that can lie in heavy and light cells respectively; and lemma 4.10, which shows how we can use the identification of heavy and light cells to partition the data set D, one partition per level, to get the subsets D_l for $l \in [L]$.

Data: For every level $l \in [L]$, a succinct histogram of heavy-hitter cells CH^l with error bound CH^l_E and maximum frequency omitted CH^l_M.

$\mathcal{H}_l \leftarrow \emptyset$
$\mathcal{L}_l \leftarrow \emptyset$
end
$\mathcal{H}_0 \leftarrow \mathcal{C}_0$
for $i = l \in \{2, \ldots, L-1\}$ **do**
  for $C \in \mathsf{CH}^l$ **do**
    if $\mathsf{CH}^l(C) \geq \frac{\beta d^2 \,\mathrm{OPT}}{t_l^2 kLd} + \mathsf{CH}^l_E$ *and* $\mathrm{Anc}_1(C_j) \in \mathcal{H}_{l-1}$ **then**
      $\mathcal{H}_l \leftarrow \mathcal{H}_l \cup \{C_j\}$
    else
      $\mathcal{L}_l \leftarrow \mathcal{L}_l \cup \{C_j\}$
  end
  $\mathcal{L}_l \leftarrow \mathcal{L}_l \cup \mathcal{C}_l \backslash \mathcal{H}_l$
end
$\mathcal{L}_{L-1} \leftarrow \mathcal{C}_{L-1}$
for $l \in [L]$ **do**
  $\mathcal{M}_l \leftarrow \{C \in \mathcal{L}_l : \mathrm{Anc}_1(C) \in \mathcal{H}_{l-1}\}$
end
return $\{\mathcal{H}_l, \mathcal{L}_l, \mathcal{M}_l : l \in [L]\}$

Algorithm 2: Heavy cell marker

We work over the domain $[0,1)^d$. We start by dividing this domain recursively in a dyadic fashion, with $L = \lceil \lg n \rceil$ levels in all.

Definition 4.1. We formalize the construction of the dyadic hierarchy of cells.

1. A *cell* is a dyadic cube in $(0,1]^d$. Explicitly, if we let the set of cells at level l be denoted $\mathcal{C}_l$; then

$$\mathcal{C}_l := \left\{ \prod_{e=1}^{d} \left[\frac{j_e}{2^l}, \frac{j_e + 1}{2^l} \right) : j \in \{0, 1, \ldots, 2^l - 1\}^d \right\}.$$

   We also define the notation $\mathcal{C} := \cup_i \mathcal{C}_i$.

2. We let $t_l = 2^{-l}$ for $l \in [L]$; with this notation, every $C \in \mathcal{C}_l$ has side-length t_l. Note that with these definitions the minimum side-length $t_L \leq \frac{1}{n}$.

3. For all $l \in [L]$, $\mathrm{Anc}_i : 2^{\mathcal{C}} \to 2^{\mathcal{C}}$ and $\mathrm{Ch}_i : 2^{\mathcal{C}} \to 2^{\mathcal{C}}$ are defined by the following expressions:

$$\text{for } \mathcal{C}' \subset \mathcal{C}_l,\ \mathrm{Anc}_i(\mathcal{C}') = \{C \in \mathcal{C}_{l-i} : C' \subset C \text{ for some } C' \in \mathcal{C}'\},$$
$$\text{for } \mathcal{C}' \subset \mathcal{C}_l,\ \mathrm{Ch}_i(\mathcal{C}') = \{C \in \mathcal{C}_{l+i} : C \subset C' \text{ for some } C' \in \mathcal{C}'\}.$$

   We set $\mathcal{C}_i = \{[0,1)^d\}$ for $i < 0$; with this definition seeking the ancestor at a level above 0 always returns the entire domain. It will not be necessary to define cells below level L. We abuse notation so that any singleton set of cells is identified with the element in it; with this notation we also have $\mathrm{Ch}_l : \mathcal{C} \to 2^{\mathcal{C}}$ and $\mathrm{Anc}_l : \mathcal{C} \to \mathcal{C}$.

4. $\mathrm{Anc}^* := \mathrm{Anc}_{1.5 \lg d}$. Note that for $C \in \mathcal{C}_l$, $\mathrm{Anc}^*(C)$ is the unique cell that contains C and has side length $d^{3/2} t_l$.

We recall the following lemma from Braverman et al. [2017]:

Lemma 4.2 (Lemma 2.2, Braverman et al. [2017]). *Let S be a finite set of points and $\mathcal{C}_R$ be a d-dimensional rectangular grid of cells with unit length R shifted by a uniformly random displacement in $[0, R]$ along each dimension. With probability at least $1 - \beta$, $\left|\{C \in \mathcal{C}_R : z(S, C) \leq R^2/d^2\}\right| = O(|S|/\beta)$.*

Proof. The number of cells with side-length R within an ℓ_2 distance of $x < R/2$ from any $s \in S$ can be bounded from above by $N_1 N_2 \dots N_d$ where N_i is 1 if there is no cell wall within a distance of s along the ith dimension and 2 otherwise. Since the random shifts along each dimension are independent, it follows that

$$\mathbb{E}\left[\prod_{i=1}^{d} N_i\right] = \prod_{i=1}^{d} \mathbb{E}[N_i].$$

The probability of s_i lying within a distance of x units of one of the sides is $(2x/R)$. It follows that

$$\begin{aligned}\mathbb{E}[N_i] &= \left(1 - \frac{2x}{R}\right) \cdot 1 + \frac{2x}{R} \cdot 2 \\ &= 1 + \frac{2x}{R}.\end{aligned}$$

Substituting in the first display, we get

$$\Rightarrow \mathbb{E}\left[\prod_{i=1}^{d} N_i\right] = \left(1 + \frac{2x}{R}\right)^d.$$

It follows that for $x \leq \frac{R}{d}$, the expected number of cells within an ℓ_2 distance of x from s is at most $O(1)$. By linearity of expectation, the expected number of cells within a distance of R/d of S is at most $O(|S|)$. By Markov's inequality, with probability $1 - \beta$, the number of cells within a ℓ_2 distance of R/d from S is $\leq O(|S|/\beta)$. The result follows directly as by definition $z(\cdot, \cdot)$ is the ℓ_2^2 distance. □

Remark 4.3. We fix any arbitrary optimal k-means clustering solution S_{OPT} with clustering cost $\leq$ OPT and condition on the event that the number of cells in any level l within a distance of t_l/d from S_{OPT} is at most $O(kL/\beta)$. By scaling the failure probability in lemma 4.2 by a factor of $\frac{1}{L}$ and applying the union bound over L such events (one for each level) we see that this event holds with probability $1 - \beta$. We will also assume that $\text{OPT} \geq k\sqrt{n}$. Note that if $\text{OPT} < k\sqrt{n}$ then for any choice of S_{OPT} $f_D(S_{\text{OPT}}) \leq k\sqrt{n}$. Once we obtain an $O(1)$ multiplicative approximation to $k\sqrt{n}$, this term can be absorbed by the additive error term, so the guarantee as stated will hold unconditionally.

We partition the $\mathcal{C}_l$ into collections of *heavy* and *light* cells at every level depending on the number of data points within each cell. We perform this partitioning algorithmically via algorithm 2.

Definition 4.4. $C \in \mathcal{C}$ is called *heavy* if $C \in \mathcal{H}_l$ for some $l \in [L]$ where $\mathcal{H}_l \subset \mathcal{C}_l$ is defined by the output of algorithm 2. Similarly, $C \in \mathcal{C}_l$ is called *light* if $C \in \mathcal{L}_l$ for some $l \in [L]$ for $\mathcal{L}_l$ defined by the output of algorithm 2. We summarize the notation of algorithm 2 for cells and collections of cells here:

1. We denote the set of heavy cells at level l by $\mathcal{H}_l$, and the set of light cells by $\mathcal{L}_l$.
2. We denote the collection of all heavy cells by $\mathcal{H} := \cup_l \mathcal{H}_l$ and the collection of all light cells by $\mathcal{L} := \cup_l \mathcal{L}_l$.
3. The center of any cell C is denoted by $o(C)$ (this may be thought of as the *origin* of C).
4. The cell at level l containing $p \in D$ is denoted by $\mathcal{C}_l(p)$.

We summarize some basic properties of heavy and light cells in lemma 4.5 as a sanity check.

Lemma 4.5. *The following statements hold:*

1. $\forall l \in [L], \mathcal{C}_l = \mathcal{H}_l \sqcup \mathcal{L}_l$.

2. *If* $C \in \mathcal{L}$, *then* $\mathrm{Ch}_l(C) \subset \mathcal{L}\ \forall l \geq 0$.

3. $\mathcal{H}_0 = \mathcal{C}_0 = \{[0,1)^d\}$

4. $\mathcal{L}_{L-1} = \mathcal{C}_{L-1}$.

Proof. 1. This statement follows from the algorithm description - for every level $< L$ a cell recovered from Bitstogram is marked either heavy or light, and all other cells in that level are marked light. For level L all cells are marked light.

2. This statement holds because a cell is marked heavy only if its parent was marked heavy in the previous iteration.

3. This statement holds by line 5 of algorithm 2.

4. This statement holds by line 15 of algorithm 2.

□

The definition of the succinct cell count histograms CH^l occurs later in this section in algorithm 4. Its properties follow entirely from the Bitstogram guarantee and the definition of the value mapping. Since it is used in algorithm 2 and is necessary in the analysis of algorithm 2, we will state and prove them here.

Corollary 4.6. $\mathsf{CH}^l_E = O\left(\frac{1}{\epsilon_{\mathsf{CH}}}\sqrt{n \log n/\beta}\right)$ *and* $\mathsf{CH}^l_M = O\left(\frac{1}{\epsilon_{\mathsf{CH}}}\sqrt{n \operatorname{poly}\log n/\beta}\right)$.

Proof. Fix any $l \in [L]$. Looking ahead, we see that CH^l is derived from a call to Bitstogram on line 9 of algorithm 4 with mapping $f_l : p \mapsto C_l(p)$, privacy parameter ϵ_{CH}, and failure probability β/L. We note that the size of the co-domain for the mapping f_l is at most 2^{dL}. Since $d, L = O(\log n)$, substituting we get the stated bounds. □

Note that since $|V| = 2^{dL} = \Omega(n)$, we can bound $\mathsf{CH}^l_E = O(\mathsf{CH}^l_M)$.

Remark 4.7. We recall that the significance of corollary 4.6 is that the Bitstogram guarantee gives us that with probability $1-\beta$, for every $C \in \mathcal{C}_l$ such that $|D \cap C| \geq \mathsf{CH}^l_M$, $C \in \mathsf{CH}^l$ and $\left|\mathsf{CH}^l(C) - |D \cap C|\right| \leq \mathsf{CH}^l_M$.

In lemma 4.8 we characterize the accumulation of data in heavy and light cells across different levels.

Lemma 4.8. *For all* $l \in [L]$, *the following properties hold:*

1. *If* $C \in \mathcal{H}_l$, $|D \cap C| \geq \max\left(\mathsf{CH}^l_M, \frac{\beta\,\mathrm{OPT}}{t_l^2 kLd}\right)$.

2. *If* $C \in \mathcal{L}_l$, $|D \cap C| < \min\left(\mathsf{CH}^l_M, \frac{\beta\,\mathrm{OPT}}{t_l^2 kLd} + 2\mathsf{CH}^l_M\right) = \mathsf{CH}^l_M$.

3. $|\mathcal{H}_l| \leq O\left(\frac{kL}{\beta}\right)$.

Proof. 1. If a cell $C \in \mathcal{C}_l$ is marked heavy, then it must have occurred in the histogram CH^l and so $|D \cap C| \geq \mathsf{CH}^l_M$, or it is the solitary top cell. In the former case, since the count estimate crossed the threshold to be considered heavy, $|D \cap C| \geq \frac{\beta\,\mathrm{OPT}}{t_l^2 kLd} + \mathsf{CH}^l_E - \mathsf{CH}^l_E = \frac{\beta\,\mathrm{OPT}}{t_l^2 kLd}$. In the latter case, substituting $l = 0$ we see that the desired lower bound is $\frac{\mathrm{OPT}}{kLd}$. Since OPT can be at most n, and $|D \cap C| = n$, the bound holds.

2. If a cell C is marked light then either $|D \cap C| < \mathsf{CH}^l_M$, $\mathrm{Anc}_1(C) \in \mathcal{L}_{l-1}$, or it is a bottom level cell. In the first three cases, by induction down the levels l, since $|D \cap C| \subset |D \cap \mathrm{Anc}_1(C)|$ and both $\frac{\beta d^2\,\mathrm{OPT}}{t_l^2 kLd}$ and CH^l_M are monotonically increasing with l, the result follows by the induction hypothesis. Note that since the top cell is always marked heavy, the base case is vacuously true. In the last case, we substitute $l = \log n - 1$ to get $\mathsf{CH}^l_M \geq \frac{\beta\,\mathrm{OPT}}{t_l^2 kLd} \geq \frac{\beta n^2 k\sqrt{n}}{kLd} \geq \beta n^{2.5}/\log^2 n$, which is asymptotically impossible for failure probability $\beta \geq \frac{1}{n^2}$ and certainly for $\beta = \Theta(1)$.

3. We fix any optimal k-means solution S_{OPT}. By remark 4.3, $\left|\{C \in \mathcal{C}_l : z(C, S_{\mathrm{OPT}}) \leq 1/(4^l d^2)\}\right| = O(kL/\beta)$. For any $C \in \mathcal{H}_l$ such that $z(C, S_{\mathrm{OPT}}) > 1/(4^l d^2)$, from statement 1 we have that $|C \cap D| = \max\left(\frac{\beta\,\mathrm{OPT}}{t_l^2 kLd}, \mathsf{CH}_M^l\right) \geq \frac{\beta d^2\,\mathrm{OPT}}{t_l^2 kL}$. It follows that $f_{C\cap D}(S_{\mathrm{OPT}}) > \frac{t_l^2}{d^2} \cdot \frac{\beta d^2\,\mathrm{OPT}}{t_l^2 kL} = \frac{\beta\,\mathrm{OPT}}{kL}$. Since $\sum_{C\in\mathcal{H}_l} f_{C\cap D}(S_{\mathrm{OPT}}) < f_D(S_{\mathrm{OPT}}) \leq \mathrm{OPT}$ it follows that there cannot be more than $\frac{kL}{\beta}$ many such C. In sum, $|\mathcal{H}_l| \leq O\left(\frac{kL}{\beta}\right)$.

□

We now define a decomposition of the data set D using the definitions of the heavy and light cells.

Definition 4.9. For $l \in [L]$, we define $D_l = \{p \in D : \mathcal{C}_{l-1}(p) \in \mathcal{H}, \mathcal{C}_l(p) \in \mathcal{L}\}$.

Lemma 4.10. *The following statements hold.*

1. $D = \sqcup_{l\in[L]} D_l$.
2. $\forall l \in [L],\ |D_l| = O(d^2\,\mathrm{OPT}/t_l^2) + O\left(\frac{kL\mathsf{CH}_M}{\beta}\right)$.
3. $\sum_{l\in[L]} \frac{1}{4^l d^2 L}|D_l| = O(\mathrm{OPT}) + O\left(\frac{kL\mathsf{CH}_M}{\beta}\right)$.

