THE POWER OF INPUT PERTURBATION: DIFFERENTIALLY PRIVATE LEARNING FROM PAIRWISE DISTANCES

Anonymous authors
Paper under double-blind review

ABSTRACT

Differential privacy (DP) has been extensively studied for metric learning problems within specific geometric contexts, such as Euclidean space. However, in many applications, the input metric is given as a matrix of pairwise distances. This paper systematically investigates the power of *input perturbation* for pairwise distances in general metric spaces under the ρ-distance DP model. The approach is remarkably simple: we add independent Gaussian noise to the pairwise distance entries, optionally repair the matrix to the nearest metric, and release the matrix for downstream tasks. This mechanism offers significant practical utility, as a single privatized distance matrix can be reused across various downstream applications. Despite the simplicity, we prove that this method achieves a nearly tight $\tilde{O}(\rho\sqrt{n\log(1/\delta)}/\varepsilon)$ additive error for (ε, δ)-DP metric release. We further apply this approach to a host of metric learning tasks, including minimum spanning trees (MST), k-clustering, hierarchical clustering (HC), and coresets, yielding bounds that are either tight or match the state of the art. Experiments across 12 datasets demonstrate the advantages of this approach in utility and time complexity compared to existing benchmarks. The experiments also reveal error-efficiency trade-offs via different metric repair strategies for practical private metric release.

1 INTRODUCTION

Pairwise distances provide a common language for describing relationships in data. Many distance measures are adopted in applications to capture relationships among different types of objects. For example, edit distance counts the necessary edits to transform one sequence into another (Levenshtein, 1966), shortest path distance captures how closely nodes are connected in a network (Dijkstra, 1959), Gromov-Hausdorff distance measures the distortion needed to match two shapes (Gromov, 1981). Metric spaces capture the common structure of distance measures through properties including non-negativity, reflexivity, symmetry and the triangle inequality. Many fundamental optimization problems evaluate solutions using only pairwise distances and thus admit natural formulations in metric spaces. Examples include clustering and facility location (Arya et al., 2004), nearest-neighbor search (Clarkson, 1999), minimum spanning trees (Kruskal, 1956), the traveling salesman problem (Christofides, 1976), and optimal transport (Kantorovich, 1942), among others. Solving these problems using only pairwise distances is particularly useful when coordinates are either difficult to obtain or highly distorted. One application scenario is that recommendation systems can derive distances between items from co-click records without first learning item embeddings. This motivates studying these problems when the only available representation of the objects is their distance matrix.

In this paper, we study private computation in metric spaces when the distance matrix is the only input. Distances may carry sensitive information such as social relationships. A change to one individual's record can alter her distance from everyone else; therefore, information about that individual spreads across the matrix. Privacy concerns also extend to the optimization results computed from these distances (Gupta et al., 2010). This motivates the development of mechanisms with a differential privacy guarantee (Dwork et al., 2006) for data in a metric space, thereby protecting an individual's position relative to others.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

Private computation in Euclidean space has been extensively studied. Results include private algorithms for geometric range counting (Muthukrishnan & Nikolov, 2012), distance estimation (Kenthapadi et al., 2013), and coreset construction (Feldman et al., 2009). Clustering has received particular attention, including locating small clusters (Nissim et al., 2016) and obtaining strong approximation guarantees in the central, local, and shuffle models (Ghazi et al., 2020; Chang et al., 2021; Cohen-Addad et al., 2022; Dupré la Tour et al., 2024). Under the privacy notion of an arbitrary point replacement, however, clustering incurs an unavoidable additive error that depends on the domain diameter. To address this dependence, Epasto et al. (2023) proposed distance-based privacy, which protects movements of a point within a prescribed radius ρ. Subsequent work studies Euclidean minimum spanning trees under this assumption (Zou et al., 2026). Distance-based privacy fits our distance-matrix input naturally, and we therefore adopt this model for general metric spaces. More background and related work are discussed in Appendix B.

One challenge with this setting is how to obtain accurate private algorithms when only pairwise distances are available. Input perturbation, namely, adding independent noise to these distances, provides a direct approach. Without coordinates, however, it is less clear what alternatives are available or how they might improve accuracy. Even for input perturbation, the noisy distances may violate the triangle inequality, and their effect on the solution quality depends on the objective. This motivated us to study both the limits of input perturbation and the opportunities offered by the metric structure, through several fundamental problems in metric space: metric release, minimum spanning trees, coresets, and clustering.

Privacy model. Our input specifies pairwise distances without a prescribed ambient space in which point movement is measured. We therefore formulate the bounded-movement privacy of Epasto et al. (2023) directly on distance matrices.

Let $\mathcal{M}_n$ contain all symmetric, nonnegative matrices with zero diagonal that satisfy the triangle inequality. We define the distance-based differential privacy below.

Definition 1.1 (Distance-based differential privacy)**.** For $\rho > 0$, matrices $D, D' \in \mathcal{M}_n$ are ρ-neighbors, denoted by $D \sim_\rho D'$, if some $i \in [n]$ satisfies

$$D_{jk} = D'_{jk} \quad \text{for all } j, k \neq i, \qquad \max_{j \in [n]} |D_{ij} - D'_{ij}| \leq \rho.$$

A randomized algorithm $\mathcal{A}$ satisfies (ε, δ) differential privacy if, for every $D \sim_\rho D'$ and every measurable output event S,

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

The case of $\delta = 0$ is called ε-pure privacy.

This definition captures bounded point movement using only pairwise distances. If one point moves by at most ρ, each of its distances to the other points changes by at most ρ.

Our contributions. We demonstrate the *power of input perturbation* when the only available information is the pairwise distance matrix.

First, we establish nearly tight bounds for private metric release, showing that input perturbation achieves nearly optimal accuracy. Second, we develop algorithms based solely on post-processing the privatized distance matrix, with applications to minimum spanning trees, k-clustering, hierarchical clustering, and coreset construction. For each problem, we prove that the simple algorithm achieves near-tight utility bounds, which means the "input perturbation + post-processing" pipeline is powerful and versatile. Third, we evaluate our algorithms on real and synthetic datasets to examine their accuracy, runtime, and the effect of private metric release on downstream tasks.

The following statement shows our results on private metric release.

Result 1 (Informal of Theorems 1 and 2)**.** An algorithm that adds independent Gaussian noise with $\sigma = \Theta\left(\frac{\rho}{\varepsilon}\sqrt{n\log(1/\delta)}\right)$ to pairwise distances satisfies (ε, δ)-DP under the ρ-distance DP model and induce entry-wise $\widetilde{O}(\rho\sqrt{n\log(1/\delta)}/\varepsilon)$ error. With an additional step of metric repair, a pseudometric distance matrix can be obtained with the same error guarantee.

Furthermore, under certain parameter conditions, every (ε, δ)-DP mechanism under the ρ-distance DP model incurs maximum entry-wise error $\Omega(\rho\sqrt{n}/\varepsilon)$ on some input.

A pseudometric allows distinct indices to have zero distance. For private metric release and downstream tasks that require a pseudometric structure, the repair step is necessary. Privacy is already guaranteed by the noise addition and is preserved by the step of metric repair as post-processing.

Application to downstream tasks. We obtain nearly tight bounds for minimum spanning trees, single-linkage merge heights, and the hierarchical clustering dissimilarity objective through input perturbation followed by post-processing. The same methodology, however, yields tight bounds for k-clustering only when k is a constant. We also establish an upper bound for coreset construction. We summarize these results as follows.

Result 2. Post-processing the distance matrix released in Result 1 can lead to the follows.

- For (ε, δ)-DP minimum spanning trees, an upper bound of $\widetilde{O}(\rho n^{3/2}/\varepsilon)$, which is tight up to logarithmic terms (Theorem 3).
- For (ε, δ)-DP k-clustering for constant k, $(O(1), \widetilde{O}(\rho n/\varepsilon))$-approximation for k-median and $(O(1), \widetilde{O}(\rho^2 n/\varepsilon^2))$ for k-means, which are tight up to logarithmic terms (Theorem 5).
- For (ε, δ)-DP Hierarchical clustering, nearly-tight upper bounds under per-merge, merge height, and a dissimilarity function of Cohen-Addad et al. (2019) (Theorems 9 and 10 and proposition H.2).
- For (ε, δ)-DP coreset, $(O(1), \widetilde{O}(\rho n^{3/2}/\varepsilon))$-approximation for k-median and $(O(1), \widetilde{O}(\rho^2 n^2/\varepsilon^2))$-approximation for k-means with $\widetilde{O}(k^2)$ points (Theorem 12).