Proof. 1. By lemma 4.5, we see that the solitary top cell $[0,1)^d$ is heavy, and that $\mathcal{C}_L \subset \mathcal{L}$. Further, for every $l \in [L]$, if $C \in \mathcal{L}_l$ then $\mathrm{Ch}_j(C) \in \mathcal{L}$. It follow that for any point p, in the sequence of cells $\mathcal{C}_0(p), \mathcal{C}_1(p), \ldots, \mathcal{C}_L(p)$ there exists a unique index l^* such that $\mathcal{C}_{l^*-1}(p) \in \mathcal{H}$ and $\mathcal{C}_{l^*}(p) \in \mathcal{L}$. The existence of such an index shows that the sets D_l cover D, and the uniqueness shows that this is in fact a partition.

2. By definition, D_l is a subset of $\cup_{C\in\mathcal{L}_l} C$, which means we can write $D_l = \cup_{C\in\mathcal{L}_l} D \cap C$. This union can in turn be written as a disjoint union of points in light cells at a distance $\leq t_l/d$ from S_{OPT} and points in light cells $> t_l/d$ away from S_{OPT}. From lemma 4.8, any light cell contains at most CH_M many points of D. Since there are at most $O(kL/\beta)$ cells with side length t_l within a distance of t_l/d, it follows that there are at most $\mathsf{CH}_M \cdot O\left(\frac{kL}{\beta}\right)$ many points within a distance of t_l/d from S_{OPT}. Since the total clustering cost for S_{OPT} must equal OPT, there can be at most $d^2\,\mathrm{OPT}\,\frac{2}{l}$ many points more than t_l/d away from S_{OPT}. Therefore in sum $|D_l| \leq O(d^2\,\mathrm{OPT}/t_l^2) + O\left(\frac{kL\mathsf{CH}_M}{\beta}\right)$.

3. The second bound follows directly from the first.

□

4.3 Candidate center allocation

We begin by giving a brief overview of the main steps in algorithm 4.

Step 1 - Initialization and first interaction: We start by setting up the dimension reduction, scaling and projection map Q. We then have our first round of interaction with the agents where we make L calls to Bitstogram in parallel to receive estimates of how many points lie in each cell. We then make geometrically varying guesses for OPT $k\sqrt{n}2^f$ for $f \in [F]$ where $F = \log_2 \frac{n}{\sqrt{nk}}$; note that with this definition our guesses vary in powers of two from $k\sqrt{n}$ to n. For each guess we generate a marking of cells $\{\mathcal{H}_l^f, \mathcal{L}_l^f, \mathcal{M}_l^f\}$ by calls to algorithm 2, where $\mathcal{M}_l^f$ (think *medium*) is notation of convenience to denote light cells with heavy parents. Note that D_l defined earlier is precisely the set of data points which happen to lie in cells in $\mathcal{M}_l^f$ in level l.

Data: Guess for OPT $= k\sqrt{n} \cdot 2^f$, Cell labels $(\mathcal{H}_l, \mathcal{M}_l, \mathcal{L}_l)$, Bucket histogram $\mathsf{BH}_{l,m,r}$, Bucket Sum Oracle $\mathsf{BSO}_{l,m,r}$ for $l \in [L]$, $m \in [M]$, $r \in [R]$,

Data drawn from global variables: number of levels L, number of LSH scales M, number of repetitions R

$S_{\mathcal{H}} \leftarrow \{o(C) : \exists i C \in \mathcal{H}_i\}$

$T_l = \frac{p(1)}{2} \cdot \max\left(\frac{\beta \,\mathrm{OPT}}{t_l^2 k L^2 d}, \frac{4\mathsf{BH}_M}{p(1)}, O\left(\frac{c_G\sqrt{n \operatorname{poly}\log n/\beta}}{\epsilon_{\mathsf{BSO}}}\right)\right)$; /* Bucket threshold */

$S_l \leftarrow \emptyset$

for $l \in [L]$, $m \in [M]$, $r \in [R]$ **do**

  for $(b, \hat{n}_b) \in \mathsf{BH}_{l,m,r}$ *such that* $\hat{n}_B \geq T_l$ **do**

    $\hat{b} \leftarrow \frac{\mathsf{BSO}_{l,m,r}(b)}{\mathsf{BH}_{l,m,r}(b)}$

    $\Pi_l(\hat{b}) \leftarrow$ project $\hat{b}$ to $\Lambda_l^f(\cup_{C \in \mathcal{H}_l^f} C)$

    $S_l \leftarrow S_l \cup \{\hat{b}\}$

  end

end

return $S_{\mathcal{H}} \cup \bigcup_{l=1}^{L} S_l$

Algorithm 3: Candidate Center Allocation for k-Means in Dimension-Reduced Space given OPT

Step 2 - Candidate center allocation and second interaction We start by defining M, the number of LSH scales, and R the number of independent repetitions of the hashing subroutine to boost the success probability. We then define a mapping $\Lambda_l^f : \mathbb{R}^d \to \mathbb{R}^{\mathcal{H}_l^f} \times \mathbb{R}^d$ where the co-domain is a synthetic space mimicking the union of all heavy cells upon which we can define our LSH functions. Concretely, for points p such that $\mathrm{Anc}^*(\mathcal{C}_l(p))$ is a heavy cell, the image is a 2-tuple of a $|\mathcal{H}_{l-1.5\lg d}|$ length indicator vector indicating which heavy ancestor cell p lies in, and the p's position with respect to the center of its ancestor cell. Finally we construct the mapping $H_{l,m,r,f}$ which computes the output of a $(p(1), p(c), r_{l,m}, cr_{l,m})$ hash function if the point p lies in D_l (which is true if and only if $\mathcal{C}_l(p) \in \mathcal{M}_l$) and a null bucket value otherwise (i.e. no participation). The heavy buckets of these hash functions are privately recovered via calls to Bitstogram and we also recover the sums of all vectors mapping to heavy buckets via a call to HeavySumsOracle, the consequence succinct histogram and sum oracle are $\mathsf{BH}_{l,m,r,f}$ and $\mathsf{BSO}_{l,m,r,f}$ respectively.

For every guess for OPT we pass these histograms and oracles to algorithm 3, which allocates a candidate center $\Pi_l(\hat{b})$ for every heavy bucket b whose count $\hat{n}_b$ crosses a certain threshold T_l. The location at which it allocates that actual center is found by querying the oracle for the sum of all points mapping to this bucket to get a value $\mathsf{BSO}^{l,m,r,f}(b)$ and dividing this vector sum by the histogram count $\mathsf{BH}^{l,m,r,f}(b) = \hat{n}_b$; this is an estimate of the average over all points mapping to this bucket. We then project this average to the space Λ_l to get the point $\Pi_l(\hat{b})$. Algorithm 3 also allocates a candidate center at the center of every heavy cell. It then returns all centers found to the calling function algorithm 4 which stores the centers passed by the call with the guess $k\sqrt{n} \cdot 2^f$ for OPT in S_f. The net bi-criteria solution then is simply $S = \cup_{f \in [F]} S^f$ which it passes to the center recovery algorithm algorithm 5 along with the dimension reduction and random shift mapping Q.

The main results of this section are lemma 4.21, which allows us to derive a guarantee and lemma 4.23, which bounds the total number of candidate centers allocated.

Definition 4.11. We record some notation that will be convenient to use in the course of our analysis.

1. We denote the data set in the original space $\mathbb{R}^{d'}$ by D' and in the dimension reduced space $\mathbb{R}^d$ (after scaling, projection, and translation) by $D = M(D')$.

2. We let the optimal clustering cost in the original space be denote OPT', and the dimension reduced optimal clustering cost be denoted OPT.

Setting: Distributed dataset $D' \subset \mathbb{R}^{d'}$ over n agents

/* Step 1: Initialization and first interaction */

$\gamma \leftarrow$ uniformly random vector in $[-1/2, 1/2]^d$

$T : \mathbb{R}^{d'} \to \mathbb{R}^d$ dimension reduction for $d = O(\log(k/\alpha\beta)/\alpha^2)$

$S : \mathbb{R}^d \to \mathbb{R}^d$ scaling by a factor $\frac{1}{2(1+\alpha)}$

$P : \mathbb{R}^d \to B(0, 1/2)$ projection to $B(0, 1/2)$ followed by translation by γ

$Q = P \circ S \circ T : \mathbb{R}^{d'} \to B(0, 1) \subset \mathbb{R}^d$

$L = \lg n$

Do in parallel:

  $\mathsf{CH}^l \leftarrow \mathsf{Bitstogram}(\mathcal{C}_l \circ Q(\cdot), \epsilon_{\mathsf{CH}}, \beta/L)$; /* Cell-wise Histogram of points */

end

$F = \log_2 \frac{n}{\sqrt{n}k}$; /* Exponent of 2 in guess for OPT */

for $f \in [F]$ **do**

  $\{\mathcal{H}_i^f, \mathcal{L}_i^f, \mathcal{M}_i^f : i \in [L]\} \leftarrow \mathsf{HeavyCellMarker}(\{\mathsf{CH}^l : l \in [L]\}, \text{guess for OPT} = k\sqrt{n} \cdot 2^f)$

end

/* Step 2: Candidate center allocation and second interaction */

$M = 1 + \log_2 d^{3/2}\sqrt{L} = O(\log \log n)$; /* Number of LSH scales */

$r_{l,m} = \frac{2^m t_l}{d\sqrt{L}}$ for $m \in [M]$; /* LSH scales */

$R = O\left(\frac{\log kL^2/\beta}{p(1)}\right)$; /* Number of repetitions for LSH */

$\lambda_l = (14c + 5)t_l\sqrt{d}$

$\Lambda_l^f(\cdot) := p \mapsto \begin{cases} (\lambda_l 1_{\mathrm{Anc}^*(C(p))}, p - o(C_l(p))) \text{ if } p \in \cup_{C \in \mathrm{Anc}^*(\mathcal{H}^f)} C \\ 0 \text{ otherwise} \end{cases}$; /* Mapping to LSH domain */

$$H_{l,r,m,f}(p) = \begin{cases} (p(1), p(c), r_{l,m}, cr_{l,m})\text{-sensitive Hash function on the space } \Lambda_l^f \text{ if } \mathcal{C}_l(p) \in \mathcal{M}_l \\ \perp \text{ otherwise} \end{cases}$$

Do in parallel for $f \in [F], l \in [L], m \in [M], r \in [R]$**:**

  $\mathsf{BH}_{l,m,r,f} \leftarrow \mathsf{Bitstogram}(H^f_{l,m,r}, \beta, \epsilon_{\mathsf{BH}})$; /* Bucket-wise Histogram of points */

  $\mathsf{BSO}_{l,m,r,f} \leftarrow \mathsf{HeavySumsOracle}(H_{l,m,r,f}, \Lambda_l, \beta, \epsilon_{\mathsf{BSO}})$; /* Bucket Sum Oracle */

end

$S \leftarrow \emptyset$

for $f \in [F]$ **do**

  $S^f \leftarrow$ algorithm 3 (Guess for OPT $= k\sqrt{n} \cdot 2^f, \{\mathcal{H}_i^f, \mathcal{L}_i^f, \mathcal{M}_i^f : i \in [L]\}, \mathsf{BH}_{l,m,r,f}, \mathsf{BSO}_{l,m,r,f}$)

  $S \leftarrow S \cup S^f$

end

return (algorithm $5(S, Q)$)

Algorithm 4: LDP k-means with low additive error

3. We fix an arbitrary optimal solution S_{OPT} in the dimension reduced space; we will show that our allocation of candidate centers competes well with S_{OPT}. Note that in particular $f_D(S_{\mathrm{OPT}}) = \mathrm{OPT}$.

Lemma 4.12. *With probability $1-\beta$, we have that for every clustering $(D'_1, \dots, D'_k)$ of D',*

$$\sum_{i\in k}\sum_{p\in D'_i} s\left(p, \frac{\sum_{q\in D'_i} q}{|D'_i|}\right) \simeq_{1+O(\alpha)} \sum_{i\in k}\sum_{p\in M(D'_i)} s\left(Q(p), \frac{\sum_{q\in D'_i} Q(q)}{|D'_i|}\right).$$

Further, with this notation $\mathrm{OPT} = \Theta(\mathrm{OPT}')$ and $D = M(D') \subset [0,1)^d$.

Proof. We write $Q = P \circ S \circ T$, where T is the dimension reduction to $O(\log(k/\alpha\beta)/\alpha^2)$, S is the scaling by a factor of $1/2(1+\alpha)$, and P is projection to the unit ball. Given any clustering $(D_1, \dots, D_k)$ of D, by theorem 2.7 we have that

$$\sum_{i\in k}\sum_{p\in D_i} s\left(p, \frac{\sum_{q\in D_i} q}{|D_i|}\right) \simeq_{1+\alpha} \sum_{i\in k}\sum_{p\in D_i} s\left(T(p), \frac{\sum_{q\in D_i} T(q)}{|D_i|}\right).$$

The scaling map changes all ℓ_2-distances by precisely the scaling factor, so we also have that

$$\sum_{i\in k}\sum_{p\in D_i} s\left(T(p), \frac{\sum_{q\in D_i} T(q)}{|D_i|}\right) = \frac{1}{(1+\alpha)^2}\sum_{i\in k}\sum_{p\in S\circ T D_i} s\left(S\circ T(p), \frac{\sum_{q\in D_i} S\circ T(q)}{|D_i|}\right).$$

Since with probability $1-\beta$ all points lie in $B(0,1/2)$ after scaling by a factor of $1/2(1+\alpha)$, the projection map does not move any point and hence the same clustering cost is preserved. Finally, translating all points by the same offset γ makes no difference to the clustering cost. The fact that $\mathrm{OPT} = \Theta(\mathrm{OPT}')$ is a direct consequence of the equality between clustering costs (up to small multiplicative approximation) derived above. □

Definition 4.13. Fixing any level l, we make some definitions to aid our cost analysis.

1. Let $D_l^\dagger := \{p \in D_l : z(p, S_{\mathrm{OPT}}) \le dt_l^2\}$. We make this definition because for every $p \in D_l$, $o(\mathcal{C}_{l-1}(p)) \in S_{\mathcal{H}}$ and $z(p, o(\mathcal{C}_{l-1}(p))) \le dt_l^2$, so $D_l^\dagger$ is the set of points that remains to be covered competitively with S_{OPT} by allocating candidate cluster centers via LSH.
2. For $s \in S_{\mathrm{OPT}}$, let $D_l^\dagger(s) = \{p \in D_l^\dagger : \arg\min_{s'\in S_{\mathrm{OPT}}} z(p,s') = s\}$.
3. From remark 4.3, we see that there are at most $O(kL/\beta)$ non-empty intersections of cells C with clusters $D_l^\dagger(s)$. For every such cell C, we call $D_l^*(s) \cap C$ a *cluster section*.
4. Let s_A be the optimal center for a cluster section A. We can partition A depending on what distance any point in it lies from s_A via geometrically increasing thresholds $\left\{\frac{t_l}{d\sqrt{L}}, \frac{2t_l}{d\sqrt{L}}, \frac{2^2 t_l}{2^l d\sqrt{L}}, \dots, \sqrt{d}t_l\right\}$. For ease of notation we denote these thresholds $r_{l,1}, \dots, r_{l,M}$ and set $r_{l,0} = 0$. By definition, there are $M = \log_2(d^{3/2}\sqrt{L}) = O(\log\log n)$ many such thresholds. With this notation we define the geometrically thresholded partitions of A as $A_j := \{p \in A : \|p - s_A\| \in [r_{l,j}, r_{l,j+1})\}$ for $j = 0, \dots M$. Further, let A_0^m denote the partial union $\bigcup_{j=0}^m A_j$ for $m = 0, \dots, M$.