Zou et al. (2026) studies private MST in Euclidean space, which uses coordinates and allows for $(1+\gamma)$ multiplicative approximation to obtain a smaller additive error. We study general metric distance matrices and obtain a purely additive guarantee of $\widetilde{O}(\rho n^{1.5}/\varepsilon)$. Their lower bound construction can be adapted to our metric distance matrix setting. We also give a direct two-star construction that provides a self-contained proof of the same lower bound. The k-clustering algorithms in Result 2 require exhaustive search over partitions and lead to exponential time. We give polynomial-time algorithms for k-median through recursive metric compression. For fixed $0 < \eta < 1/2$, this method gives an $(O_\eta(1), \widetilde{O}_\eta(\rho n^{1+\eta} k^{1/2-\eta}/\varepsilon))$-approximation for k-median. Smaller η improves the dependence on n but requires deeper recursion and larger approximation factors. We believe this approach also works for k-means and leave this as a future direction.

Table 1 demonstrates the detailed accuracy guarantees and running times, together with lower bounds of all problems mentioned in Result 2.

Problem	Multi error	Additive error	Additive lower bound	Running time
Minimum spanning tree	1	$\widetilde{O}(\rho n^{3/2}/\varepsilon)$	$\Omega(\rho n^{3/2}/\varepsilon)$	$\widetilde{O}(n^2)$
Single-linkage merge heights	1	$\widetilde{O}(\rho\sqrt{n}/\varepsilon)$	$\Omega(\rho\sqrt{n}/\varepsilon)$	$\widetilde{O}(n^2)$
HC dissimilarity objective	1	$\widetilde{O}(\rho n^3/\varepsilon)$	$\Omega(\rho n^3/\varepsilon)$	Exponential
k-median	2	$\widetilde{O}(\rho n\sqrt{k}/\varepsilon)$	$\Omega(\rho n \log k/\varepsilon)$	Exponential
k-median (recursive)	$O_\eta(1)$	$\widetilde{O}_\eta(\rho n^{1+\eta} k^{1/2-\eta}/\varepsilon)$	$\Omega(\rho n \log k/\varepsilon)$	$O_\eta(n^3 + k^2 n^2 \log n)$
k-means	16	$\widetilde{O}(\rho^2 nk/\varepsilon^2)$	$\Omega(\rho^2 n \log^2 k/\varepsilon^2)$	Exponential
(k, z)-coreset	$1 \pm \eta$	$\widetilde{O}_z(\rho^z n^{1+z/2}/(\eta^{z-1}\varepsilon^z))$	—	$\widetilde{O}_z(n^3)$

Table 1: Downstream applications by post-processing the metric in Result 1.

Experiments. We evaluate our algorithms on 12 real and synthetic datasets on the tasks of metric release, MST and k-clustering. The experiments compare accuracy and runtime across several metric repair methods and examine their effects on downstream tasks. For Euclidean MST, we compare with Zou et al. (2026) and show that the input perturbation followed by Kruskal's algorithm achieves lower error and runs faster. Our code is available online.[1]

[1] https://anonymous.4open.science/r/dp-metric-space/

2 PRELIMINARIES

We recall standard tools from differential privacy (Dwork et al., 2014). Throughout this section, privacy and sensitivity are with respect to the ρ-neighboring relation in Definition 1.1.

Definition 2.1 (Sensitivity)**.** The ℓ_2-sensitivity of a function $f : \mathcal{M}_n \to \mathbb{R}^m$ is

$$\Delta_2(f) = \sup_{D \sim_\rho D'} \|f(D) - f(D')\|_2.$$

Proposition 2.1 (Gaussian mechanism)**.** *Let $f : \mathcal{M}_n \to \mathbb{R}^m$ have finite ℓ_2-sensitivity $\Delta_2(f)$. For $0 < \varepsilon < 1$ and $0 < \delta < 1$, the mechanism $\mathcal{A}(D) = f(D) + Z$, where $Z \sim \mathcal{N}(0, \sigma^2 I_m)$, is (ε, δ)-DP if*

$$\sigma \geq \frac{\sqrt{2\ln(1.25/\delta)}\Delta_2(f)}{\varepsilon},$$

Proposition 2.2 (Post-processing)**.** *Let $\mathcal{A} : \mathcal{M}_n \to \mathcal{Y}$ be an (ε, δ)-DP algorithm. For any deterministic or randomized mapping $g : \mathcal{Y} \to \mathcal{Z}$, the composition $g \circ \mathcal{A}$ is (ε, δ)-DP.*

Proposition 2.3 (Adaptive composition)**.** *Let $\mathcal{A}_1, \ldots, \mathcal{A}_T$ be randomized algorithms applied sequentially to the same input. Suppose that, for every $t \in [T]$ and every fixed sequence of preceding outputs $y_1, \ldots, y_{t-1}$, the algorithm $\mathcal{A}_t(\cdot; y_1, \ldots, y_{t-1})$ is $(\varepsilon_t, \delta_t)$-DP. Then releasing $(Y_1, \ldots, Y_T)$, where $Y_t = \mathcal{A}_t(D; Y_1, \ldots, Y_{t-1})$, satisfies $(\sum_{t=1}^T \varepsilon_t, \sum_{t=1}^T \delta_t)$-DP.*

Our lower bounds use group privacy to show that a private algorithm cannot be accurate on all constructed inputs.

Proposition 2.4 (Group privacy)**.** *Let $\mathcal{A}$ be (ε, δ)-DP. Suppose $D_0, \ldots, D_h \in \mathcal{M}_n$ satisfy $D_{t-1} \sim_\rho D_t$ for every $t \in [h]$, where $h \geq 1$ is an integer. For every measurable event S,*

$$\Pr[\mathcal{A}(D_0) \in S] \leq e^{h\varepsilon} \Pr[\mathcal{A}(D_h) \in S] + \delta \sum_{t=0}^{h-1} e^{t\varepsilon}.$$

Additional technical tools used in the appendices are collected in Appendix A.

3 PRIVATE METRIC RELEASE

We first consider privatizing the entire metric space, that is, releasing all $\binom{n}{2}$ pairwise distances privately. We add Gaussian noise and then project the perturbed distances onto the cone of pseudometric distance matrices. This simple approach gives $\widetilde{O}(\rho\sqrt{n}/\varepsilon)$ worst-case additive error to pairwise distances. Next, we show that this error bound is optimal.

3.1 UPPER BOUND

Given the distance matrix D and privacy parameters ε, δ, we propose a mechanism that proceeds in two steps. Let $\mathcal{M}_n$ denote the cone of valid distance matrices: symmetric, nonnegative, zero diagonal, and satisfying the triangle inequality. It is closed and convex.

Algorithm 1. `DP-Release` $(D, \texttt{REP})$: input perturbation to the distance matrix
Input: a metric distance matrix D of dimension $n \times n$; a flag $\texttt{REP} \in \{\texttt{True}, \texttt{False}\}$
Output: a symmetric noisy matrix; a pseudometric distance matrix if $\texttt{REP} = \texttt{True}$

1. Choose parameter $\sigma = \Theta\left(\frac{\rho\sqrt{n\log(1/\delta)}}{\varepsilon}\right)$.
2. Compute a noisy matrix $\tilde{D} = D + Z$, where Z is symmetric with $Z_{ii} = 0$ and off-diagonal entries $Z_{ij} \sim \mathcal{N}(0, \sigma^2)$ drawn independently for $i < j$.
3. Repair and projection:
   - If $\texttt{REP} = \texttt{True}$, project $\tilde{D}$ back to the cone of valid metrics, return $\hat{D} = \operatorname{argmin}_{M \in \mathcal{M}_n} \|M - \tilde{D}\|_\infty$.

- Otherwise, simply release $\tilde{D}$.

Adding Gaussian noise may produce negative distances or violate the triangle inequality. We repair the noisy matrix by finding a nearest pseudometric under the ℓ_∞ norm. This is a metric-nearness problem that can be written as a linear program with $O(n^3)$ triangle constraints (Brickell et al., 2008). The repaired matrix may assign zero distance to distinct indices.

Theorem 1. *For $n \geq 2$, $0 < \varepsilon \leq 1$, and $0 < \delta < 1/2$, Algorithm 1 satisfies (ε, δ)-DP under the ρ-distance model, and with probability at least $1 - \beta$, outputs $\hat{D}$ that satisfies*

$$\alpha \;:=\; \max_{i,j} |\hat{D}_{ij} - D_{ij}| \;=\; O\left(\frac{\rho\sqrt{n\log(1/\delta)(\log n/\beta)}}{\varepsilon}\right).$$

Proof. We first bound the ℓ_2-sensitivity of the distance matrix, viewed as the vector of its $\binom{n}{2}$ upper diagonal entries. If $D \sim_\rho D'$, these vectors differ in at most $n-1$ entries, and each difference is at most ρ. Thus $\Delta_2 \leq \rho\sqrt{n-1}$. By the Gaussian mechanism, $\sigma \geq \frac{\Delta_2}{\varepsilon}\sqrt{2\ln(1.25/\delta)}$ guarantees (ε, δ)-DP under the ρ-distance model, and substituting Δ_2 yields the stated σ. The projection in the second step is post-processing and incurs no additional privacy loss.

For accuracy, note that the true matrix D lies in $\mathcal{M}_n$ and is therefore feasible for the projection, so the optimality of $\hat{D}$ gives $\|\hat{D} - \tilde{D}\|_\infty \leq \|D - \tilde{D}\|_\infty$. Hence

$$\|\hat{D} - D\|_\infty \leq \|\hat{D} - \tilde{D}\|_\infty + \|\tilde{D} - D\|_\infty \leq 2\|Z\|_\infty,$$

and the error is bounded by twice the maximum noise magnitude. Since Z contains fewer than n^2 independent Gaussian variables with variance σ^2, the standard Gaussian tail bound and a union bound give $\|Z\|_\infty = O(\sigma\sqrt{\log(n/\beta)})$ with probability at least $1 - \beta$. Substituting σ gives the claimed bound on α. □

Remark 1. The error bound for the repair step follows only from the optimality of $\widehat{D}$ and the triangle inequality of the norm, so the same argument applies to any norm. For example, for a nearest pseudometric in the Frobenius norm, the same argument gives $\|\widehat{D} - D\|_F \leq 2\|\widetilde{D} - D\|_F$. This projection can be formulated as a convex quadratic program. Practical algorithms for metric nearness and more general metric-constrained optimization have been developed in (Sonthalia & Gilbert, 2022; Li et al., 2023; Tang et al., 2025).

3.2 Lower Bound

We now show that the bound of Theorem 1 is tight up to logarithmic factors. The argument reduces metric release to answering 1-way marginals on a database, for which the following result is known.

Proposition 3.1 (Lower Bound for 1-Way Marginals (Bun et al., 2014; Steinke & Ullman, 2016))**.** *Let $k, N \in \mathbb{N}$ and $\varepsilon \leq 1$. Consider a database $Z \in (\{0,1\}^k)^N$ of N users, and k counting queries $q_1, \ldots, q_k$, where $q_m(Z) = \sum_{u=1}^N Z_{u,m}$. Let $\mathcal{W}$ be any (ε, δ)-DP mechanism, under standard Hamming adjacency where one user's row changes. If $\delta = o(1/N)$, then there exists a database Z on which the expected maximum additive error satisfies*

$$\mathbb{E}\left[\max_{m\in[k]} |\mathcal{W}(Z)_m - q_m(Z)|\right] \geq \Omega\left(\min\left(N, \frac{\sqrt{k}}{\varepsilon}\right)\right).$$

The idea is to encode a database by the position of a single point, such that one user of the database corresponds to one ρ-move of that point, and reading off the distances from it recovers all the marginals at once. Formally, we have the lower-bound result as follows.

Theorem 2. *Let $\mathcal{A}$ be any (ε, δ)-DP algorithm under the ρ-distance model with $\varepsilon \leq 1$ and $\delta = o(\varepsilon/\sqrt{n})$. There exists a metric space $(\mathcal{X}, d)$ on n points such that the expected maximum additive error on the released distance matrix satisfies*

$$\mathbb{E}\left[\max_{i,j\in[n]} |d(x_i, x_j) - \hat{D}_{ij}|\right] = \Omega\left(\frac{\rho\sqrt{n}}{\varepsilon}\right).$$

We defer the proof to Appendix C.

4 DP MINIMUM SPANNING TREES

We apply the private metric release to the minimum spanning tree problem. The output of a mechanism M is a spanning tree on the public indices. Define $w_D(T) = \sum_{\{i,j\}\in T} D_{ij}$ as the total weight of the edges in tree T, $\mathrm{MST}(D) = \min_T w_D(T)$ as the minimum weight over all spanning trees T, and $\mathrm{err}(M, D) = \mathbb{E}[w_D(M(D))] - \mathrm{MST}(D)$ as the additive error of the total weight of the output tree $M(D)$ operating on the true weight of D, compared with the true MST cost on D.

MST with metric release. We use input perturbation and release the perturbed distance matrix $\widetilde{D} = D + Z$ from Algorithm 1 with `REP = False`, i.e., no repair. Run Kruskal's algorithm on $\widetilde{D}$. Note that Kruskal's algorithm allows negative weights and does not require the triangle inequality. For every noise realization, if T^* is an MST of D and $\widehat{T}$ is an MST of $\widetilde{D}$, then by Theorem 1,

$$w_D(\widehat{T}) - w_D(T^*) \le 2(n-1)\|Z\|_\infty.$$

Consequently, the algorithm is (ε, δ)-DP under the ρ-distance model, and has expected additive error $O\left(\frac{\rho n^{3/2}}{\varepsilon}\sqrt{\log(1/\delta)\log(n)}\right)$, and runs in $O(n^2 \log n)$ time by sorting edges in $\widetilde{D}$. Replacing $\log(n)$ by $\log(n/\beta)$ gives the corresponding bound with probability $1-\beta$, leading to the following.

Theorem 3. *For a metric space $(\mathcal{X}, d)$ and the pairwise distance matrix D, running standard MST algorithm (e.g., Kruskal) on the matrix released by Algorithm 1, with or without metric repair, produces a spanning tree that is (ε, δ)-DP under the ρ-distance model and has additive error of*

$$O\left(\frac{\rho n^{3/2}}{\varepsilon}\sqrt{\log(1/\delta)\log(n/\beta)}\right),$$

with probability at least $1-\beta$.

A similar result has been proven by a recent work Zou et al. (2026) for Euclidean space, and (Hladík & Tětek, 2025) studied DP algorithms for MST using a different neighboring notion. The upper bound is nearly tight: in Appendix D, we present a lower bound proof for the $\Omega(\frac{\rho n^{1.5}}{\varepsilon})$ additive error. We note that the construction and techniques in (Zou et al., 2026) can also lead to the same lower bound, and we include the proof for completeness and readability reasons.

5 DP k-CLUSTERING, HIERARCHICAL CLUSTERING, AND CORESETS

An important family of downstream tasks is clustering. In k-clustering, we aim to partition the input elements into k clusters such that the quality of the clusters is optimized (e.g., k-means and k-medians). In hierarchical clustering, we aim to find a hierarchical decomposition such that the distances match the separation of the elements in the hierarchy. Another popular application is computing a coreset, a subset of the input data whose clustering structure is similar to that of the optimal clustering structure. All results share the theme of using input perturbation for metric release and applying clustering algorithms on the perturbed distances. Due to space limitations, the details are deferred to the appendix (Appendix E, Appendix G, Appendix H, Appendix I). We briefly explain the main technical ideas and contributions.

k-clustering. We presented several DP mechanisms that all achieve k-clustering results with a constant multiplicative error and an additive error which depends on ρ, n, k, ε (See Table 1). First of all, if running time is not a concern, one can enumerate all possible k-clustering solutions and pick the best one using the released perturbed distances. Note that since the distances are perturbed, we cannot find the best center for each cluster. Rather, the sum of the average distances from each point to all other points in the same cluster is a constant approximation of the optimal cost and can serve as a surrogate function. We present an analysis of upper bounds on the additive error using this scheme, as well as matching lower bounds. Next, we show k-clustering DP algorithms of polynomial running time. Here we introduce several new ideas. First, we show that the local search method still works for perturbed distances that satisfy the triangle inequality and symmetry. Then we use a recursive clustering algorithm. After we partition the input points (arbitrarily) into groups, we recursively compute the k-clustering in each group. These clusters are treated as super nodes and a k-clustering on these super nodes can be expanded to be a k-clustering with bounded cost. We

show that the multiplicative error is still a constant C_η with a parameter η, the running time becomes fully polynomial, and the additive error can be made to be proportional to $n^{1+\eta}$. Finally, we show that if we only allow additive errors (i.e., multiplicative factor =1), the additive lower bound scales in $\rho n^2/\varepsilon$ for pure DP and $\rho n^{3/2}/\varepsilon$ for (ε, δ)-DP. In both settings, these bounds are super linear in n (unlike the near linear bound if constant multiplicative approximation factors are allowed).