Lemma 4.14. *With probability $1-\beta$, for all $l \in [L]$ we have the following bound on the number of cluster sections.*

$$\sum_{s\in S_{\mathrm{OPT}}} \left|\{C \in \mathrm{Anc}^*(\mathcal{H}_l) : C \cap D_l^\dagger(s) \neq \emptyset\}\right| = O(kL/\beta)$$

Proof. From lemma 4.2, we know that

$$\mathbb{E}\left[\left|\{C \in \text{Anc}^*(\mathcal{H}_l) : C \cap D_l^\dagger(s) \neq \emptyset\}\right|\right] = O(1).$$

By linearity of expectation and Markov's inequality (in that order), it follows that with probability $1 - \beta/L$,

$$\sum_{s \in S_{\text{OPT}}} \left|\{C \in \text{Anc}^*(\mathcal{H}_l) : C \cap D_l^\dagger(s) \neq \emptyset\}\right| = O(kL/\beta).$$

□

To catch cluster sections which have some $O\left(\frac{\beta\,\text{OPT}}{t_l^2 kL^2 d}\right)$-many points, we use LSH functions applied on a union of heavy cells, where we modify the norm so that the distance between two different cells is always $> 2c \cdot \sqrt{d} \cdot (d^{3/2} t_l)$, i.e. $2c \cdot \text{Diam}(C)$ units where C is any cell in the ancestor level. The diameter of this entire space is still $O(\text{Diam}(C))$ and since the smallest distance at which we need to allocate cluster centers to serve cluster sections is $\frac{t_l}{d\sqrt{L}}$, the ratio of the distance to the farthest false positive to the ratio of the distance of the closest clustering distance is $O(\text{poly}\log n)$.

As discussed in the beginning of the section, this allows us to use LSH functions with reasonable parameters and not end up allocating too many candidate centers. The average of heavy buckets corresponding to sufficiently large cluster sections lie within a distance of c times the threshold at which we are competing with S_{OPT}. Then, by the triangle inequality it will follow that since any two cells are at a distance of strictly greater than $2c\,\text{Diam}(C)$, this average will be closer to the cell in which the cluster section lies than to all other cells. Since a cell is a convex set, we can project to this cell and be assured that the distance to the cluster section the candidate center is meant to cover is not any greater and therefore that we have allocated a candidate center at the desired distance.

Remark 4.15. For the rest of this subsection, we analyse the call to algorithm 3 with the "correct" value of f, i.e. the call such that $k\sqrt{n}2^{f-1} \leq \text{OPT} < k\sqrt{n}2^f$. Since $k\sqrt{n}2^f = \Theta(\text{OPT})$, we will simply refer to $k\sqrt{n}2^f$ as OPT. We will need to scale all failure probabilities by F to ensure that the guarantees hold simultaenously for all calls to algorithm 3 with probability $1 - \beta$.

Definition 4.16 (Synthetic space for LSH functions)**.** 1. Let λ_l denote $(14c+5)r_{l,M} = (14c+5)t_l\sqrt{d}$.

2. Let $\Lambda_l = \mathbb{R}^{\text{Anc}^*(\mathcal{H}_l)} \times \mathbb{R}^d$. We define a mapping $\Lambda_l : [0,1)^d \to \Lambda_l$ as follows;

$$\Lambda_l(p) = \begin{cases} (\lambda_l 1_{\text{Anc}^*(C_l(p))}, p - o(\mathcal{C}_l(p))) & \text{if } p \in D_l \\ 0 & \text{otherwise} \end{cases}$$

   We note that for $p \in D_l$, $\Lambda_l(p)$ is a two-tuple consisting of scaled indicator vector and a copy of the cell itself translated so that the center of the cell lies at the origin. In words, if $p \in D_l$ then the indicator vector indicates which cell in the ancestor level a point lies in. Since this space as defined lies in $\mathbb{R}^{\text{Anc}^*(\mathcal{H}_l)} \times R^d$, it inherits the ℓ_2 norm in the canonical way which we denote $\|\cdot\|_\Lambda$.

3. We have the projection maps to the factor spaces $p_1 : \Lambda_l \to \mathbb{R}^{\text{Anc}^*(\mathcal{H}_l)}$ and $p_2 : \Lambda_l \to \mathbb{R}^d$. Since $\Lambda_l = \mathbb{R}^{\text{Anc}^*(\mathcal{H}_l)} \times \mathbb{R}^d$ is the direct sum of the vector subspaces $\mathbb{R}^{\text{Anc}^*(\mathcal{H}_l)}$ and $\mathbb{R}^d$ we have that $\|\cdot\|_{\Lambda_l}^2 = \|p_1(\cdot)\|^2 + \|p_2(\cdot)\|^2$.

Lemma 4.17. *The following statements hold.*

1. *For any $p, q \in [0,1)^d$ if it is the case that $\text{Anc}^*(\mathcal{C}_l(p)) \neq \text{Anc}^*(\mathcal{C}_l(q))$, then $\|p - q\|_\Lambda > \lambda_l$.*

2. *The diameter of the set of points $\Lambda_l(D_l)$ is $O(\lambda_l)$.*

Proof. 1. By the properties of $\|\cdot\|_{\Lambda_l}$, we have that $\|p - q\|_\Lambda \geq \|p_1(p - q)\|$. Since $\text{Anc}^*(\mathcal{C}_l(p)) \neq \text{Anc}^*(\mathcal{C}_l(q))$, $\|p_1(p-q)\| \geq \lambda_l\sqrt{2}$ ($\|p_1(p-q)\|$ being the difference of two different basis vectors in $\mathbb{R}^{\text{Anc}^*(\mathcal{C}_l)}$ scaled by λ_l. The result follows directly.

2. The bound follows by appealing to the properties of $\|\cdot\|_\Lambda$.

$$\begin{aligned}\|p-q\|_\Lambda &= \sqrt{\|p_1(p-q)\|^2 + \|p_2(p-1)\|^2} \\ &\le \|p_1(p-q)\| + \|p_2(p-q)\| \\ &\le \lambda_l\sqrt{2} + d^{3/2}t_l \\ &= O(\lambda_l).\end{aligned}$$

□

Lemma 4.18. *For $c > \sqrt{2}$, there is a choice of LSH parameters such that*

$$\begin{aligned}\frac{p^2(1)}{p(c)} &\ge \frac{k \operatorname{poly}\log n}{\epsilon_{\mathsf{CH}}\beta^2} \\ p(1) &\ge \tilde{\Omega}\left((k \operatorname{poly}\log n)^{-1/c'}\right), \\ \log N_B &= \tilde{O}(\operatorname{poly}\log n/(\epsilon_{\mathsf{CH}}\beta)),\end{aligned}$$

where the $\tilde{O}$ and $\tilde{\Omega}$ notation suppress $O(\log\log n)$ terms and $c' = c^2/8 - 1/4$. It will be convenient to write $1/c' = O(1/(2c^2-1))$.

Proof. This result is a direct corollary of lemma 2.15. We bound all occurrences of $\log k$ from above by $\log n$. □

We state and prove the guarantees of the bucket histogram as derived from the Bitstogram guarantee.

Lemma 4.19. *For every $l \in [L]$, $m \in [M]$, $r \in [R]$, and $f \in [F]$ with probability $1 - \beta/(LMRF)$ in every call to algorithm 3,*

$$\begin{aligned}\mathsf{BH}_E^{l,m,r,f} &= O\left(\frac{1}{\epsilon_{\mathsf{BH}}}\sqrt{n\log n/\beta}\right) \\ \mathsf{BH}_M^{l,m,r,f} &= O\left(\frac{1}{\epsilon_{\mathsf{BH}}}\sqrt{n \operatorname{poly}\log n/\beta}\right)\end{aligned}$$

As these bounds are invariant in l, m, r, and f we will find it convenient to drop these indices without loss. Since $\mathsf{BH}_M = \Omega(\mathsf{BH}_E)$, we will simply use a uniform bound of BH_M for the estimation error of any frequency query.

Proof. Fix any $l \in [L]$, $m \in [M]$, $r \in [R]$, and $f \in [F]$. We see that $\mathsf{BH}_{l,m,r,f}$ is derived from a call to Bitstogram with the mapping $h_{l,m,r} = p \mapsto H_{\mathrm{Anc}(C_l(p),m,r)}(p)$ and failure probability $\beta/(LMR)$. If the size of the co-domain is the number of buckets N_B for this LSH function, then from lemma 2.11 we have that

$$\mathsf{BH}_M^{l,m,r,f} = O\left(\frac{1}{\epsilon_{\mathsf{BH}}}\sqrt{n\log(N_B \cdot LMRF/\beta)\log(LMRF/\beta)}\right).$$

From lemma 4.18 we have that $\log N_B = \tilde{O}(\operatorname{poly}\log n/(\epsilon_{\mathsf{CH}}\beta))$. Note that since $L, M, F = O(\log n)$, $R = O\left(\frac{\log(kL^2/\beta)}{p(1)}\right)$ and $p(1) = \tilde{\Omega}\left((k \operatorname{poly}\log n)^{-O(1/(2c^2-1))}\right)$, it follows that $\log(LMRF/\beta) = O(\log n/\beta)$. Substituting, we get the stated bound. The expression for $\mathsf{BH}_E^{l,m,r,f}$ follows similarly. □

We see that $\mathsf{BH}_E^{l,m,r,f} = O(\mathsf{BH}_M^{l,m,r,f})$, which we use throughout the remainder of the proof.

Remark 4.20. We observe that by the union bound, lemma 4.19 implies that with probability $1 - \beta$, for all levels $l \in [L]$, thresholds $m \in [M]$, repetitions $r \in [R]$, and OPT guess parameter $f \in [F]$, the frequency query estimation error bound BH_M holds.

Lemma 4.21. *Let A_0^m be a partial union for some cluster section $A \subset D_l^\dagger(s^*) \cap C$ such that $|A_0^m| \geq \max\left(\frac{\beta\,\mathrm{OPT}}{t_l^2 kL^2 d}, \frac{2\mathsf{BH}_M}{p(1)}, O\left(\frac{c_G\sqrt{n}\,\mathrm{poly}\log n/\beta}{\epsilon_{\mathsf{BSO}}}\right)\right)$. With probability $1-\frac{\beta}{kL^2MF}$ there is a point $\Pi_l(\hat{p}_m) \in S_l$ such that for every point $p \in A_0^m$, $\|p-\hat{p}_m\| = O(cr_{l,m})$.*

Proof. We observe that if r is the diameter of A_0^m in Λ_l then $r \leq r_{l,m+1}$ (as all points lie inside the same ancestor cell, the distance between them does not increase in the space Λ_l). Lemma 2.16 gives us that for any fixed arbitrary point $p_m \in A_0^m$, if the average of all points that collide with p_m under a $(p(1), p(c), 2r_{l,m+1}, 2r_{l,m+1}c)$-sensitive hash function is denoted $\bar{p}_m$ then with probability $p(1)/4$,

$$\|p_m - \bar{p}_m\|_{\Lambda_l} \leq 2cr_{l,m+1} + \frac{8p(c)|D_l|}{p^2(1)|A_0^m|}\Delta'.$$

Since $|D_l| = O(\,\mathrm{OPT}\, d^2/t_l^2)$, Δ' is $\mathrm{Diam}(\Lambda_l) = O(ct_l\sqrt{d})$, and $|A_0^m| \geq \frac{\beta\,\mathrm{OPT}}{t_l^2 kL^2 d}$, we get

$$\begin{aligned}
\|p_m - \bar{p}_m\|_{\Lambda_l} &\leq 2cr_{l,m+1} + \frac{p(c)}{p^2(1)} \cdot \frac{(O(\,\mathrm{OPT}\, d^2/t_l^2) + O(\frac{1}{\epsilon_{\mathsf{CH}}\beta}k\sqrt{n}\,\mathrm{poly}\log n))t_l^2 kL^2 d}{\beta\,\mathrm{OPT}} \cdot ct_l\sqrt{d} \\
&\leq 2cr_{l,m+1} + \frac{p(c)}{p^2(1)}\left(\frac{O(\,\mathrm{OPT}\, d^2/t_l^2)\cdot t_l^2 kL^2 d}{\beta\,\mathrm{OPT}} + \frac{(O(\frac{1}{\epsilon_{\mathsf{CH}}\beta}k\sqrt{n}\,\mathrm{poly}\log n))t_l^2 kL^2 d}{\beta\,\mathrm{OPT}} \cdot ct_l\sqrt{d}\right) \\
&\leq 2cr_{l,m+1} + \frac{p(c)}{p^2(1)}\left(\frac{O(ct_l k\,\mathrm{poly}\log n)}{\beta} + \frac{O(ct_l k\,\mathrm{poly}\log n)}{\epsilon_{\mathsf{CH}}\beta^2}\right)
\end{aligned}$$

where in the above we use that $\mathrm{OPT} \geq k\sqrt{n}$. Since $r_{l,m+1} \geq \frac{t_l}{d\sqrt{L}}$ if we choose the LSH parameter such that

$$\frac{p(c)}{p^2(1)} \leq \frac{\epsilon_{\mathsf{CH}}\beta^2}{k\,\mathrm{poly}\log n}$$

then $\|p_m - \bar{p}_m\|_{\Lambda_l} \leq 3cr_{l,m+1}$. Note that the bound on this ratio does not vary with threshold $r_{l,m}$ or level l. This explains the uniform choice of LSH parameters used in algorithm 4. Lemma 4.18 bounds from below the probability $p(1)$ and from above the number of buckets N_B for this choice of LSH parameter.

For any successful run, since $\|p_m - \bar{p}_m\|_{\Lambda_l} \leq 3cr_{l,m+1}$, by the triangle inequality, for any $p \in A_m$

$$\begin{aligned}
\|p - \bar{p}_m\|_{\Lambda_l} &\leq \|p - p_m\|_{\Lambda_l} + \|p_m - \bar{p}_m\|_{\Lambda_l} \\
&\leq r_{l,m+1} + 3cr_{l,m+1} \\
&\leq (3c+1)(2r_{l,m}).
\end{aligned}$$

Since the success probability $p(1)/4$ does not depend upon the level or threshold, a uniform number of $R = O(\log(kL^2F/\beta)/p(1)) = O((\frac{k\log n}{\beta})^{O(1/(2c^2-1))}\log n)$ many independent repetitions of this LSH scheme boost the success probability from $\frac{p(1)}{4}$ to $1-\frac{\beta}{kL^2MF}$. Lemma 2.16 also guarantees that in the successful LSH run at least $\frac{p(1)}{2}\cdot|A_0^m|$ many points in A_0^m will collide with p_m. If $\frac{p(1)}{2}\cdot|A_0^m| \geq \mathsf{BH}_M$, then $h_{l,m,r}(p_m) \in \mathsf{BH}_{l,m,r}$ (lemma 4.19), where $\mathsf{BH}_M = O\left(\frac{1}{\epsilon_{\mathsf{BH}}}\sqrt{n\,\mathrm{poly}\log n/\beta}\right)$. For every $H_{l,m,r,f}(p_m) \in \mathsf{BH}_{l,m,r,f}$, algorithm 4 computes an estimate for $\bar{p}_m$ which we denote

$$\hat{p}_m := \frac{\mathsf{BSO}_{l,m,r}(p_m)}{\mathsf{BH}_{l,m,r,f}(p_m)}.$$

By lemma B.1, if $h_{l,m,r}(p_m) \in \mathsf{BH}_{l,m,r,f}$, then the estimation error on querying $\mathsf{BSO}(h_{l,m,r}(p_m))$ obeys the bound

$$\left\|\frac{\sum_{p:H_{l,m,r,f}(p)=H_{l,m,r,f}(p_m)} p - o(C)}{|\{p : H_{l,m,r,f}(p) = H_{l,m,r,f}(p_m)\}|} - \frac{\mathsf{BSO}_{l,m,r,f}(p_m)}{\mathsf{BH}_{l,m,r,f}(p_m)}\right\|_{\Lambda_l} \leq \frac{1}{\Omega(|A_0^m|)}\cdot O\left(\frac{c_G\,\mathrm{Diam}(\Lambda_l)}{\epsilon_{\mathsf{BSO}}}\sqrt{n\log^2 n/\beta}\right)$$

$$\leq O\left(\frac{c_G\lambda_l\sqrt{n\log^2 n/\beta}}{|A_0^m|\epsilon_{\mathsf{BSO}}}\right).$$

In the above we used that $d = O(\log n)$ to substitute for it in the estimation error. It follows that if $|A_0^m| \geq \frac{c_G c_{\mathsf{BSO}}\,\mathrm{Diam}(\Lambda_l)\sqrt{n\log^2 n/\beta}}{cr_{l,m}\epsilon_{\mathsf{BSO}}}$ for some universal constant c_{BSO} derived from the $\mathsf{HeavySumsOracle}$ guarantee then the additional estimation error in $\|\cdot\|_\Lambda$ norm incurred is $cr_{l,m}$. Substituting, we get that it would suffice to have

$$\begin{aligned}
&|A_0^m| \geq \Omega\left(\frac{c_G\lambda_l\sqrt{n\log^2 n/\beta}}{cr_{l,m}\epsilon_{\mathsf{BSO}}}\right)\\
\Leftrightarrow\ &|A_0^m| \geq \Omega\left(\frac{c_G\cdot cd^2 t_l\cdot\sqrt{n\log^2 n/\beta}}{cr_{l,m}\epsilon_{\mathsf{BSO}}}\right)\\
\Leftarrow\ &|A_0^m| \geq \Omega\left(\frac{2c_G\sqrt{nd^6 L\log^2 n/\beta}}{\epsilon_{\mathsf{BSO}}}\right)
\end{aligned}$$

where in the above we lower bound $r_{l,m}$ by $r_{l,1}$.

So in sum, we have that for any cluster union A_0^m such that $|A_0^m| \geq \frac{c_G\sqrt{nd^6 L\log^2 n/\beta}}{\epsilon_{\mathsf{BSO}}}$, for some fixed arbitrary point $p_m \in A_0^m$, with probability $1-\beta/(kL^2F)$ there exists a hash function $H_{l,m,r,f}$ for some $r \in [R]$ such that the estimate of the average $\hat{p}_m$ over the bucket that p_m maps to lies within a distance of $cr_{l,m}$ units of $\bar{p}_m$, the true average over the heavy bucket, which lies within a distance of $(6c+2)r_{l,m}$ of the point p_m, and by the triangle inequality $\|p_m - \hat{p}_m\| \leq (7c+2)r_{l,m}$.

Now since the distance between any two different cells in the space Λ_l is strictly greater than $(14c+5)r_{l,m} > 2\|p_m - \hat{p}_m\|$, it follows from the triangle inequality that $\Pi_l(\hat{p}_m)$ the projection of $\hat{p}_m$ onto $\cup_{C\in\mathcal{H}_l} C$ lies in the cell C. Indeed, it was to ensure this guarantee that we chose our value of λ_l. Since $\Pi_l(\hat{p}_m)$ is a projection onto a convex set, $\|\Pi_l(\hat{p}_m) - \hat{p}_m\| \leq \|p_m - \hat{p}_m\|$. Now since the diameter of A_0^m is $r_{l,m+1} = 2r_{l,m}$, it follows that every point in A_0^m lies within a distance of $O(cr_{l,m})$ units of $\Pi_l(\hat{p}_m)$. □

Lemma 4.22. *We have the following bound on S_l, the number of candidate centers allocated per level in algorithm 3.*

$$|S_l| = O\left(\frac{k\,\mathrm{poly}\log n}{\beta}\right)^{1+O(1/(2c^2-1))}$$

Proof. A candidate center is allocated for every $b \in \mathsf{BH}^{l,m,r}$ such that $\mathsf{BH}^{l,m,r}(b) \geq T_l - \mathsf{BH}_M$ where

$$T_l = \frac{p(1)}{2}\cdot\max\left(\frac{\beta\,\mathrm{OPT}}{t_l^2 kL^2 d}, \frac{4\mathsf{BH}_M}{p(1)}, O\left(\frac{c_G\sqrt{n\,\mathrm{poly}\log n/\beta}}{\epsilon_{\mathsf{BSO}}}\right)\right).$$

The set of buckets which are identified as having these many points is at most the set of buckets which have $T_l - 2\mathsf{BH}_M$ many points in them. Therefore we want to bound from above the quantity $|D_l|/(T_l - 2\mathsf{BH}_M)$. Since $T_l \geq 4\mathsf{BH}_M$, we can write

$$\begin{aligned}
\frac{|D_l|}{T_l - 2\mathsf{BH}_M} &\leq \frac{2|D_l|}{T_l}\\
&\leq \frac{O(d^2\,\mathrm{OPT}/t_l^2) + O\left(\frac{kL\mathsf{CH}_M}{\beta}\right)}{\max\left\{\frac{p(1)}{2}\frac{\beta\,\mathrm{OPT}}{t_l^2 kL^2 d}, 2\mathsf{BH}_M\right\}}
\end{aligned}$$

$$\leq \frac{O(d^2\,\mathrm{OPT}\,/t_l^2)}{\frac{p(1)}{2}\frac{\beta\,\mathrm{OPT}}{t_l^2 kL^2 d}} + O\left(\frac{kL\mathsf{CH}_M}{\beta}\cdot\frac{1}{\mathsf{BH}_M}\right)$$
$$\leq O\left(\frac{2}{p(1)}\cdot\frac{kL^2d^3}{\beta}\right) + O\left(\frac{kL\mathsf{CH}_M}{\beta\mathsf{BH}_M}\right).$$

We use that $\mathrm{OPT} \geq k\sqrt{n}$, $\mathsf{CH}_M = O\left(\frac{1}{\epsilon_{\mathsf{CH}}}\sqrt{n\,\mathrm{poly}\log n/\beta}\right)$, $\mathsf{BH}_M = \frac{1}{\epsilon_{\mathsf{BH}}}\sqrt{n\,\mathrm{poly}\log n}$ and that

$$\frac{1}{k^{O(1/(2c^2-1))}\,\mathrm{poly}\log n} \leq \frac{\epsilon_{\mathsf{CH}}}{\epsilon_{\mathsf{BH}}} \leq k^{O(1/(2c^2-1))}\,\mathrm{poly}\log n$$

to get

$$\frac{|D_l|}{T_l - 2\mathsf{BH}_M} \leq O\left(\frac{2}{p(1)}\cdot\frac{kL^2d^3}{\beta}\right) + \frac{kL\mathsf{CH}_M}{\beta\mathsf{BH}_M}$$
$$= O\left(\frac{k\,\mathrm{poly}\log n}{\beta}\right)^{1+O(1/(2c^2-1))} + \frac{k\,\mathrm{poly}\log n}{\beta}$$
$$= O\left(\frac{k\,\mathrm{poly}\log n}{\beta}\right)^{1+O(1/(2c^2-1))}.$$

Taking the union over all possible values of (l, m, r) and absorbing the addition log factors in the poly log term, we get the stated bound. The fact that the bounds on the ratio of ϵ_{CH} to ϵ_{BH} is adhered to can be checked in the proof of the main theorem at the end of this section. □

Lemma 4.23. *In algorithm 4, the following bound holds for the total number of candidate centers allocated.*

$$|S| = \left(\frac{k\,\mathrm{poly}\log n}{\beta}\right)^{1+O(1/(2c^2-1))}$$

Proof. We observe that S in algorithm 4 equals $\cup_{f\in O(\log n)} S_f$ is the union of $F = O(\log n)$ many sets of candidate centers returned by calls to algorithm 3. It therefore suffices to bound the set of candidate centers

$$\left|S_{\mathcal{H}} \cup \bigcup_{l\in[L]} S_l\right|$$

returned by algorithm 3. From lemma 4.22, by adding the bounds for S_l over L levels, absorbing the factor of L into the poly log n term and noting that the $|S_{\mathcal{H}}| = O(kL^2/\beta)\cdot L$ summand is asymptotically dominated by $|\cup_{l\in L} S_l|$, we get that

$$\left|S^f\right| = O\left(\frac{k\,\mathrm{poly}\log n}{\beta}\right)^{1+O(1/(2c^2-1))}.$$

The stated bound now follows simply be absorbing an $O(\log n)$ factor in the poly log term. □

Definition 4.24. Let f^* denote the "correct" call to algorithm 3, i.e. the unique value of $f \in [F]$ such that $k\sqrt{n}2^{f-1} \leq \mathrm{OPT} < k\sqrt{n}2^f$

Lemma 4.25. *Let $A \subset D_l^\dagger(s) \cap C$ be some cluster section. Then with probability $1 - \beta/kL^2$ we have that*

$$f_A(S^{f^*}) = O(f_A(S_{\mathrm{OPT}}) + \max\left(\frac{\beta\,\mathrm{OPT}}{t_l^2 kL^2 d}, \frac{4\mathsf{BH}_M}{p(1)}, O\left(\frac{c_G\sqrt{n\,\mathrm{poly}\log n/\beta}}{\epsilon_{\mathsf{BSO}}}\right)\right)\cdot dt_l^2 + \frac{c^2 t_l^2}{d^2 L}\cdot|A|.$$

Proof. Let m' be the largest index such that $|A_0^{m'}| < \max\left(\frac{\beta\,\mathrm{OPT}}{t_l^2 kL^2 d}, \frac{4\mathsf{BH}_M}{p(1)}, O\left(\frac{c_G\sqrt{n\,\mathrm{poly}\log n/\beta}}{\epsilon_{\mathsf{BSO}}}\right)\right)$. We can write

$$f_A(S^{f^*}) \le f_{A_0^{m'}}(S^{f^*}) + \sum_{m=m'+1}^{M} f_{A_m}(S^{f^*}).$$

For every $m > m'$, by lemma 4.21 we know there is a candidate center $\Pi_l(\hat{p}_m) \in S^{f^*}$ such that for every point $p \in A_0^m$, $\|p - \Pi_l(\hat{p}_m)\| = O(cr_{l,m})$. Since for all $p \in A_m \subset A_0^m$, $\|p - s\| \ge r_{l,m}$, it follows that with probability $1-\beta/kL^2M$, $f_{A_m}(S) \le O(f_{A_m}(S_{\mathrm{OPT}}) + c^2 r_{l,1}^2$. By the union bound, we have that this guarantee holds for all thresholds with probability $1-\beta/kL^2$. Substituting this bound, we get

$$\begin{aligned} f_A(S^{f^*}) &\le \max\left(\frac{\beta\,\mathrm{OPT}}{t_l^2 kL^2 d}, \frac{4\mathsf{BH}_m}{p(1)}, O\left(\frac{c_G\sqrt{n\,\mathrm{poly}\log n/\beta}}{\epsilon_{\mathsf{BSO}}}\right)\right)\cdot dt_l^2 + \sum_{m=m'+1}^{M}\left[O(cf_{A_m}(S_{\mathrm{OPT}})) + c^2 r_{l,1}^2\right] \\ &\le \max\left(\frac{\beta\,\mathrm{OPT}}{t_l^2 kL^2 d}, \frac{4\mathsf{BH}_M}{p(1)}, O\left(\frac{c_G\sqrt{n\,\mathrm{poly}\log n/\beta}}{\epsilon_{\mathsf{BSO}}}\right)\right)\cdot dt_l^2 + O(f_A(S_{\mathrm{OPT}}) + \frac{c^2 t_l^2}{d^2 L}\cdot|A|. \end{aligned}$$

□

Lemma 4.26. *The following bound holds.*

$$\begin{aligned} f_{D_l}(S^{f^*}) = O(f_{D_l}(S_{\mathrm{OPT}})) + O(\,\mathrm{OPT}\,/L) + O(\mathsf{CH}_M/(d^2\beta)) \\ + O\left(\frac{c_G}{\epsilon_{\mathsf{BSO}}\beta}(k\,\mathrm{poly}\log n)^{1+O(1/(2c^2-1))}\sqrt{n}\right) \end{aligned}$$

Proof. Let $\mathcal{A}$ be the set of all cluster sections. We know that $|\mathcal{A}| = O(kL/\beta)$ and that $D_l^\dagger = \sqcup_{A\in\mathcal{A}} A$. We can write

$$\begin{aligned} f_{D_l^\dagger}(S^{f^*}) &= \sum_{A\in\mathcal{A}} f_{\mathcal{A}}(S^{f^*}) \\ &= \sum_{A\in\mathcal{A}}\left[O(f_A(S_{\mathrm{OPT}}) + \frac{c^2 t_l^2}{d^2 L}\cdot|A| + \max\left(\frac{\beta\,\mathrm{OPT}}{t_l^2 kL^2 d}, \frac{4\mathsf{BH}_M}{p(1)}, O\left(\frac{c_G\sqrt{n\,\mathrm{poly}\log n/\beta}}{\epsilon_{\mathsf{BSO}}}\right)\right)\cdot dt_l^2\right] \\ &= O(f_{D_l^\dagger}(S_{\mathrm{OPT}})) + \frac{c^2 t_l^2}{d^2 L}\cdot\left|D_l^\dagger\right| + \max\left(O\left(\frac{\mathrm{OPT}}{L}\right), \frac{4\mathsf{BH}_M}{p(1)} O\left(\frac{kLd}{\beta}\right), O\left(\frac{c_G k\sqrt{n\,\mathrm{poly}\log n/\beta}}{\epsilon_{\mathsf{BSO}}\beta}\right)\right) \end{aligned}$$

where in the above we absorb a factor of d in the $\mathrm{poly}\log n$ expression. To bound the second term, we recall that $|D_l| \le O(d^2\,\mathrm{OPT}\,/t_l^2) + O(kL\mathsf{CH}_M/\beta)$. Substituting, we get

$$\begin{aligned} \frac{c^2 t_l^2}{d^2 L}\cdot\left|D_l^\dagger\right| &\le \frac{c^2 t_l^2}{d^2 L}\cdot(O(d^2\,\mathrm{OPT}\,/t_l^2) + O(kL\mathsf{CH}_M/\beta)) \\ &\le O(\,\mathrm{OPT}\,/L) + O(k\mathsf{CH}_M/(d^2\beta)) \end{aligned}$$

We now simplify the last term in the upper bound for $f_{D_l^\dagger}$ by noting that for any call to algorithm 4, the parameter c is a constant, that $d, L = \log n$, and that since we can let $\epsilon_{\mathsf{BSO}} = \epsilon_{\mathsf{BH}}$ (as they are called exactly the same number of times). In sum

$$\begin{aligned} &\max\left(O\left(\frac{\mathrm{OPT}}{L}\right), \frac{4\mathsf{BH}_M}{p(1)} O\left(\frac{kLd}{\beta}\right), O\left(\frac{c_G k\sqrt{n\,\mathrm{poly}\log n/\beta}}{\epsilon_{\mathsf{BSO}}\beta}\right)\right) \\ &\le O\left(\frac{c_G}{\epsilon_{\mathsf{BSO}}\beta}(k\,\mathrm{poly}\log n)^{1+O(1/(2c^2-1))}\sqrt{n}\right). \end{aligned}$$