Hierarchical clustering. Similarly, we can apply various hierarchical clustering (HC) algorithms on the private metric released. We studied HC algorithms with two families of algorithms: hierarchical agglomerative clustering (HAC) and HC with objective functions. For HAC, we studied two error notions, a variant of the per-merge error ((Lavastida et al., 2021; Dhulipala et al., 2022)) and the merge height error. We proved $\widetilde{O}(\rho\sqrt{n}/\varepsilon)$ additive error for both notions using our framework; furthermore, the merge height error is shown to be near-tight using a lower bound following a similar idea as Theorem 2. For HC with objective functions, we used the dissimilarity objective (a score to maximize) introduced by (Cohen-Addad et al., 2019). Our framework is able to achieve (ε, δ)-DP with $\widetilde{O}(\rho n^3/\varepsilon)$ error, which is nearly tight by a matching lower bound. Both bounds for HC with the dissimilarity objective rely on the crucial observation that the score for metric space with unit weights is $S_n = \frac{n^3-n}{3}$ regardless of the tree structure. For the upper bound, we use S_n to show that the score induced by any tree on random Gaussian matrices cannot be too large. As such, an optimal HC tree on the private release induces a large enough score on the original metric. For the lower bound, we use S_n to control the "baseline cost" for a planted structure, and we argue that any HC tree that are private must violate enough pairs in the planted structure, which translates to large additive error.

Coresets. Finally, we apply our framework to strong coresets for (k, z)-clustering. A strong coreset S of a dataset $[n]$ is a weighted subset of points for *any* set of k centers C, the cost induced by the centers on the S is similar to the cost on $[n]$. By post-processing the private metric release, we show that it is possible to construct an (ε, δ)-DP strong coreset using $\widetilde{O}(k^2)$ centers to achieve $(O(1), \widetilde{O}((\frac{\rho}{\varepsilon})^z n^{1+z/2}))$ approximation for any constant z. Since the distances are given by a $n \times n$ distance matrix D in our setting, our algorithm stores $\widetilde{O}(k^2 n)$ entries from the distance matrix to recover the clustering for any given center set C. Our coreset algorithm uses the *sensitivity sampling* framework provided by (Feldman & Langberg, 2011). Since we conduct metric repairing before post-processing, all properties for the coreset algorithm remain valid with the released metric. We then prove that if the ℓ_∞ difference between two metrics are bounded, then the coreset for one metric would be a coreset for another with some additive error. Combining the above steps leads to our bound for DP coresets.

6 Experiments

We evaluate the input-perturbation with post-processing pipeline on three tasks: private metric release, minimum spanning trees, and k-median clustering. We first compare various repair methods for the metric release, and then measure how the private matrix affects the downstream task performance; for MST we compare with the Euclidean algorithm of Zou et al. (2026) on Euclidean datasets. Due to the limited space, we report the k-median clustering results in Appendix J.4.

Setup. Unless stated otherwise, we use $\varepsilon = 1$ and $\delta = 10^{-4}$. ρ is public and fixed before any mechanism is run: $\rho = 0.1$ for the graph metrics, and ρ equals 2% of the median pairwise distance for all other datasets. This is only an experimental choice, note that ρ must be specified independently of the input. The Gaussian noise is calibrated with the analytic Gaussian mechanism at sensitivity $\rho\sqrt{n-1}$. At least 3 noise seeds are used.

6.1 Private Metric Release

Datasets and methods. We use 12 datasets, two of them synthetic (Expander and Diamond), covering eight different metrics; Table 2 gives a summary of all datasets. More details of datasets can be found in Appendix J.1. We compare two metric repair methods with three heuristics:

- ℓ_∞ projection (∞-SP, ∞-LS): the pseudometric nearest to the noisy matrix in maximum entry-wise distance, as analyzed in Theorem 1. This projection is not unique: ∞-SP returns the largest

optimal pseudometric, computed with shortest paths (SP), and ∞-LS returns the optimal one closest to the noisy matrix in least squares (LS).

- ℓ_2 projection (2-Tri, 2-Cyc): the pseudometric nearest to the noisy matrix in squared error, the unique solution of a convex quadratic program with the triangle inequality constraints. We solve it with Project-and-Forget (Sonthalia & Gilbert, 2022), enforcing either the triangle inequalities directly (2-Tri) or the equivalent cycle inequalities found by shortest paths (2-Cyc).
- Inflated closure (SP-Closure): add a high-probability bound on the largest noise entry to every entry, clip at zero, and take the shortest-path closure.
- Multidimensional scaling (MDS) (Torgerson, 1952): Apply the classical MDS, keep the positive eigenvalues of the clipped noisy matrix to reconstruct Euclidean distances.
- Fréchet embedding (Fréchet): represent each point by its row of the noisy matrix and return the maximum-norm distances between rows, which always form a pseudometric.

The last three methods are heuristics that we adapt: they do not compute an optimal projection, but they are fast and can be useful in practice, so we evaluate them together with the projection solvers. We implemented three solvers for each projection; Appendix J.2 describes and compares all six, and the main table reports four of them.

Table 2: Datasets for private metric release.

Dataset	Metric	n	Dataset	Metric	n
MNIST	W_1	500	MovieLens	Jaccard	943
CIFAR-10	W_1	500	Mushroom	Hamming	1,000
ModelNet10	Hausdorff	500	Email	shortest path	986
Characters	discrete Fréchet	500	Facebook	shortest path	4,039
Sushi	Kendall	1,000	Expander	shortest path	256
Swiss-Prot	Edit	1,000	Diamond	shortest path	172

Table 3 reports means over three random seeds. Time includes noise generation and repair. We allow a 300-second solver budget. A solver that reaches this budget may return a feasible candidate without an optimality certificate. For the iterative solvers, running times in seconds are shown in parentheses; ∞-SP and the heuristics take under 0.25 seconds, except on Facebook (at most 25 seconds). † marks candidates returned at the budget without an optimality certificate in at least one seed. Running times are averaged over the seeds, so a † can appear with a mean below 300 seconds when only one seed reaches the budget, as for ∞-LS on ModelNet10 and Diamond. Results for $\varepsilon \in \{0.1, 0.5, 1.5, 2\}$ are given in Appendix J.3.

Table 3: Maximum entrywise error for private metric release, runtime in seconds in parentheses.

	ℓ_∞ projection		ℓ_2 projection		Heuristics		
Dataset	∞-SP	∞-LS	2-Tri	2-Cyc	SP-Closure	MDS	Fréchet
MNIST	0.857	0.200 (98.2)	**0.199** (0.63)	**0.199** (2.6)	1.33	0.862	1.31
CIFAR-10	1.79	**0.407** (99.6)	0.408 (2.0)	0.410 (3.1)	2.83	1.85	2.70
ModelNet10	4.47	0.881† (153)	**0.882** (0.43)	**0.882** (2.3)	7.28	4.75	6.96
Characters	1.91	**0.481** (64.0)	0.485 (9.9)	0.485 (6.3)	2.69	2.12	2.64
Sushi	228	41.4† (301)	**37.5** (6.2)	**37.5** (40.0)	324	226	318
Swiss-Prot	3395	625† (301)	**531** (7.0)	**531** (43.5)	4788	3341	4723
Mushroom	149	26.8† (301)	**22.4** (4.4)	**22.4** (38.5)	205	143	202
MovieLens	11.0	1.67† (300)	**1.46** (3.4)	**1.46** (28.3)	15.2	10.7	14.6
Email	61.7	6.52† (301)	**5.86** (7.8)	5.87 (62.0)	82.4	57.7	80.8
Facebook	**144**	8.28† (320)	8.21† (335)	8.00† (307)	174	147	188
Expander	27.8	**4.83** (10.8)	4.85 (0.09)	4.86 (0.47)	41.4	23.6	36.3
Diamond	20.3	7.69† (105)	**7.92** (0.53)	7.93 (1.1)	34.7	23.6	27.9

We have a few observations from Table 3. First, on every dataset except Facebook, the smallest error comes from one of the three iterative solvers (∞-LS, 2-Tri, or 2-Cyc), which are much more accurate than ∞-SP, at the cost of much more running time. The ℓ_2 projection does not give optimal entrywise error theoretically, but empirically it works well. Among the heuristics, MDS is always the most accurate, comparable to ∞-SP. Second, when the three iterative solvers finish, they reach nearly the same error, so the real difference is speed: 2-Tri usually finishes in a few seconds and ∞-LS is the slowest. Third, the iterative solvers do not scale well. On Facebook, the largest dataset, none

of them finishes within the time budget, and their outputs set most distances to zero, so their small errors do not reflect good repairs. Only ∞-SP and the heuristics finish there, within 25 seconds.