We recall that $D_l^\dagger = \{p \in D_l : z(p, S_{\mathrm{OPT}}) < dt_l^2\}$. It follows that if $p \in D_l \backslash D_l^\dagger$ then $z(p, S_{\mathrm{OPT}}) > dt_l^2$, in which case $z(p, S_{\mathcal{H}}) < z(p, S_{\mathrm{OPT}})$. In sum, $f_{D_l}(S^{f^*}) \leq f_{D_l^\dagger}(S^{f^*}) + f_{D_l}(S_{\mathrm{OPT}})$, from which the stated bound follows directly. □

Lemma 4.27. *The following bound holds.*

$$f_D(S) = O(\mathrm{OPT}) + O\left(\left(\frac{c_G}{\epsilon_{\mathsf{BSO}}\beta} + \frac{1}{\epsilon_{\mathsf{CH}}}\right)(k \operatorname{poly}\log n)^{1+O(1/(2c^2-1))}\sqrt{n}\right).$$

Proof.

$$\begin{aligned} f_D(S^{f^*}) &= \sum_{l\in[L]} f_{D_l}(S^{f^*}) \\ &= \sum_{l\in[L]} \Bigg[O(f_{D_l}(S_{\mathrm{OPT}})) + O(\mathrm{OPT}/L) + O(k\mathsf{CH}_M/(d^2\beta)) \\ &\quad + O\left(\frac{c_G}{\epsilon_{\mathsf{BSO}}\beta}(k \operatorname{poly}\log n)^{1+O(1/(2c^2-1))}\sqrt{n}\right)\Bigg] \\ &= O(\mathrm{OPT}) + O(k\mathsf{CH}_M/(d^2\beta)) + O\left(\frac{c_G}{\epsilon_{\mathsf{BSO}}\beta}(k \operatorname{poly}\log n)^{1+O(1/(2c^2-1))}\sqrt{n}\right) \end{aligned}$$

where in the above we use that $L = \log n$ to absorb a factor of L in the $\operatorname{poly}\log n$ expression. Now since $S^{f^*} \subset S$, we can write

$$O(\mathrm{OPT}) + O(k\mathsf{CH}_M/(d^2\beta)) + O\left(\frac{c_G}{\epsilon_{\mathsf{BSO}}\beta}(k \operatorname{poly}\log n)^{1+O(1/(2c^2-1))}\sqrt{n}\right)$$

We now simplify this expression by opening up the expression for CH_M, and by absorbing the c^2 term in the big-Oh notation.

$$f_D(S) = O(\mathrm{OPT}) + O\left(\left(\frac{c_G}{\epsilon_{\mathsf{BSO}}\beta} + \frac{1}{\epsilon_{\mathsf{CH}}}\right)(k \operatorname{poly}\log n)^{1+O(1/(2c^2-1))}\sqrt{n}\right).$$

□

4.4 Cost analysis

In this subsection we complete the cost analysis of this algorithm. In the previous section we showed that the candidate centers allocated serve as a good bi-criteria solution for the k-means problem with respect to the dimension reduced data set D. We will be able to use this in turn to show that proxy data set D^* constructed in algorithm 5 has a similar k-means clustering function to that of D. This result implies that the k cluster centers derived from non-private clustering of D^* work well as cluster centers for D. Finally, we conclude our cost analysis by bounding the cost incurred when clustering the original data set D' with the k centers in S' returned after undoing the dimension reduction.

Definition 4.28 (Proxy dataset)**.** 1. From algorithm 5 we see that

$$D^* = \{s \in S \text{ with multiplicity } \hat{n}_s \text{ for all } (s, \hat{n}_s) \in \mathsf{CCH}\}.$$

We call this the *proxy data set* for D.

2. We let $D(s) = \{p \in D : \arg\min_{s_1 \in S} z(p, s_1) = s\}$.

Lemma 4.29. *With probability* $1 - 2\beta$ *we have that*

Data: Bicriteria k-means relaxation S for k-means clustering under dimension reducing transformation M, the tranformation $M : \mathbb{R}^{d'} \to \mathbb{R}^d$

$s(p) := p \mapsto \arg\min_{s \in S} \|p - s\|_2$

$\mathsf{CCH} = \mathsf{Bitstogram}(s(\cdot), \beta, \epsilon_{\mathsf{SH}})$; `/* Candidate center histogram */`

$D^* \leftarrow \{s \in S \text{ with multiplicity } \mathsf{SH}(s)\}$

$S^* = \{s_1^*, \ldots, s_k^*\} \leftarrow \mathsf{Standard}\ k - \mathsf{Means}$

$s^*(p) := p \mapsto \arg\min_{s^* \in S^*} \|M(p) - s^*\|_2$

Do in parallel:

Agents reveal $\hat{v}(p)$ for $p \in D'$ where

$$v(p)_s = \begin{cases} p \text{ if } s = s^*(p) \\ 0 \text{ otherwise} \end{cases}$$

$$\hat{v}(p) = v(p) + N\left(0, \frac{c_{G'}^2 \cdot 2}{\epsilon_{G'}^2} \mathbb{I}_{d'k}\right)$$

$\mathsf{SH} = \mathsf{Bitstogram}(s^*(\cdot), \beta, \epsilon_{\mathsf{SH}})$; `/* Cluster centers histogram */`

end

$\hat{v} = \sum_{p \in D'} \hat{v}(p)$

$\hat{s}^* = \sum_{p \in D'} \hat{s}^*(p)$

for $j = 1, \ldots, k$ **do**

$\hat{\mu}_j = \frac{\hat{v}_j}{\mathsf{SH}(s_j^*)}$

end

return $S' = \{\hat{\mu}_1, \ldots, \hat{\mu}_k\}$

Algorithm 5: 2-Round Center Recovery

1. *For all $s \in S$ we have $|\{p \in D : s(p) = s\}| - \mathsf{CCH}(s)| \le O(\frac{1}{\epsilon_{\mathsf{CCH}}} \log n/\beta)$.*

2. *For all $s^* \in S^*$ we have that $||p \in D' : s^*(M(p)) = s^*|| \le O(\frac{1}{\epsilon_{\mathsf{SH}}} \log n/\beta)$*

Proof. The stated bounds follow from the Bitstogram guarantee. We use the values CCH_M and SH_M as uniform error bounds. Note that the size of the co-domain for $s(\cdot)$ is $|S|$ and $\log|S| = O(\log n)$. Similarly the size of the co-domain for $s^*(\cdot)$ is $|S^*| = k$, so the second bound follows directly as well. □

Lemma 4.30. *The k-means clustering functions of D and D^* are similar. Concretely, for any finite set S_1, the following bounds hold.*

$$f_{D^*}(S_1) \le 2f_D(S) + 2f_D(S_1) + O\left(\frac{|S|}{\epsilon_{\mathsf{CCH}}}\sqrt{n}\log n/\beta\right),$$
$$f_D(S_1) \le 2f_D(S) + 2f_{D^*}(S_1) + O\left(\frac{|S|}{\epsilon_{\mathsf{CCH}}}\sqrt{n}\log n/\beta\right).$$

As a direct corollary,

$$f_{D^*}(S_{\mathrm{OPT}}) \le O(\mathrm{OPT}) + O\left(\left(\frac{c_G}{\epsilon_{\mathsf{BSO}}\beta} + \frac{1}{\epsilon_{\mathsf{CH}}}\right)(k \operatorname{poly}\log n)^{1+O(1/(2c^2-1))}\sqrt{n}\right) + O\left(\frac{|S|}{\epsilon_{\mathsf{CCH}}}\sqrt{n}\log n/\beta\right).$$

Proof. We can enumerate all points in D^* by counting each candidate center in $s \in S$ a total of $\hat{n}_s$ many times.

$$\begin{aligned}
f_{D^*}(S_1) &= \sum_{p^* \in D^*} \min_{s \in S_1} z(p^*, s) \\
&= \sum_{s \in S} \hat{n}_s \min_{s' \in S_1} z(s, s') \\
&= \sum_{s \in S} |D(s)| \min_{s' \in S_1} z(s, s') + \hat{n}_s - |D(s)| \\
&\le \sum_{p \in D} z(s(p), \arg\min_{s' \in S_1} z(s(p), s')) + O\left(\frac{|S|}{\epsilon_{\mathsf{CCH}}}\sqrt{n}\log n/\beta\right) \\
&\le \sum_{p \in D} z(s(p), \arg\min_{s' \in S_1} z(p, s')) + O\left(\frac{|S|}{\epsilon_{\mathsf{CCH}}}\sqrt{n}\log n/\beta\right) \\
&\le \sum_{p \in D} 2z(s(p), p) + 2z(p, \arg\min_{s' \in S_1} z(p, s')) + O\left(\frac{|S|}{\epsilon_{\mathsf{CCH}}}\sqrt{n}\log n/\beta\right) \\
&\le 2f_D(S) + 2f_D(S_1) + O\left(\frac{|S|}{\epsilon_{\mathsf{CCH}}}\sqrt{n}\log n/\beta\right)
\end{aligned}$$

where we apply the weak triangle inequality for ℓ_2^2 distance. Proceeding similarly,

$$\begin{aligned}
f_D(S_1) &= \sum_{p \in D} \min_{s \in S_1} z(p, s) \\
&= \sum_{s \in S} \sum_{p \in D(s)} \min_{s \in S_1} z(p, s) \\
&\le \sum_{s \in S} \sum_{p \in D(s)} z(p, \arg\min_{s_1 \in S_1} z(s, s_1))
\end{aligned}$$

$$
\begin{aligned}
&\leq \sum_{s\in S}\sum_{p\in D(s)} \min_{s\in S_1} 2z(p,s) + 2z(s, \arg\min_{s_1\in S_1} z(s,s_1)) \\
&\leq 2f_D(S) + \sum_{s\in S} |D(s)| 2z(s, \arg\min_{s_1\in S_1} z(s,s_1)) \\
&\leq 2f_D(S) + \sum_{s\in S} (\hat{n}_s) 2z(s, \arg\min_{s_1\in S_1} z(s,s_1)) + (|D(s)| - \hat{n}_S) \\
&\leq 2f_D(S) + 2f_{D^*}(S_1) + O\left(\frac{|S|}{\epsilon_{\mathsf{CCH}}}\sqrt{n}\log n/\beta\right)
\end{aligned}
$$

The corollary follows by substituting our upper bound for $f_D(S)$ in its place. □

Lemma 4.31. *If the set S^* is such that*

$$f_{D^*}(S^*) \leq \eta \min_{S_1:|S_1|=k} f_{D^*}(S_1)$$

for some universal constant η (for instance the guarantee of the non-private clustering algorithm) then

$$
\begin{aligned}
f_{D^*}(S^*) &= O(\,\mathrm{OPT}\,) + O\left(\left(\frac{c_G}{\epsilon_{\mathsf{BSO}}\beta} + \frac{1}{\epsilon_{\mathsf{CH}}}\right)(k\,\mathrm{poly}\log n)^{1+O(1/(2c^2-1))}\sqrt{n}\right) + O\left(\frac{|S|}{\epsilon_{\mathsf{CCH}}}\sqrt{n}\log n/\beta\right) \\
f_D(S^*) &= O(\,\mathrm{OPT}\,) + O\left(\left(\frac{c_G}{\epsilon_{\mathsf{BSO}}\beta} + \frac{1}{\epsilon_{\mathsf{CH}}}\right)(k\,\mathrm{poly}\log n)^{1+O(1/(2c^2-1))}\sqrt{n}\right) + O\left(\frac{|S|}{\epsilon_{\mathsf{CCH}}}\sqrt{n}\log n/\beta\right).
\end{aligned}
$$

Proof. The first bound follows from the lemma 4.30 by noting that $f_{D^*}(S_{\mathrm{OPT}})$ is an upper bound for $\min_{S':|S'|=k} f_{D^*}(S')$, and by absorbing the universal constant η in the big-Oh notation. The second bound follows from the first bound and lemma 4.30. □

We have shown that the k-means solution found in the dimension reduced space for the proxy dataset works well for the dimension reduced dataset. Now we use the cluster sets hence derived to privately estimate cluster centers in the original space.

Given a clustering of D' in the original space by identifying points with the clusters derived from S^* in the dimension reduced space, we know that the k-means cost of the clustering is of the same order as the k-means cost in the dimension reduced space, as proved in lemma 4.12. We recover the cluster centers in the original space via noisy averaging. In algorithm 5, each point holds a k-tuple of d'-dimensional vector $v(p)$ which we can naturally identify as a kd' dimensional vector. If s_i^* is closest to p in the low-dimensional space (breaking ties arbitrarily), then the ith tuple value is p and all other tuples are the zero vector. To preserve privacy, agents release this vector via the Gaussian mechanism.

Lemma 4.32.

$$
\begin{aligned}
f_{D'}(S') = O(\,\mathrm{OPT}\,) &+ O\left(\left(\frac{c_G}{\epsilon_{\mathsf{BSO}}\beta} + \frac{1}{\epsilon_{\mathsf{CH}}}\right)(k\,\mathrm{poly}\log n)^{1+O(1/(2c^2-1))}\sqrt{n}\right) + O\left(\frac{|S|}{\epsilon_{\mathsf{CCH}}}\sqrt{n}\log n/\beta\right) \\
&+ O\left(\left(\frac{c_{G'}}{\epsilon_{G'}} + \frac{1}{\epsilon_{\mathsf{SH}}}\right)k\sqrt{d'n}\log n/\beta\right).
\end{aligned}
$$

Proof. For $s \in S^*$ let $D'(s) = \{p \in D' : M(p) \in D(s)\}$, where we recall that M was the composition of the dimension reduction, scaling, projection and translation maps, and $D(s) = \{p \in D : \arg\min_{s_1\in S^*} z(p,s_1) = s\}$. Let $\mu_j = \sum_{p\in D(s_j)} p/|D(s_j)|$. From lemma 4.12 we have that

$$f_{D'}(\{\mu_1,\ldots,\mu_k\}) = O(f_D(S^*)).$$

In lemma 4.31 we have derived the bound

$$f_D(S^*) \leq O(\,\mathrm{OPT}\,) + O\left(\left(\frac{c_G}{\epsilon_{\mathsf{BSO}}\beta} + \frac{1}{\epsilon_{\mathsf{CH}}}\right)(k\,\mathrm{poly}\log n)^{1+O(1/(2c^2-1))}\sqrt{n}\right) + O\left(\frac{|S|}{\epsilon_{\mathsf{CCH}}}\sqrt{n}\log n/\beta\right).$$

In algorithm 5 we construct estimates $\hat{\mu_j} = \hat{v}_j / \mathsf{SH}(s_j^*)$ for the μ_j. We now bound the addition error incurred during this approximation step.

We see that $\hat{v}_j = \sum_{p \in D'(s_j^*)} p + N\left(0, \frac{2c_G^2}{\epsilon_G^2}\mathbb{I}_{d'}\right)$. If we denote the random noise added by the agent with data p by η_p, then we have

$$P\left(\left\|\sum_{p \in D'(s_j^*)} \eta_p\right\| \geq t\right) \leq \exp\left(\frac{-\epsilon_G^2 t^2}{16 d' n c_G^2}\right).$$

So there is a choice of

$$t = O\left(\frac{c_{G'}\sqrt{d' n \log k/\beta}}{\epsilon_{G'}}\right)$$

such that $\|\hat{v}_j - \sum_{p \in D'(s_j^*)} p\| \leq t$ with probability $1 - \beta/k$. From lemma 4.29, we have that

$$|\mathsf{SH}(s_j^*) - D(s_j^*)| \leq O\left(\frac{1}{\epsilon_{\mathsf{SH}}}\sqrt{n}\log n/\beta\right).$$

It follows that by the union bound that all these bounds hold simultaneously with probability $1 - 2\beta$. For all clusters $D'(s_j^*)$ which have more than $2\mathsf{SH}_M$ data points we have that $\mathsf{SH}(s_j^*) = \Theta(\left|D'(s_j^*)\right|)$, and for all smaller clusters since the diameter of the data domain is 1 unit, $f_{D'(s_j^*)} \leq \left|D'(s_j^*)\right| = O\left(\frac{|S|}{\epsilon_{\mathsf{CCH}}}\sqrt{n}\log n/\beta\right)$ unconditionally. Assuming that the former case holds, we get that the error bounds for $\hat{v}_s$ and $\mathsf{SH}(s_j^*)$ give us

$$\begin{aligned}
\left\|\hat{\mu}_j - \mu_j\right\| &= \left\|\frac{\hat{v}_j}{\mathsf{SH}(s_j^*)} - \frac{\sum_{p \in D'(s_j^*)} p}{\left|D(s_j^*)\right|}\right\| \\
&= \left\|\frac{\hat{v}_j}{\mathsf{SH}(s_j^*)} - \frac{\sum_{p \in D'(s_j^*)} p}{\mathsf{SH}(s_j^*)}\right\| + \left\|\frac{\sum_{p \in D'(s_j^*)} p}{\mathsf{SH}(s_j^*)} - \frac{\sum_{p \in D'(s_j^*)} p}{\left|D(s_j^*)\right|}\right\| \\
&\leq O\left(\frac{c_{G'}\sqrt{d' n \log k/\beta}}{\epsilon_{G'}\left|D'(s_j^*)\right|}\right) + O\left(\frac{1}{\epsilon_{\mathsf{SH}}\left|D'(s_j^*)\right|}\sqrt{n}\log n/\beta\right)\|\mu_j\|
\end{aligned}$$

We can bound $\|\mu_j\|$ from above by $O(1)$ since the domain is of unit diameter. We can then state a simplified bound of

$$\left\|\hat{\mu}_j - \mu_j\right\| = O\left(\left(\frac{c_{G'}}{\epsilon_{G'}} + \frac{1}{\epsilon_{\mathsf{SH}}}\right)\frac{\sqrt{d' n}\log n/\beta}{\left|D'(s_j^*)\right|}\right).$$

From lemma D.3, we can bound the cost of cluster $D'(s_j^*)$ via $S' = \{\hat{\mu}_j : j = 1, \ldots, k\}$ by the following relation

$$\begin{aligned}
f_{D'(s_j^*)}(S') &\leq f_{D'(s_j)}(\{\mu_1, \ldots, \mu_k\}) + |D'(s_j)|\left\|\mu_j - \hat{\mu}_j\right\|^2 \\
&\leq O(f_{D(s_j^*)}(S^*)) + |D'(s_j)| O\left(\left(\frac{c_{G'}}{\epsilon_{G'}} + \frac{1}{\epsilon_{\mathsf{SH}}}\right)^2 \frac{d' n \log^2 n/\beta}{\left|D'(s_j^*)\right|^2}\right)
\end{aligned}$$

For each cluster $D'(s_j^*)$, we see that if $D'(s_j^*) \geq \left(\frac{c_{G'}}{\epsilon_{G'}} + \frac{1}{\epsilon_{\mathsf{SH}}}\right)\sqrt{d' n}\log n/\beta$, then

$$f_{D'(s_j^*)}(S') \leq O(f_{D(s_j^*)}(S^*)) + O\left(\left(\frac{c_{G'}}{\epsilon_{G'}} + \frac{1}{\epsilon_{\mathsf{SH}}}\right)\frac{\sqrt{d' n}\log n/\beta}{\left|D'(s_j^*)\right|}\right)$$

On the other hand, if $D'(s) < \left(\frac{c_{G'}}{\epsilon_{G'}} + \frac{1}{\epsilon_{\mathsf{SH}}}\right)\sqrt{d'n}\log n/\beta$, then we have the same bound unconditionally since the diameter of the data domain is $O(1)$. Summing up over cluster over all size ranges, we get

$$\begin{aligned} f_{D'}(S') &= O(f_D(S^*)) + O\left(\frac{|S|}{\epsilon_{\mathsf{CCH}}}k\sqrt{n}\log n/\beta\right) + O\left(\left(\frac{c_{G'}}{\epsilon_{G'}} + \frac{1}{\epsilon_{\mathsf{SH}}}\right)k\sqrt{d'n}\log n/\beta\right) \\ &= O(\,\mathrm{OPT}\,) + O\left(\left(\frac{c_G}{\epsilon_{\mathsf{BSO}}\beta} + \frac{1}{\epsilon_{\mathsf{CH}}}\right)(k\,\mathrm{poly}\log n)^{1+O(1/(2c^2-1))}\sqrt{n}\right) + O\left(\frac{|S|}{\epsilon_{\mathsf{CCH}}}\sqrt{n}\log n/\beta\right) \\ &+ O\left(\left(\frac{c_{G'}}{\epsilon_{G'}} + \frac{1}{\epsilon_{\mathsf{SH}}}\right)k\sqrt{d'n}\log n/\beta\right). \end{aligned}$$

□

We can now derive the main result of this section.

Theorem 1.2. *Algorithm 4 is an (ϵ, δ)-locally differentially private algorithm such that given $c > \sqrt{2}$, after four rounds of interaction with a private distributed data set $D' \subset \mathbb{R}^{d'}$ of size n outputs a set S' of size k such that with probability $1 - \beta$,*

$$f_{D'}(S') = O(\,\mathrm{OPT}\,') + O\left(\frac{1}{\epsilon}\sqrt{d'n\ln(n/\delta)}\right)\left(\frac{k\,\mathrm{poly}\log n}{\beta}\right)^{1+O(1/(2c^2-1))}.$$

Proof. To prove this theorem, we will account for all privacy loss and then scale the privacy parameters used in each data access subroutine to ensure a net (ϵ, δ) privacy loss guarantee. We will then substitute these parameters into lemma 4.32 to derive the bound on the cost incurred with this choice of parameters.

We see that data access occurs in 4 rounds through the following mechanisms:

1. L calls in parallel to Bitstogram to construct CH^l for $l \in [L]$ with privacy parameter ϵ_{CH}.
2. $FLMR$ calls in parallel to Bitstogram and HeavySumsOracle to construct $\mathsf{BH}_{l,m,r,f}$ and $\mathsf{BSO}_{l,m,r,f}$ for $l \in [L], m \in [M], r \in [R]$ and $f \in [F]$. The two types of calls have respective privacy parameters ϵ_{BH} and $(\epsilon_{\mathsf{BSO}}, \delta_{\mathsf{BSO}})$ (note that δ_{BSO} occurs in our cost guarantee inside the Gaussian mechanism parameter c_G). Recall that during the course of our analysis we required that $\epsilon_{\mathsf{BH}} = \epsilon_{\mathsf{BSO}}$ with the observation that they were called an equal number of times.
3. One call to Bitstogram to construct CCH with privacy parameter ϵ_{CCH}
4. Gaussian mechanism and one call to Bitstogram to construct SH in parallel when computing the noisy averages over cluster sets derived from low-dimensional clustering. The respective privacy parameters are $(\epsilon_{G'}, \delta_{G'})$ and ϵ_{SH} (note that $\delta_{G'}$ occurs in our cost guarantee inside the Gaussian mechanism parameter $c_{G'}^2$).

We allocate private parameters of $(\epsilon/4, 0)$, $(\epsilon/4, \delta/2)$, $(\epsilon/4, 0)$ and $(\epsilon/4, \delta/2)$ to each of these four steps, and sub-divide the privacy parameters within. Since

$$\begin{aligned} FLMR &= O(\log n)\cdot O(\log n)\cdot O(\log\log n)\cdot O\left(k\,\mathrm{poly}\log n\right)^{O(1/(2c^2-1))} \\ &= k^{O(1/(2c^2-1))}\,\mathrm{poly}\log n \end{aligned}$$

we can write

$$\begin{aligned} \epsilon_{\mathsf{CH}} &= \frac{\epsilon}{4\log n} \\ \epsilon_{\mathsf{BH}} = \epsilon_{\mathsf{BSO}} &= \frac{\epsilon}{8k^{O(1/(2c^2-1))}\,\mathrm{poly}\log n} \\ \delta_{\mathsf{BSO}} &= \frac{\delta}{2k^{O(1/(2c^2-1))}\,\mathrm{poly}\log n} \end{aligned}$$

$$
\begin{aligned}
&\Rightarrow c_G < O(1/(2c^2-1))\sqrt{\ln(n/\delta)} \\
&\epsilon_{\mathsf{CCH}} = \frac{\epsilon}{4} \\
&\epsilon_{G'} = \epsilon_{\mathsf{SH}} = \frac{\epsilon}{8} \\
&\delta_{G'} = \frac{\delta}{2}. \\
&\Rightarrow c_{G'} = O(\sqrt{\ln(1/\delta)}).
\end{aligned}
$$

Substituting these terms along with the bound

$$
|S| \leq O\left(\frac{k \operatorname{poly}\log n}{\beta}\right)^{1+O(1/(2c^2-1))}
$$

in the cost guarantee of lemma 4.32, we get

$$
\begin{aligned}
f_{D'}(S') &= O(\operatorname{OPT}) + O\left(\left(\frac{\sqrt{\ln(n/\delta)}}{\epsilon\beta} + \frac{\log n}{\epsilon}\right)(k \operatorname{poly}\log n)^{1+O(1/(2c^2-1))}\sqrt{n}\right) \\
&\quad + O\left(\frac{1}{\epsilon}\sqrt{n}\left(\frac{k \operatorname{poly}\log n}{\beta}\right)^{1+O(1/(2c^2-1))}\right) + O\left(\left(\frac{\sqrt{\ln(1/\delta)}}{\epsilon}\right)k\sqrt{d'n}\log n/\beta\right) \\
&\leq O(\operatorname{OPT}) + O\left(\frac{1}{\epsilon}\sqrt{d'n\ln(n/\delta)}\right)\left(\frac{k \operatorname{poly}\log n}{\beta}\right)^{1+O(1/(2c^2-1))}.
\end{aligned}
$$

□

Acknowledgments

All the authors were supported by the National Science Foundation under NSF grants AF 1909314 and CAREER 1750716.

References

Alexandr Andoni and Piotr Indyk. Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions. In *47th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2006), 21-24 October 2006, Berkeley, California, USA, Proceedings*, pages 459–468. IEEE Computer Society, 2006. doi: 10.1109/FOCS.2006.49. URL `https://doi.org/10.1109/FOCS.2006.49`.

Maria-Florina Balcan, Travis Dick, Yingyu Liang, Wenlong Mou, and Hongyang Zhang. Differentially private clustering in high-dimensional euclidean spaces. In *International Conference on Machine Learning*, pages 322–331. PMLR, 2017.

Raef Bassily, Kobbi Nissim, Uri Stemmer, and Abhradeep Thakurta. Practical locally private heavy hitters. *J. Mach. Learn. Res.*, 21:16:1–16:42, 2020. URL `http://jmlr.org/papers/v21/18-786.html`.

Andrea Bittau, Úlfar Erlingsson, Petros Maniatis, Ilya Mironov, Ananth Raghunathan, David Lie, Mitch Rudominer, Ushasree Kode, Julien Tinnes, and Bernhard Seefeld. Prochlo: Strong privacy for analytics in the crowd. In *Proceedings of the 26th Symposium on Operating Systems Principles*, pages 441–459, 2017.

Vladimir Braverman, Gereon Frahling, Harry Lang, Christian Sohler, and Lin F. Yang. Clustering high dimensional dynamic data streams. *CoRR*, abs/1706.03887, 2017. URL `http://arxiv.org/abs/1706.03887`.

T.-H. Hubert Chan, Elaine Shi, and Dawn Song. Optimal lower bound for differentially private multi-party aggregation. In Leah Epstein and Paolo Ferragina, editors, *Algorithms - ESA 2012 - 20th Annual European Symposium, Ljubljana, Slovenia, September 10-12, 2012. Proceedings*, volume 7501 of *Lecture Notes in Computer Science*, pages 277–288. Springer, 2012. doi: 10.1007/978-3-642-33090-2_25. URL https://doi.org/10.1007/978-3-642-33090-2_25.

Alisa Chang, Badih Ghazi, Ravi Kumar, and Pasin Manurangsi. Locally private k-means in one round. *arXiv preprint arXiv:2104.09734*, 2021.

Anamay Chaturvedi, Huy Nguyen, and Eric Xu. Differentially private k-means clustering via exponential mechanism and max cover. *arXiv preprint arXiv:2009.01220*, 2020.

Albert Cheu, Adam Smith, Jonathan Ullman, David Zeber, and Maxim Zhilyaev. Distributed differential privacy via shuffling. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 375–403. Springer, 2019.

Cynthia Dwork and Aaron Roth. The algorithmic foundations of differential privacy. *Found. Trends Theor. Comput. Sci.*, 9(3-4):211–407, 2014. doi: 10.1561/0400000042. URL https://doi.org/10.1561/0400000042.

Cynthia Dwork, Krishnaram Kenthapadi, Frank McSherry, Ilya Mironov, and Moni Naor. Our data, ourselves: Privacy via distributed noise generation. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 486–503. Springer, 2006.

Úlfar Erlingsson, Vitaly Feldman, Ilya Mironov, Ananth Raghunathan, Kunal Talwar, and Abhradeep Thakurta. Amplification by shuffling: From local to central differential privacy via anonymity. In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 2468–2479. SIAM, 2019.

Dan Feldman, Chongyuan Xiang, Ruihao Zhu, and Daniela Rus. Coresets for differentially private k-means clustering and applications to privacy in mobile sensor networks. In *2017 16th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*, pages 3–16. IEEE, 2017.

William B Johnson and Joram Lindenstrauss. Extensions of lipschitz mappings into a hilbert space. *Contemporary mathematics*, 26(189-206):1, 1984.

Matthew Jones, Huy Lê Nguyen, and Thy Nguyen. Differentially private clustering via maximum coverage. *arXiv preprint arXiv:2008.12388*, 2020.

Haim Kaplan and Uri Stemmer. Differentially private k-means with constant multiplicative error. *arXiv preprint arXiv:1804.08001*, 2018.

Shiva Prasad Kasiviswanathan, Homin K Lee, Kobbi Nissim, Sofya Raskhodnikova, and Adam Smith. What can we learn privately? *SIAM Journal on Computing*, 40(3):793–826, 2011.

Konstantin Makarychev, Yury Makarychev, and Ilya P. Razenshteyn. Performance of johnson-lindenstrauss transform for k-means and k-medians clustering. In Moses Charikar and Edith Cohen, editors, *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing, STOC 2019, Phoenix, AZ, USA, June 23-26, 2019*, pages 1027–1038. ACM, 2019. doi: 10.1145/3313276.3316350. URL https://doi.org/10.1145/3313276.3316350.

Kobbi Nissim and Uri Stemmer. Clustering algorithms for the centralized and local models. In *Algorithmic Learning Theory*, pages 619–653. PMLR, 2018.