Choosing a repair. Our results suggest a simple rule. If utility is the priority, use one of the iterative solvers: 2-Tri usually converges the fastest of the three. If running time matters more, or the dataset is too large for the iterative solvers, MDS is a good choice.

6.2 Private Minimum Spanning Trees

We compare our MST algorithm (Theorem 3) with the Euclidean algorithm of Zou et al. (2026), following their experimental setup in Euclidean space. Their algorithm maps the points to a lower dimension with a Johnson–Lindenstrauss (JL) transform, adds Laplace noise to the projected coordinates, and computes an MST on the resulting Euclidean distances. We use the same data with $n = 500$ and varying dimension d from 1000 to 5000. We again set $\varepsilon = 1$, $\delta = 10^{-4}$, and $\rho = 1$; for their algorithm we use the projected dimension 60 from their code. Our algorithm corresponds to "No repair", i.e., adding Gaussian noise to the distance matrix and running Kruskal's algorithm. The reported results are averaged over 5 repetitions.

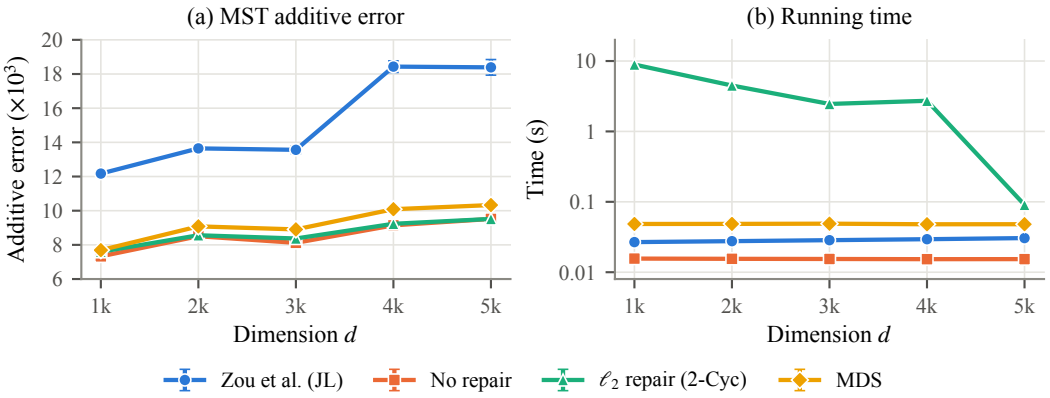


Figure 1: Private MST additive error and running time on Euclidean data.

Figure 1 shows that our algorithm gives a smaller error than the algorithm of Zou et al. (2026) in every dimension, and it is also faster. A likely reason is the JL transform distorts each distance by a multiplicative factor, so its error grows with the scale of the data, but the noise we add does not depend on d. The ℓ_2 repair also becomes faster as d grows, because the distances grow while the noise stays fixed, so the noisy matrix violates fewer triangle inequalities and has less to repair.

7 Conclusion

We investigate input perturbation, a simple private mechanism that adds independent Gaussian noise to pairwise distances, for the setting when the pairwise distance matrix is the input to a number of fundamental optimization problems. We prove nearly tight bounds for private metric release. Via different post-processing procedures, the released distances yield algorithms for minimum spanning trees, k-clustering, hierarchical clustering and coresets, with nearly matching bounds for several of them. We evaluate a subset of the algorithms empirically. We show error-efficiency trade-off for metric release of the proposed algorithms together with simple heuristics, demonstrating the usefulness of input perturbation in practice. Future directions include pursuing tight bounds and developing other mechanisms (e.g. problem-specific private projections) for the problems concerned.

AI Disclosure

Part of the theoretical results were obtained via significant interactions with frontier AI models, including ChatGPT, Claude, and Gemini. The authors have checked all the analysis, understood the core ideas of the analysis, and will take full responsibility for the correctness. The authors also used AI tools to polish writing and help with implementations.

REFERENCES

Vijay Arya, Naveen Garg, Rohit Khandekar, Adam Meyerson, Kamesh Munagala, and Vinayaka Pandit. Local search heuristics for k-median and facility location problems. *SIAM Journal on Computing*, 33(3):544–562, 2004. doi: 10.1137/S0097539702416402.

Justin Brickell, Inderjit S Dhillon, Suvrit Sra, and Joel A Tropp. The metric nearness problem. *SIAM Journal on Matrix Analysis and Applications*, 30(1):375–396, 2008.

Mark Bun, Jonathan Ullman, and Salil Vadhan. Fingerprinting codes and the price of approximate differential privacy. In *Proceedings of the forty-sixth annual ACM symposium on Theory of computing*, pp. 1–10, 2014.

Gunnar E. Carlsson and Facundo Mémoli. Characterization, stability and convergence of hierarchical clustering methods. *J. Mach. Learn. Res.*, 11:1425–1470, 2010. doi: 10.5555/1756006.1859898. URL https://dl.acm.org/doi/10.5555/1756006.1859898.

Alisa Chang, Badih Ghazi, Ravi Kumar, and Pasin Manurangsi. Locally private k-means in one round. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 1441–1451. PMLR, 2021. URL https://proceedings.mlr.press/v139/chang21a.html.

Nicos Christofides. Worst-case analysis of a new heuristic for the travelling salesman problem. Management Sciences Research Report 388, Graduate School of Industrial Administration, Carnegie Mellon University, 1976.

Kenneth L. Clarkson. Nearest neighbor queries in metric spaces. *Discrete & Computational Geometry*, 22(1):63–93, 1999. doi: 10.1007/PL00009449.

Vincent Cohen-Addad, Varun Kanade, Frederik Mallmann-Trenn, and Claire Mathieu. Hierarchical clustering: Objective functions and algorithms. *J. ACM*, 66(4):26:1–26:42, 2019. doi: 10.1145/3321386. URL https://doi.org/10.1145/3321386.

Vincent Cohen-Addad, Alessandro Epasto, Vahab Mirrokni, Shyam Narayanan, and Peilin Zhong. Near-optimal private and scalable k-clustering. In *Advances in Neural Information Processing Systems*, volume 35, pp. 10462–10475, 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/43f55776896a2e33239c2954519f605e-Abstract-Conference.html.

Sanjoy Dasgupta. A cost function for similarity-based hierarchical clustering. In Daniel Wichs and Yishay Mansour (eds.), *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2016, Cambridge, MA, USA, June 18-21, 2016*, pp. 118–127. ACM, 2016. doi: 10.1145/2897518.2897527. URL https://doi.org/10.1145/2897518.2897527.

Chengyuan Deng, Jie Gao, Jalaj Upadhyay, Chen Wang, and Samson Zhou. On the price of differential privacy for hierarchical clustering. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/f122a6778999d1d2b25cdc6927930f1a-Abstract-Conference.html.

Laxman Dhulipala, David Eisenstat, Jakub Lacki, Vahab Mirrokni, and Jessica Shi. Hierarchical agglomerative graph clustering in poly-logarithmic depth. In Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022. URL http://papers.nips.cc/paper_files/paper/2022/hash/909de96145d97514b143dfde03e6cd2b-Abstract-Conference.html.

Edsger W. Dijkstra. A note on two problems in connexion with graphs. *Numerische Mathematik*, 1: 269–271, 1959. doi: 10.1007/BF01386390.

Max Dupré la Tour, Monika Henzinger, and David Saulpic. Making old things new: A unified algorithm for differentially private clustering. In *International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 12046–12086, 2024. URL https://proceedings.mlr.press/v235/dupre-la-tour24a.html.

Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of Cryptography*, volume 3876 of *Lecture Notes in Computer Science*, pp. 265–284. Springer, 2006. doi: 10.1007/11681878_14.

Cynthia Dwork, Aaron Roth, et al. The algorithmic foundations of differential privacy. *Foundations and trends® in theoretical computer science*, 9(3–4):211–407, 2014.

Alessandro Epasto, Vahab Mirrokni, Shyam Narayanan, and Peilin Zhong. k-means clustering with distance-based privacy. *Advances in Neural Information Processing Systems*, 36:19570–19593, 2023.

Dan Feldman and Michael Langberg. A unified framework for approximating and clustering data. In Lance Fortnow and Salil P. Vadhan (eds.), *Proceedings of the 43rd ACM Symposium on Theory of Computing, STOC 2011, San Jose, CA, USA, 6-8 June 2011*, pp. 569–578. ACM, 2011. doi: 10.1145/1993636.1993712.

Dan Feldman, Amos Fiat, Haim Kaplan, and Kobbi Nissim. Private coresets. In *Proceedings of the Forty-First Annual ACM Symposium on Theory of Computing*, pp. 361–370. ACM, 2009. doi: 10.1145/1536414.1536465.

Ofer Gabber and Zvi Galil. Explicit constructions of linear-sized superconcentrators. *Journal of Computer and System Sciences*, 22(3):407–420, 1981. doi: 10.1016/0022-0000(81)90040-4.

Badih Ghazi, Ravi Kumar, and Pasin Manurangsi. Differentially private clustering: Tight approximation ratios. In *Advances in Neural Information Processing Systems*, volume 33, pp. 4040–4054, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/299dc35e747eb77177d9cea10a802da2-Abstract.html.

Mikhael Gromov. *Structures métriques pour les variétés riemanniennes*, volume 1 of *Textes Mathématiques*. CEDIC/Fernand Nathan, Paris, 1981. Rédigé par Jacques Lafontaine et Pierre Pansu.

Anupam Gupta, Katrina Ligett, Frank McSherry, Aaron Roth, and Kunal Talwar. Differentially private combinatorial optimization. In *Proceedings of the Twenty-First Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 1106–1125. SIAM, 2010. doi: 10.1137/1.9781611973075.90.

F. Maxwell Harper and Joseph A. Konstan. The MovieLens datasets: History and context. *ACM Transactions on Interactive Intelligent Systems*, 5(4):19:1–19:19, 2015. doi: 10.1145/2827872.

Richard Hladík and Jakub Tětek. Near-universally-optimal differentially private minimum spanning trees. In *Symposium on Foundations of Responsible Computing*, volume 329 of *LIPIcs*, pp. 6:1–6:19, 2025. doi: 10.4230/LIPIcs.FORC.2025.6.

Jacob Imola, Alessandro Epasto, Mohammad Mahdian, Vincent Cohen-Addad, and Vahab Mirrokni. Differentially private hierarchical clustering with provable approximation guarantees. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 14353–14375. PMLR, 2023. URL https://proceedings.mlr.press/v202/imola23a.html.

Jacob Imola, Rasmus Pagh, and Lukas Retschmeier. Lower bounds for private graph optimization problems using reconstruction attacks. *arXiv preprint arXiv:2609.10877*, 2026. URL https://arxiv.org/abs/2609.10877.

Toshihiro Kamishima. SUSHI preference data sets, 2016. URL https://www.kamishima.net/sushi/.

Leonid V. Kantorovich. On the translocation of masses. *Comptes Rendus (Doklady) de l'Académie des Sciences de l'URSS*, 37:199–201, 1942.

Krishnaram Kenthapadi, Aleksandra Korolova, Ilya Mironov, and Nina Mishra. Privacy via the Johnson–Lindenstrauss transform. *Journal of Privacy and Confidentiality*, 5(1):39–71, 2013. doi: 10.29012/jpc.v5i1.625.

Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

Joseph B. Kruskal. On the shortest spanning subtree of a graph and the traveling salesman problem. *Proceedings of the American Mathematical Society*, 7(1):48–50, 1956. doi: 10.1090/S0002-9939-1956-0078686-7.

Thomas Lavastida, Kefu Lu, Benjamin Moseley, and Yuyan Wang. Scaling average-linkage via sparse cluster embeddings. In Vineeth N. Balasubramanian and Ivor W. Tsang (eds.), *Asian Conference on Machine Learning, ACML 2021, 17-19 November 2021, Virtual Event*, volume 157 of *Proceedings of Machine Learning Research*, pp. 1429–1444. PMLR, 2021. URL https://proceedings.mlr.press/v157/lavastida21a.html.

Yann LeCun, Leon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Vladimir I. Levenshtein. Binary codes capable of correcting deletions, insertions, and reversals. *Soviet Physics Doklady*, 10(8):707–710, 1966.

Wenye Li, Fangchen Yu, and Zichen Ma. Metric nearness made practical. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 8648–8656, 2023.

Nathan Linial, Eran London, and Yuri Rabinovich. The geometry of graphs and some of its algorithmic applications. *Combinatorica*, 15(2):215–245, 1995. doi: 10.1007/BF01200757.

Julian McAuley and Jure Leskovec. Learning to discover social circles in ego networks. In *Advances in Neural Information Processing Systems*, volume 25, 2012.

S. Muthukrishnan and Aleksandar Nikolov. Optimal private halfspace counting via discrepancy. In *Proceedings of the Forty-Fourth Annual ACM Symposium on Theory of Computing*, pp. 1285–1292. ACM, 2012. doi: 10.1145/2213977.2214090.

Ilan Newman and Yuri Rabinovich. A lower bound on the distortion of embedding planar metrics into euclidean space. *Discrete & Computational Geometry*, 29(1):77–81, 2003. doi: 10.1007/s00454-002-2813-5.

Kobbi Nissim, Uri Stemmer, and Salil Vadhan. Locating a small cluster privately. In *Proceedings of the 35th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems*, pp. 413–427. ACM, 2016. doi: 10.1145/2902251.2902296.

Rasmus Pagh, Lukas Retschmeier, Hao Wu, and Hanwen Zhang. Optimal bounds for private minimum spanning trees via input perturbation. *Proceedings of the ACM on Management of Data*, 3 (2), 2025. URL https://arxiv.org/abs/2412.10130.

Adam Sealfon. Shortest paths and distances with differential privacy. In *Proceedings of the 35th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems*, pp. 29–41. ACM, 2016. doi: 10.1145/2902251.2902291. URL https://arxiv.org/abs/1511.04631.

Rishi Sonthalia and Anna C. Gilbert. Project and forget: Solving large-scale metric constrained problems. *Journal of Machine Learning Research*, 23(326):1–54, 2022. URL https://www.jmlr.org/papers/v23/20-1424.html.

Thomas Steinke and Jonathan Ullman. Between pure and approximate differential privacy. *The Journal of Privacy and Confidentiality*, 7(2), 2016.

Thomas Steinke and Jonathan Ullman. Tight lower bounds for differentially private selection. In *IEEE Symposium on Foundations of Computer Science*, 2017. URL https://arxiv.org/abs/1704.03024.

Peipei Tang, Bo Jiang, and Chengjing Wang. An efficient algorithm for the ℓ_p norm based metric nearness problem. *Mathematics of Computation*, 94(355):2495–2532, 2025. doi: 10.1090/mcom/4026. URL https://doi.org/10.1090/mcom/4026.

The UniProt Consortium. UniProt: The universal protein knowledgebase in 2025. *Nucleic Acids Research*, 53(D1):D609–D617, 2025. doi: 10.1093/nar/gkae1010.

Warren S. Torgerson. Multidimensional scaling: I. Theory and method. *Psychometrika*, 17(4): 401–419, 1952.

UCI Machine Learning Repository. Mushroom. UCI Machine Learning Repository, 1981.

Ben Williams. Character trajectories. UCI Machine Learning Repository, 2006.

Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3D ShapeNets: A deep representation for volumetric shapes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1912–1920, 2015.

Hao Yin, Austin R. Benson, Jure Leskovec, and David F. Gleich. Local higher-order graph clustering. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 555–564, 2017. doi: 10.1145/3097983.3098069.

Zongrui Zou, Alessandro Epasto, Chenglin Fan, and Rudrajit Das. Differentially private minimum spanning tree in euclidean graphs. In *International Conference on Artificial Intelligence and Statistics*, volume 300 of *Proceedings of Machine Learning Research*, pp. 379–387, 2026. URL https://proceedings.mlr.press/v300/zou26a.html.

A Technical Preliminaries

We list the technical tools we use in the proofs of this paper. The first tool we used is the standard gaussian tail bound.

Proposition A.1 (Gaussian tail bound)**.** *Let $X \sim \mathcal{N}(0, \sigma^2)$, where $\sigma > 0$. For every $t \geq 0$,*

$$\Pr\left(|X| \geq t\right) \leq 2\exp\left(-\frac{t^2}{2\sigma^2}\right).$$

Consequently, if $X_1, \ldots, X_N$ each have distribution $\mathcal{N}(0, \sigma^2)$, then for every $0 < \beta < 1$,

$$\Pr\left(\max_{r\in[N]} |X_r| \geq \sigma\sqrt{2\log\frac{2N}{\beta}}\right) \leq \beta.$$

The next inequality bounds the sum of squared Gaussians. This bound is less standard but still well-known. We give a self-contained proof for completeness.