Roberto Imbuzeiro Oliveira. Concentration of the adjacency matrix and of the laplacian in random graphs with independent edges. *arXiv preprint arXiv:0911.0600*, 2009.

Uri Stemmer. Locally private k-means clustering. In *Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 548–559. SIAM, 2020.

Joel A. Tropp. User-friendly tail bounds for sums of random matrices. *Found. Comput. Math.*, 12(4):389–434, 2012. doi: 10.1007/s10208-011-9099-z. URL `https://doi.org/10.1007/s10208-011-9099-z`.

Joel A. Tropp. An introduction to matrix concentration inequalities. *Found. Trends Mach. Learn.*, 8(1-2): 1–230, 2015. doi: 10.1561/2200000048. URL `https://doi.org/10.1561/2200000048`.

A Concentration bounds

We recall some basic concentrations bounds that we draw upon for our proofs.

Lemma A.1 (Hoeffding's inequality)**.** *Given n i.i.d. Bernoulli random variables X_i that take values in $\{0,1\}$ with mean p,*

$$P\left(\left|\sum_{i\in[n]} X_i - np\right| > t\right) \leq 2\exp(-2t^2/n).$$

Lemma A.2 (Chernoff bound for Gaussian random variables)**.** *Given n i.i.d. Gaussian random variables $\eta_i \sim N(0,\sigma^2)$,*

$$P\left(\left|\sum_{i\in[n]} \eta_i\right| > t\right) \leq 2\exp\left(\frac{-t^2}{2n\sigma^2}\right).$$

We will also need the following more involved concentration bound to bound the estimation error of the HeavySumsOracle developed later as a tool which allows us to reduce the round complexity of our protocols. We follow the formulation in §1.6.2 of Tropp [2015], who attributes it to Oliveira [2009] and Tropp [2012].

Lemma A.3 (Matrix Bernstein's inequality)**.** *Let $S_1,\ldots,S_n$ be independence centered random matrices with common dimension $d_1 \times d_2$ and assume that each one is uniformly bounded, i.e.*

$$\mathbb{E}[S_k] = 0,$$
$$\|S_k\| \leq L \;\forall k \in [n].$$

Let $Z = \sum_{k=1}^n S_k$ and $v(Z)$ denote the matrix variance statistic of Z i,e,

$$\begin{aligned} v(Z) &= \max\{\|\mathbb{E}(ZZ^*)\|, \|\mathbb{E}Z^*Z\|\} \\ &= \max\left\{\left\|\sum_{k=1}^n \mathbb{E}S_kS_k^*\right\|, \left\|\sum_{k=1}^n \mathbb{E}S_k^*S_k\right\|\right\}. \end{aligned}$$

Then

$$P(\|Z\| \geq t) \leq (d_1+d_2)\cdot\exp\left(\frac{-t^2/2}{v(Z)+Lt/3}\right) \;\forall t \geq 0.$$

Further,

$$\mathbb{E}[\|Z\|] \leq \sqrt{2v(Z)\log(d_1+d_2)} + \frac{1}{3}L\log(d_1+d_2).$$

B Bitstogram and the Heavy Sums Oracle

The contents of this subsection are used in the cost analysis for both clustering algorithms. In the sequel we make extensive use of locally private frequency estimation. For private frequency estimation a lower bound of $\Omega_\epsilon(\sqrt{n})$ is known [Chan et al., 2012]. A state of the art construction for this problem is the Bitstogram algorithm Bassily et al. [2020], which is an ϵ-LDP algorithm for the heavy-hitters problem that achieves low error.

Lemma 2.11 (Algorithm Bitstogram, Bassily et al. [2020])**.** *Let V be a finite domain of values, let $f : D' \to V$, and let $n(v)$ denote the frequency with which v occurs in $f(D')$. Let $\epsilon \leq 1$. Algorithm* Bitstogram(f, ϵ, β) *interacts with the set of n users in 1 round and satisfies ϵ-LDP. Further, it returns a list $L = ((v_i, a_i))_i$ of value-frequency pairs with length $\tilde{O}(\sqrt{n})$ such that with probability $1 - \beta$ the following statements hold:*

1. *For every $(v, a) \in L$, $\|a - f(v)\| \leq E$ where $E = O\left(\frac{1}{\epsilon}\sqrt{n \log(n/\beta)}\right)$.*
2. *For every $v \in V$ such that $f(v) \geq M$, $v \in L$, where $M = O\left(\frac{1}{\epsilon}\sqrt{n \log |V|/\beta \log(1/\beta)}\right)$.*

We overload notation to treat the list returned by Bitstogram *returns as either a set of (heavy-hitter, frequency) pairs or a function which may be queried on a value to return either the corresponding frequency if it is a heavy hitter or a value of 0 otherwise. A subscript of M will denote the upper bound on the maximum frequency omitted. We see that whenever $|V| = \Omega(n)$, $M = \Omega(E)$ and* Bitstogram *promises a uniform error bound of M when estimating the frequency of any element in the co-domain for an appropriate choice of constants.*

We introduce an extension of the Bitstogram algorithm called HeavySumsOracle that allows us to query the sums of some vector valued function over the set of elements that map to a queried heavy-hitter value. For a given value-mapping function $f : \mathcal{X} \to \mathcal{V}$ and a vector-valued function $g : \mathcal{X} \to \mathbb{R}^d$ the sum estimation oracle privately returns for every heavy hitter $v \in \mathcal{V}$ the sum of all agents that map to x, i.e. $\sum_{p:f(p)=x} p$. We recall that Bitstogram is a modular algorithm with two subroutines; a frequency oracle that privately estimates the frequency of any value in the data universe, and a succinct histogram construction that constructs the heavy hitters in a bit-wise manner by making relatively few calls to the frequency oracle. The construction of HeavySumsOracle essentially mimics the frequency oracle construction called Hashtogram from Bassily et al. [2020] and can be run in parallel with Bitstogram, allowing us to reduce the round complexity of our protocols.

1 Public randomness: Uniformly random matrix $Z \in \{\pm 1\}^{|\mathcal{V}| \times n}$
2 Setting: Agent $j \in [n]$ holds $x_j \in \mathcal{X}$, public functions $f : \mathcal{X} \to \mathcal{V}$, $g : \mathcal{X} \to [0, b]^d$, g has known bounded sensitivity $\Delta_{g,2}$.
3 For $j \in [n]$ let $y_j \leftarrow Z[f(x_j), j] \cdot g(x_j) + \eta_j$ for $\eta_j \sim N\left(0, \frac{4c^2\Delta_{g,2}^2}{\epsilon^2}\right)$ where c^2 is according to lemma 2.10
4 On input $v \in \mathcal{V}$ return $S(v) = \sum_{j \in [n]} y_j \cdot Z[v, j]$ and wait for next query

Algorithm 6: HeavySumsOracle

Lemma 2.12 (HeavySumsOracle)**.** *Let $f : \mathcal{X} \to \mathcal{V}$, $g : \mathcal{X} \to B(0, \Delta/2) \subset \mathbb{R}^{d'}$ be some functions where g has bounded sensitivity $\Delta_{g,2}$ and let $D' \subset \mathcal{X}$ be a distributed dataset over n users. With probability at least $1 - \beta$, for every $v \in \mathcal{V}$ that occurs in $f(D')$, if $S(v)$ is the value returned by Algorithm 6 then*

$$\left\| S(v) - \sum_{f(y)=v} g(y) \right\| \leq 2\Delta \sqrt{2n \log \frac{d'+1}{\beta}} + \frac{4c_G \Delta_{g,2}}{\epsilon} \sqrt{2d'n \log \frac{4}{\beta}}.$$

Here c_G is the constant derived from the Gaussian mechanism (lemma 2.10), and $\Delta_{g,2}$ is the ℓ_2-sensitivity of g. Note that since $\Delta_{g,2} \leq \Delta$, this also implies (whenever $\epsilon < c_G = \sqrt{2\ln(1.25/\delta)}$)

$$\left\| S(v) - \sum_{f(y)=v} g(y) \right\| \leq O\left(\frac{c_G \Delta}{\epsilon} \sqrt{d' n \log \frac{1}{\beta}} \right).$$

Further, Algorithm 6 is (ϵ, δ)-LDP.

Proof. Let the data of the jth agent be denoted x_j, and let y_j denote the value sent by the jth agent, i.e. $Z[f(x_j), j] \cdot (g(x_j) + \eta_j)$ where $\eta_j \sim N\left(0, \frac{c_G^2 \Delta^2}{\epsilon^2}\right)$.

$$\begin{aligned}
S(v) &= \sum_{j \in [n]} y_j \cdot Z[v, j] \\
&= \sum_{j \in [n]} (Z[f(x_j), j] \cdot g(x_j) + \eta_j) \cdot Z[v, j] \\
\Rightarrow \mathbb{E}[S(v)] &= \sum_{j \in [n]} \mathbb{E}[Z[f(x_j), j] \cdot Z[v, j] \cdot (g(x_j))] + \mathbb{E}[\eta_j \cdot Z[v, j]]. \\
&= \sum_{j : f(x_j) = v} \mathbb{E}[g(x_j)] + \sum_{j : f(x_j) \neq v} \mathbb{E}[Z[f(v_j), j]] \mathbb{E}[Z[f(x_j), j] \cdot g(x_j)] + 0 \\
&= \sum_{j : f(x_j) = v} g(x_j) + 0.
\end{aligned}$$

This gives us that in expectation, $S(v) = \sum_{y \in D_{f(x)}} g(y)$. Now we derive high probability bounds on the estimation error. Let S denote the quantity of interest, i.e. $\sum_{j : f(x_j) = v} g(x_j)$. We have

$$\begin{aligned}
\|S(v) - S\| &= \left\| \sum_{j \in [n]} (Z[f(x_j), j] \cdot g(x_j) + \eta_j) \cdot Z[v, j] - S \right\| \\
&\leq \left\| \sum_{j : f(x_j) \neq v} b_j g(x_j) + \sum_{j \in [n]} b_j \eta_j \right\| \\
&\leq \left\| \sum_{j : f(x_j) \neq v} b_j g(x_j) \right\| + \left\| \sum_{j \in [n]} b_j \eta_j \right\|,
\end{aligned}$$

where we let b_j denote uniformly random $\{\pm 1\}$ bits. Note that the cancellations of the summands $g(x_j)$ in $S(v)$ (where j was such that $f(x_j) = v$) with S were deterministic, but the Gaussian noise introduced to retain privacy remains for all agents. To bound the first summand, we observe that $b_j g(x_j)$ are at most n independent vectors with ℓ_2 norm at most Δ such that $\mathbb{E}[b_j g(x_j)] = \mathbb{E}[b_j]\mathbb{E}[g(x_j)] = 0$ and matrix variance $\max(\mathbb{E}[\|\langle b_j g(x_j).b_j g(x_j)\rangle\|], \mathbb{E}[\|b_j g(x_j) \otimes b_j g(x_j)\|]) = \|g(x_j)\|^2 \leq \Delta^2$. Then, identifying $b_j g(x_j)$ with S_j in lemma A.3 and bounding the size of the set $\{j : f(x_j) \neq v\}$ by n, we have that

$$P\left(\left\| \sum_{j=1}^{n} S_j \right\| \geq t \right) \leq (d' + 1) \exp\left(\frac{-t^2/2}{\Delta^2 n + \Delta t/3} \right) \forall t \geq 0.$$

For an error probability of at most β, we see that it would suffice to set t such that

$$(d' + 1) \exp\left(\frac{-t^2/2}{\Delta^2 n + \Delta t/3} \right) \leq \beta$$

$$\Leftarrow \frac{\Delta^2 n + \Delta t/3}{t^2/2} \leq \frac{1}{\log(d'+1)/\beta}$$
$$\Leftrightarrow \frac{2\Delta^2 n}{t^2} + \frac{2\Delta}{3t} \leq \frac{1}{\log(d'+1)/\beta}$$

We see that it suffices to let $t > 2\sqrt{2}\Delta\sqrt{n\log(d'+1)/\beta}$. Next, we would like to bound the second summand $\left\|\sum_{j\in[n]} b_j\eta_j\right\|$. We have that

$$\begin{aligned}
\left\|\sum_{j\in[n]} b_j\eta_j\right\|^2 &= \left\langle \sum_{j\in[n]} b_j\eta_j, \sum_{j\in[n]} b_j\eta_j \right\rangle \\
&= \sum_{j\in[n]} \|\eta_j\|^2 + \sum_{j,k\in[n]} b_j b_k \langle \eta_j, \eta_k\rangle \\
&\leq \sum_{j\in[n]}\sum_{d\in[d']} \eta_{j,d}^2 + \sum_{j,k\in[n]} b_j b_k \|\eta_j\|\|\eta_k\| \\
&\leq \sum_{j\in[n]}\sum_{d\in[d']} \eta_{j,d}^2 + \sum_{j,k\in[n]} (\|\eta_j\|^2 + \|\eta_k\|^2) \\
&\leq 2\sum_{j\in[n]}\sum_{d\in[d']} \eta_{j,d}^2
\end{aligned}$$

Since the upper bound is a sum of $d'n$ i.i.d. normal random variables with variance $\sigma^2 = \frac{4c_G^2\Delta_{g,2}^2}{\epsilon^2}$. We can now apply lemma A.2 which gives us

$$P\left(\left\|\sum_{j\in[n]} b_j\eta_j\right\| > 2t_2\right) \leq 2\exp\left(\frac{-t_2^2}{2d'n\sigma^2}\right),$$

where $\sigma^2 = \frac{4c_G^2\Delta_{g,2}^2}{\epsilon^2}$. We again set the error probability to be $\beta/2$ to get

$$\begin{aligned}
2\exp\left(\frac{-t_2^2}{8\sigma^2 d'n}\right) &\leq \frac{\beta}{2} \\
\Leftrightarrow t_2 &\geq \sigma\sqrt{8d'n\log\frac{4}{\beta}}.
\end{aligned}$$

Substituting for σ we get the stated error bound. To see why this routine is (ϵ,δ)-differentially private, we see that the sensitivity of the response $Z[f(x_j), j]\cdot g(x_j)$ is $2\Delta_{g,2}$. The privacy guarantee is hence a direct consequence of lemma 2.10.

□

The objects returned by Bitstogram and HeavySumsOracle are often used in conjunction to estimate the average vector value for collections of data points that accumulate under some value-mapping. The consequent error bound in all these applications is formalized in the following lemma.