Proposition A.2 (Concentration bound for squared Gaussians)**.** *Let $X_1, \ldots, X_m$ be independent centered Gaussian random variables, each with variance at most v^2 for some $v > 0$. Then, for every $t \geq 0$,*

$$\Pr\left(\sum_{i=1}^{m} X_i^2 > 4v^2(m+t)\right) \leq \exp\left(-t\right).$$

Proof. For a standard Gaussian random variable G and $0 \leq s < 1/2$, direct integration gives

$$\begin{aligned}\mathbb{E}[\exp(sG^2)] &= \frac{1}{\sqrt{2\pi}}\int_{-\infty}^{\infty}\exp\left(-\frac{1-2s}{2}x^2\right)dx \\ &= (1-2s)^{-1/2}. \qquad \text{(Gaussian integral)}\end{aligned}$$

Write $v_i^2 = \mathrm{Var}(X_i) \le v^2$, and note that $X_i = v_i G$. Applying this formula with $s = v_i^2/(4v^2)$ gives

$$\mathbb{E}\left[\exp\left(\frac{X_i^2}{4v^2}\right)\right] = \left(1 - \frac{v_i^2}{2v^2}\right)^{-1/2} \le \sqrt{2}. \qquad (v_i^2 \le v^2)$$

Consequently,

$$\Pr\left(\sum_{i=1}^m X_i^2 > 4v^2(m+t)\right) \le e^{-(m+t)}\mathbb{E}\left[\exp\left(\frac{1}{4v^2}\sum_{i=1}^m X_i^2\right)\right] \qquad \text{(exponential Markov inequality)}$$

$$= e^{-(m+t)}\prod_{i=1}^m \mathbb{E}\left[\exp\left(\frac{X_i^2}{4v^2}\right)\right] \qquad \text{(independence)}$$

$$\le e^{-(m+t)}2^{m/2} \le e^{-t}, \qquad (2^{m/2} \le e^m)$$

as desired. □

B Related Work

Private computation on metric data. Private optimization in general metric spaces has also been studied when the metric is public. Gupta et al. (Gupta et al., 2010) study k-median and facility location in a public metric space, where neighboring inputs differ by the addition or removal of one point. Kenthapadi et al. (Kenthapadi et al., 2013) combine random projection and Gaussian noise to estimate distances between binary vectors while protecting individual bits. Feldman et al. (Feldman et al., 2009) construct private coresets for Euclidean k-median and k-means. Epasto et al. (Epasto et al., 2023) introduce ρ-distance privacy model for Euclidean k-clustering, reducing the dependence of additive error on the domain diameter. Our privacy model is inspired directly from their work.

Private MST and hierarchical clustering. There have been many works on private MST and hierarchical clustering. Sealfon (Sealfon, 2016) introduces edge-weight privacy with public graph topology and gives algorithms and lower bounds for MSTs under ℓ_1 neighboring. Later works study MSTs under both ℓ_1 and ℓ_∞ neighboring relations (Hladík & Tětek, 2025; Pagh et al., 2025). Pagh et al. (Pagh et al., 2025) obtain nearly optimal guarantees through input perturbation. Zou et al. (Zou et al., 2026) uses the similar privacy model as (Epasto et al., 2023) and this paper, but the study focuses on Euclidean space. For hierarchical clustering, Imola et al. (Imola et al., 2023) give private approximation algorithms and lower bounds for Dasgupta's similarity objective under edge-level DP. Deng et al. (Deng et al., 2025) extends this study to weight-DP, and obtain a polynomial-time $O(\log^{3/2} n/\varepsilon)$ approximation for graphs with edge weights at least one, and an $\Omega(n^2/\varepsilon)$ additive lower bound without this assumption. Imola et al. (Imola et al., 2026) extend the lower bounds to approximate weight privacy and more general graph topologies. These HC results concern similarity graphs, whereas we study linkage errors and the dissimilarity objective on pairwise distance matrices.

C Proof of the Metric Release Lower Bound

Proof of Theorem 2. Let $k = n - 1$ and $N = \lceil\sqrt{k}/\varepsilon\rceil$, and let $Z \in \{0,1\}^{N\times k}$ be a database of N users, user u holding the vector $Z_u \in \{0,1\}^k$. We place n points P_Z in $\mathbb{R}^k$ under the ℓ_∞ metric. The mobile point sits at the scaled sum of the users,

$$x_1 = \rho\sum_{u=1}^N Z_u,$$

so its m-th coordinate is $(x_1)_m = \rho\, q_m(Z) \in [0, N\rho]$, and the remaining points are anchors placed far out along the coordinate axes, $x_j = L \cdot e_{j-1}$ for $j \in \{2,\dots,n\}$ with $L = 2N\rho$.

Each anchor reads off one marginal. For $j \geq 2$,

$$d(x_1, x_j) = \|x_1 - Le_{j-1}\|_\infty = \max\Big(L - (x_1)_{j-1},\ \max_{m \neq j-1} (x_1)_m\Big).$$

Since every coordinate of x_1 is at most $N\rho$ and $L - (x_1)_{j-1} \geq N\rho$, the maximum is attained by the first term $L - (x_1)_{j-1}$, i.e., $d(x_1, x_j) = L - \rho\, q_{j-1}(Z)$.

Changing one user is a ρ-move. If Z and Z' differ only in the row of user u, the anchors are untouched and

$$\|x_1 - x_1'\|_\infty = \rho\, \|Z_u - Z_u'\|_\infty \leq \rho,$$

since the entries in Z, Z' are binary. So $P_Z \sim_\rho P_{Z'}$, and $\mathcal{A}$ running on P_Z is (ε, δ)-DP with respect to one-user changes of Z. By our choice of N, $\delta = o(\varepsilon/\sqrt{n}) = o(1/N)$, which is what Proposition 3.1 requires.

Now we answer the marginals using $\mathcal{A}$. Given Z, build the point set P_Z and the ℓ_∞-distance matrix D, run $\mathcal{A}$ on D to obtain $\hat{D}$, and report $\hat{q}_m = (L - \hat{D}_{1,m+1})/\rho$ for each $m \in [k]$; this is a post-processing step of $\mathcal{A}$ and hence is (ε, δ)-DP. Because $d(x_1, x_{m+1}) = L - \rho\, q_m(Z)$, the error on marginal m is

$$|\hat{q}_m - q_m(Z)| = \frac{1}{\rho}\big|\hat{D}_{1,m+1} - d(x_1, x_{m+1})\big|.$$

The first row of $\hat{D}$ is part of the matrix, so if $\mathcal{A}$ has expected maximum error E then this mechanism answers all k marginals with expected maximum error at most E/ρ. By Proposition 3.1 and our choice $N = \lceil \sqrt{k}/\varepsilon \rceil$,

$$\frac{E}{\rho} \;\geq\; \Omega\left(\min\left(N, \frac{\sqrt{k}}{\varepsilon}\right)\right) \;=\; \Omega\left(\frac{\sqrt{n-1}}{\varepsilon}\right),$$

which is the claimed bound. □

D A (YET ANOTHER) PROOF FOR DP MST LOWER BOUNDS

Theorem 4. *There is an absolute constant $c > 0$ such that, for all sufficiently large n, every $\varepsilon \in (0, 1]$, and every $\delta \leq \varepsilon/n$, every mechanism M that is (ε, δ)-DP under the ρ-distance model and outputs a spanning tree has a metric input D on n points satisfying*

$$\mathrm{err}(M, D) \geq c\, \frac{\rho n^{3/2}}{\varepsilon}.$$

Together with the input-perturbation upper bound $\widetilde{O}(\rho n^{3/2}/\varepsilon)$, this determines the optimal additive error up to logarithmic factors.

High-level overview. For simplicity, we let $\rho = 1$ for the high-level discussion. The hard instance contains two special points, p and q, together with $m = n - 2$ "satellite points" $s_1, \ldots, s_m$. For each satellite s_j, the tree must decide whether to connect it to p or to q. The correct choice depends on a number c_j, which is the sum of the j-th column held by the N users in an auxiliary binary database. If $c_j < N/2$, then p is closer to s_j; if $c_j > N/2$, then q is closer to s_j.

Thus, producing a nearly optimal spanning tree requires making many correct decisions of the form "is c_j smaller or larger than $N/2$?" Choosing the wrong hub for s_j increases the weight of the tree by $O(|c_j - N/2|/\varepsilon)$. Therefore, coordinates that are far from the threshold are more costly to misclassify, which will become the main source of error in our lower bound.

We choose the database randomly as follows. For every column j, first choose a hidden probability P_j uniformly from $[0, 1]$. Then each user independently places a 1 in coordinate j with probability P_j. The count c_j is therefore an estimate of $N \cdot P_j$. A tree that has small additional weight beyond the true MST must, on average, connect satellite points according to whether P_j is below or above $1/2$, especially for coordinates where P_j is far from $1/2$.