Lemma B.1. *Given a function $f : \mathcal{X} \to \mathcal{V}$, and $g : \mathcal{X} \to B(0, \Delta/2) \subset \mathbb{R}^d$, if a succinct histogram $\mathsf{HG} : \mathcal{X} \to \mathbb{R}$ is returned by $\mathsf{Bitstogram}(f, \beta, \epsilon)$ and a sum oracle $\mathsf{SO} : \mathcal{V} \to \mathbb{R}^d$ is returned by $\mathsf{HeavySumsOracle}(f, g, \beta, \epsilon)$, then with total probability $1 - 2\beta$, for every heavy hitter if $v \in \mathsf{HG}$ S_v denotes the sum $\sum_{x\in X: f(x)=v} g(x)$ and n_v denotes its frequency $|\{x \in X : f(x) = v\}|$, the following bound holds*

$$\left\|\frac{\mathsf{SO}(v)}{\mathsf{HG}(v)} - \frac{S_v}{n_v}\right\| \leq \frac{1}{n_v - \mathsf{HG}_E}\cdot\left(\mathsf{SO}_E + \mathsf{HG}_E\left\|\frac{S_v}{n_v}\right\|\right).$$

In the above, as per our convention, HG_E *refers to the error term in the estimation of* HG*, and* SO_E *refers to the error term in the estimation error of* SO*. Note that for every heavy hitter* n_v *we can assume without loss that* $n_v \geq 2\mathsf{HG}_E$*, that* $\left\|\frac{S_v}{n_v}\right\| = O(\Delta)$*, and that* $\mathsf{HG}_E = O((c_G\Delta/\epsilon)\sqrt{dn\log n/\beta})$*, from which it follows that*

$$\left\|\frac{\mathsf{SO}(v)}{\mathsf{HG}(v)} - \frac{S_v}{n_v}\right\| \leq O\left(\frac{c_G\Delta\sqrt{dn\log n/\beta}}{\epsilon n_v}\right).$$

Proof. The proof is a direct consequence of the triangle inequality and some algebra.

$$\begin{aligned}\left\|\frac{\mathsf{SO}(v)}{\mathsf{HG}(v)} - \frac{S_v}{n_v}\right\| &= \left\|\frac{\mathsf{SO}(v)}{\mathsf{HG}(v)} - \frac{S_v}{\mathsf{HG}(v)} + \frac{S_v}{\mathsf{HG}(v)} - \frac{S_v}{n_v}\right\| \\ &\leq \frac{\|\mathsf{SO}(v) - S_v\|}{\mathsf{HG}(v)} + \frac{|n_v - \mathsf{HG}(v)|}{\mathsf{HG}(v)}\left\|\frac{S_v}{n_v}\right\| \\ &\leq \frac{1}{n_v - \mathsf{HG}_E}\cdot\left(\mathsf{SO}_E + \mathsf{HG}_E\left\|\frac{S_v}{n_v}\right\|\right).\end{aligned}$$

□

C Locality Sensitive Hashing

The contents of this subsection are used only for the construction and analysis of the multi-round k-means algorithm with low additive error. We start by recalling the definition of an LSH family.

Definition 2.13 (Locality sensitive hashing (LSH))**.** We say that a family of hash functions $H : \mathbb{R}^d \to B$ for a finite set of buckets B is *locality-sensitive* with parameters (p, q, r, cr) if for every $x, y \in \mathbb{R}^d$ for some $1 \geq p > q \geq 0$, $r > 0$ and $c > 1$

$$P(H(x) = H(y)) \begin{cases} \geq p \text{ if } d(x,y) \leq r \\ \leq q \text{ if } d(x,y) \geq cr. \end{cases}$$

In this work we use an LSH-family construction construction from Andoni and Indyk [2006].

Theorem 2.14. *For every sufficiently large* d *and* n *there exists a family* $\mathcal{H}$ *of hash functions defined on* $\mathbb{R}^d$ *such that for a dataset of size* n*,*

1. *A function from this family can be sampled, stored and computed in time* $t^{O(t)}\log n + O(dt)$*, where* t *is a free positive parameter of our choosing.*
2. *The collision probability for two points* $u, v \in \mathbb{R}^d$ *depends only on the* ℓ_2 *distance between them, which we henceforth denote by* $p(\|u - v\|)$*.*
3. *The following inequalities hold:*

$$\begin{aligned} p(1) &\geq \frac{A}{2\sqrt{t}}\frac{1}{(1+\epsilon+8\epsilon^2)^{t/2}} \\ \forall c > 1,\ p(c) &\leq \frac{2}{(1+c^2\epsilon)^{t/2}} \end{aligned}$$

   where A *is an absolute constant* < 1*, and* $\epsilon = \Theta(t^{-1/2})$*. One can choose* $\epsilon = \frac{1}{4\sqrt{t}}$*.*

4. *The number of buckets* N_B *an LSH function with parameter* t *uses is* $t^{O(t)}\log n$*.*

Note that by scaling the input to the LSH function this gives us constructions for (p, q, r, cr)-sensitive LSH families for arbitrary values of $r > 0$. Due to the occurrence of terms like $t^{O(t)}$ in the collision probabilities and the number of buckets, the performance of an LSH family is very sensitive to the choice of t. In the following lemma we show how to choose a value of t for a desired ratio of $p^2(1)$ to $p(c)$.

Lemma 2.15. *Given a fixed $c > \sqrt{2}$, for any $B > 1$, there is a choice of $t = O\left(\log^2 B\right)$ for the LSH function described in theorem 2.14 such that*

$$\begin{aligned}\frac{p^2(1)}{p(c)} &= \Omega(B),\\ p(1) &\geq \Omega(B^{-1/c'}/\log B),\\ \log N_B &= O(\log^2 B \log\log B + \log\log n),\end{aligned}$$

where $c' = (c^2/8 - 1/4)$. It will be convenient to note that $1/c' = O(1/(2c^2 - 1))$.

Proof. We have that

$$\begin{aligned}\frac{p^2(1)}{p(c)} &\geq \frac{A^2}{8t}\frac{(1 + c^2/4\sqrt{t})^{t/2}}{(1 + (1/4\sqrt{t}) + 1/2t)^t}\\ &\geq \frac{A^2}{8t}\frac{\exp\left(c^2/4\sqrt{t} - c^4/32t\right)^{t/2}}{\exp(1/4\sqrt{t} + 1/2t)^t}\\ &\geq \frac{A^2}{8t}\exp((c^2/8 - 1/4)\sqrt{t} - c^4/64 + 1/2)\\ &= \Omega\left(\exp(c'\sqrt{t})/t\right).\end{aligned}$$

where $c' = (c^2/8 - 1/4)$. It follows that for $t = (\log B + \log\log B)^2/(c')^2$,

$$\begin{aligned}\frac{p^2(1)}{p(c)} &\geq (c')^2\frac{B + \log B}{(\log B + \log\log B)^2}\\ &= \Omega(B)\\ p(1) &\geq \frac{A}{2\sqrt{t}}\exp(-\sqrt{t}/8 - 1/4)\\ &= \Omega(B^{-1/c'}/\log B)\\ \log N_B &= O(t\log t + \log\log n)\\ &= O(\log^2 B\log\log B + \log\log n).\end{aligned}$$

□

In the construction of the multi-round k-means algorithm with low additive error, we will need to estimate the average of all points that map to a given heavy bucket. Due to the pair-wise nature of the LSH guarantee, the analysis of this requires us to use an arbitrary point from the bucket as a filter to ensure that sufficiently many points close to it and not too many points far from it map to that bucket. This result and its proof follow the lines of a similar result by Nissim and Stemmer [2018], but are modified to allow for the possibility of false positives and have been phrased differently.

Lemma 2.16. *Let $C \subset D$ be a set of points with diameter r and let the diameter of D be Δ. For any $x_0 \in C$, if $\hat{x_0}$ is the average over all points colliding with x_0 under a $(p(1), p(c), r, rc)$-sensitive LSH function H applied to D, then with probability $p(1)/4$,*

$$\|x_0 - \hat{x}_0\| \leq cr + \frac{8p(c)|D|}{p^2(1)|C|}\Delta,$$

and the number of points of C that collide with x_0 is at least $\frac{p(1)C}{2}$.

Proof. Let x_0 be an arbitrary fixed point in C. Let $N \subset C$ be the set of points that lie near x_0 and collide with it under the LSH function, i.e. $N = \{y \in D : H(y) = H(x_0), d(y, x_0) \leq r\}$. Since $\forall y \in C$, $d(x_0, y) < r$, $\mathbb{E}[|N|] \geq p(1)|C|$. We note that $|N|$ is supported on $\{0, \ldots, |C|\}$ and let $p := P(|N| \geq \frac{p(1)}{2}|C|)$. Then we can write

$$\begin{aligned}
\mathbb{E}[|N|] &= \sum_{i=0}^{|C|} P(|N| = i) \cdot i \\
&= \sum_{i=0}^{\lceil \frac{p(1)}{2}|C| \rceil - 1} P(|N| = i) \cdot i + \sum_{i=\lceil \frac{p(1)}{2}|C| \rceil}^{|C|} P(|N| = i) \cdot i \\
&\leq (1-p) \cdot \frac{p(1)|C|}{2} + p \cdot |C| \\
\Rightarrow p(1)|C| &\leq \frac{p(1)|C|}{2} + p|C|(1 - \frac{p(1)}{2}) \\
\Rightarrow p &\geq \frac{p(1)}{2 - p(1)}.
\end{aligned}$$

Let $F \subset C$ be the set of all points which lie far from x_0 and collide with it under H, i.e. $\{y \in D : H(y) = H(x_0), d(y, x_0) \geq cr\}$. It again follows from the LSH guarantee that $\mathbb{E}[|F|] \leq p(c)|D| \Leftrightarrow \mathbb{E}[|D \backslash F|] \geq (1 - p(c))|D|$. If $q := P(|D \backslash F| \geq (1 - \frac{4p(c)}{p(1)})|D|)$, then we can write

$$\begin{aligned}
\mathbb{E}[|D \backslash F|] &= \sum_{i=0}^{|D|} P(|D \backslash F| = i)] \cdot i \\
&\leq (1-q) \cdot \left(1 - \frac{4p(c)}{p(1)}\right)|D| + q \cdot |D| \\
\Rightarrow (1 - p(c))|D| &\leq (1-q)\left(1 - \frac{4p(c)}{p(1)}\right)|D| + q|D| \\
\Rightarrow (1 - p(c))|D| &\leq \left(1 - \frac{4p(c)}{p(1)}\right)|D| + q\frac{4p(c)}{p(1)}|D| \\
\Rightarrow \left(\frac{4}{p(1)} - 1\right)p(c) &\leq q\frac{4p(c)}{p(1)} \\
\Rightarrow 1 - \frac{p(1)}{4} &\leq q.
\end{aligned}$$

With probability at least $1 - p(1)/4$, $|F| < 4p(c)|D|/p(1)$, and with probability at least $p(1)/2$, $|N| \geq p(1)|C|/2$. Applying the union bound on the negation of these events it follows that with probability at least $p(1)/4$ both these events hold. Conditioning on their intersection, we want to bound the distance between the average $\hat{x}_0$ of all points that collide with x_0 and x_0 itself. Let $N' = \{y \in D : H(y) = H(x_0), d(y, x_0) < cr\}$, with which definition we have $|N'| \geq |N| \geq p(1)|C|/2$. Further, the set of all points that collide with x_0 are partitioned by N' and F. It follows that

$$\begin{aligned}
\|x_0 - \hat{x}_0| &= \left\| x_0 - \frac{\sum_{y \in N'} y + \sum_{y \in F} y}{|N'| + |F|} \right| \\
&\leq \frac{\sum_{y \in N'} \|x_0 - y\| + \sum_{y \in F} \|x_0 - y\|}{|N'| + |F|} \\
&\leq \frac{cr|N'| + \Delta|F|}{|N'| + |F|}
\end{aligned}$$

$$\leq cr + \frac{|F|}{|N'|}\Delta$$
$$\leq cr + \frac{8p(c)|D|}{p(1)^2|C|}\Delta.$$

□

D Miscellaneous tools

In the course of our analysis, we will make extensive use of two weak triangle inequalities which hold for the distance function d.

Lemma D.1 (Weak triangle inequalities)**.** *1. Given points p, q and $r \in \mathbb{R}^d$ such that $z(q,r) \leq c\alpha^2 z(p,q)$, where c is some constant and $\alpha < 1$,*

$$z(p,r) \leq (1 + O(\alpha))z(p,q).$$

2. Given arbitrary points p, q and $r \in \mathbb{R}^d$,

$$z(p,r) \leq 2z(p,q) + 2z(q,r).$$

Proof. 1. The bound follows from an application of the triangle inequality for the ℓ_2 norm.

$$\begin{aligned}
&\sqrt{z(p,r)} \leq \sqrt{z(p,q)} + \sqrt{z(q,r)} \\
\Rightarrow\ &z(p,r) \leq z(p,q) + 2\sqrt{z(p,q)z(q,r)} + z(q,r) \\
&\leq z(p,q) + 2\alpha\sqrt{c}z(p,q) + c\alpha^2 z(p,q) \\
&\leq (1 + O(\alpha))z(p,q).
\end{aligned}$$

2. The bound follows from an application of the triangle inequality for the ℓ_2 norm and the A.M.-G.M. inequality.

$$\begin{aligned}
&\sqrt{z(p,r)} \leq \sqrt{z(p,q)} + \sqrt{z(q,r)} \\
\Rightarrow\ &z(p,r) \leq z(p,q) + 2\sqrt{z(p,q)z(q,r)} + z(q,r) \\
&\leq 2z(p,q) + 2z(q,r).
\end{aligned}$$

□

Lemma D.2. *Let there be a set U and a family of subsets $\mathcal{S} \subset 2^U$ such that some subfamily $\mathcal{Z} \subset \mathcal{S}$ covers U, that is*

$$\bigcup_{z \in \mathcal{Z}} z = U.$$

If we pick a collection of sets $\mathcal{C} = \{c_1, \ldots, c_Y\} \subset \mathcal{S}$ where $Y = \lceil 2|\mathcal{Z}| \log(1/\alpha) \rceil$ such that for each $i \in [Y]$

$$U_i = U \setminus \bigcup_{j=1}^{i-1} c_j$$
$$|c_i \cap U_i| \geq \frac{\max_{c \in \mathcal{S}} |c \cap U_i|}{2}$$

then $|\bigcup_i \in [Y] c_i| \geq (1 - \alpha)|U|$.

We refer the reader to Lemma 2.7, Chaturvedi et al. [2020] for a proof of lemma D.2.

Lemma D.3. *Given a k-means clustering $D'_1, \ldots, D'_k$ of a data set D' where*

$$\mu_j = \frac{\sum_{p \in D'_j} p}{|D'_k|},$$

if $\hat{\mu}_j$ is an estimate for μ_j then the k-means clustering cost with respect to $\{\hat{\mu}_j : j = 1, \ldots, k\}$ for any cluster D'_j can be bounded by

$$f_{D'_j}(\{\hat{\mu}_j : j = 1, \ldots, k\}) \leq f_{D'_j}(\mu_j) + \left|D'_j\right| \left\|\mu_j - \hat{\mu}_j\right\|^2.$$

Proof. First we observe that $f_{D'_j}(\{\hat{\mu}_j : j = 1, \ldots, k\}) \leq f_{D'_j}(\hat{\mu}_j)$ by definition. To get the stated bound, we perform a couple of algebraic manipulations.

$$\begin{aligned}
f_{D'_j}(\hat{\mu}_j) &= \sum_{p' \in D'_j} z(p', \hat{\mu}_j) \\
&= \sum_{p' \in D'_j} \left\|p' - \hat{\mu}_j\right\|^2 \\
&= \sum_{p' \in D'_j} \left\|p' - \mu_j + \mu_j - \hat{\mu}_j\right\|^2 \\
&= \sum_{p' \in D'_j} \langle p' - \mu_j + \mu_j - \hat{\mu}_j, p' - \mu_j + \mu_j - \hat{\mu}_j \rangle \\
&= \sum_{p' \in D'_j} \left\|p' - \mu_j\right\|^2 + 2\langle p' - \mu_j, \mu_j - \hat{\mu}_j \rangle + \left\|\mu_j - \hat{\mu}_j\right\|^2 \\
&= f_{D'}(\{\mu_j : j \in [k]\}) + \left|D'_j\right| \left\|\mu_j - \hat{\mu}_j\right\|^2.
\end{aligned}$$

□