The fingerprinting lower bound from Steinke and Ullman (Steinke & Ullman, 2017) says that a differentially private algorithm cannot recover this much information about all the hidden probabilities

$P_1, \ldots, P_m$ when the database has only $N = O(\sqrt{m})$ users. We convert the output tree into one bit for each satellite point, recording which hub the tree chose. If the tree had very small weight error, these bits would be too strongly correlated with the hidden probabilities, contradicting the fingerprinting theorem.

The above steps lead to an $\Omega(m^{3/2}R)$ lower bound when changing one user can move the mobile point by distance at most R. Finally, we choose R to be on the order of ρ/ε, which allows us to connect $O(1/\varepsilon)$ sub-instances that are neighboring with parameter ρ. Group privacy then allows us to apply the constant-privacy fingerprinting lower bound, and the resulting MST lower bound becomes $\Omega(\rho n^{3/2}/\varepsilon)$.

Parameters. Let $m = n - 2$, let $\gamma = 1/16$, and set

$$N = \left\lfloor \frac{\sqrt{m}}{64} \right\rfloor, \qquad r = \max\left\{1, \left\lfloor \frac{1}{\varepsilon} \right\rfloor\right\}, \qquad R = r\rho.$$

For sufficiently large n, we have $N \geq 2$, $r\varepsilon \leq 1$, and $R \geq \rho/(2\varepsilon)$.

The hard instance. The fixed points are a fixed hub q, a moving hub p_x, and satellite points $s_1, \ldots, s_m$ ($m = n - 2$). The moving hub p_x is specified by a vector $x = (x_1, \ldots, x_m) \in [0, N]^m$ such that

$$D(p_x, p_y) = R\|x - y\|_\infty, \qquad D(p_x, q) = \frac{RN}{2}, \qquad D(p_x, s_j) = 10RN + Rx_j,$$

and define

$$D(q, s_j) = 10RN + \frac{RN}{2}, \qquad D(s_i, s_j) = 15RN \quad \text{for } i \neq j.$$

We first verify that the constructed hard instance induces a metric.

Lemma D.1. *The distances above define a valid metric for every choice of $x = (x_1, \ldots, x_m) \in [0, N]^m$. Furthermore, the metric property holds even if all p_x for $x \in [0, N]^m$ are included simultaneously.*

Proof. We first consider the case when a single p_x can be involved. Since $x_j \in [0, N]$, every distance between p_x and some s_j lies in $[10RN, 11RN]$, while every distance between q and some s_j is $10.5RN$. Hence

$$D(p_x, s_j) \leq 11RN = D(p_x, q) + D(q, s_j),$$

and the other two inequalities for the triple $\{p_x, q, s_j\}$ are immediate. If we consider two satellite points s_i and s_j, we have

$$D(s_i, s_j) = 15RN \leq 20RN \leq D(s_i, p_x) + D(p_x, s_j)$$

and

$$D(s_i, s_j) = 15RN \leq 21RN = D(s_i, q) + D(q, s_j).$$

All remaining triangle inequalities follow immediately from the same bounds.

For the "furthermore" part, the points p_x form a scaled ℓ_∞ metric. Therefore, we have

$$|D(p_x, s_j) - D(p_y, s_j)| = R|x_j - y_j| \leq R\|x - y\|_\infty = D(p_x, p_y),$$

so every triangle containing two p_x, p_y points and one satellite s_j satisfies the triangle inequality. The same is true for a triangle containing two p_x, p_y points and q, since $D(p_x, p_y) \leq RN = D(p_x, q) + D(q, p_y)$. □

Encoding a database. Let $Y \in \{0, 1\}^{N \times m}$ be a database with N rows, and define

$$c_j = \sum_{u=1}^{N} Y_{uj}, \qquad c = (c_1, \ldots, c_m),$$

where the moving hub p_c is parametrized by $c = (c_1, \ldots, c_m)$. The corresponding metric instance is

$$D_Y = \{p_c, q, s_1, \ldots, s_m\}.$$

In what follows, we establish the behavior of the optimal MST for a metric induced by a database Y.

Lemma D.2 (The MST is a double-star)**.** *For every database Y, an MST of D_Y consists of the edge (p_c, q) and, for every $j \in [m]$, the edge in $\{(p_c, s_j), (q, s_j)\}$ with the smaller weight. The construction leads to MST cost*

$$\mathrm{MST}(D_Y) = \frac{RN}{2} + \sum_{j=1}^{m} \min\{D(p_c, s_j), D(q, s_j)\}.$$

In other words, satellite point s_j attaches to p_c when $c_j < N/2$ and to q when $c_j > N/2$.

Proof. The edge (p_c, q) has weight $RN/2$, which has lower weight than every other edge. As such, if we run Kruskal's algorithm, the edge (p_c, q) will be added. The distance between (s_j, s_k) for any j, k is $15RN$, and the distance between (p_c, s_j) or (q, s_j) is at most $11RN$. Thus, before Kruskal's algorithm considers any (s_j, s_k) edge, each s_j has already been connected to the component containing both hubs by its lighter hub edge. The total weight then follows from the above construction. □

Note that for any s_j, the difference between the connection to p_c and q is specified by

$$D(p_c, s_j) - D(q, s_j) = R\left(c_j - \frac{N}{2}\right),$$

which is a quantity we will use repeatedly in the rest of the proof.

We now handle the issue that the two datasets induced by different Y and Y' can have more than ρ difference. This is resolved by a standard application of group privacy as follows.

Lemma D.3 (Group privacy on one row)**.** *If Y and Y' differ in one row, then D_Y and $D_{Y'}$ can be connected by at most r successive ρ-neighboring metric instances.*

Proof. Let c and c' be the corresponding count vectors. Since changing one row changes every count by at most one, $\|c - c'\|_\infty \le 1$, and therefore

$$D(p_c, p_{c'}) = R\|c - c'\|_\infty \le r\rho.$$

For $\ell = 0, \ldots, r$, define $x^{(\ell)} = c + \frac{\ell}{r}(c' - c)$. Consecutive locations satisfy

$$D(p_{x^{(\ell-1)}}, p_{x^{(\ell)}}) = R\left\|\frac{c' - c}{r}\right\|_\infty \le \rho.$$

Replacing the moving hub p successively by $\{p_{x^{(\ell)}}\}_{\ell=0}^{r}$ gives the required chain. □

Converting an MST to sign predictions. In what next, we convert an MST algorithm to a sign prediction algorithm that fits into the fingerprinting code framework as in (Steinke & Ullman, 2017). Let T be any spanning tree and root it at p_c. Define $a_j(T) \in \{-1, +1\}$ as

$$a_j(T) = \begin{cases} -1, \text{ if the parent of } s_j \text{ is } p_c; \\ +1, \text{ otherwise} \end{cases}$$

Note that the 1 case includes when the parent s_j is some other s_k. However, as we will see in Lemma D.4, this does not affect the correctness of the proof. We further define the optimal sign by

$$a_j^\star = \mathrm{sign}\left(c_j - \frac{N}{2}\right).$$

When $c_j = N/2$, either choice is optimal and the value assigned to $a_j^\star$ is irrelevant. The next lemma lower bounds the sign cost with MST additive error.

Lemma D.4 (Excess cost is weighted sign error)**.** *For every spanning tree T,*

$$w_{D_Y}(T) - \mathrm{MST}(D_Y) \ge R \sum_{j : a_j(T) \neq a_j^\star} \left|c_j - \frac{N}{2}\right|.$$

Proof. Starting from p_c as the root of T, every vertex except p_c has one parent edge, and therefore

$$w_{D_Y}(T) = D(q, \mathrm{par}(q)) + \sum_{j=1}^{m} D(s_j, \mathrm{par}(s_j)).$$

Every edge incident to q has weight at least $D(p_c, q) = RN/2$, and every edge incident to s_j has weight at least $\min\{D(p_c, s_j), D(q, s_j)\}$. We discount the edge (p_c, q) for simplicity. Consequently,

$$w_{D_Y}(T) - \mathrm{MST}(D_Y) \geq \sum_{j=1}^{m} \left(D(s_j, \mathrm{par}(s_j)) - \min\{D(p_c, s_j), D(q, s_j)\}\right).$$

If $a_j(T) \neq a_j^\star$ and the parent of s_j is the other hub, then s_j is attached to the farther hub and the extra cost is exactly

$$|D(p_c, s_j) - D(q, s_j)| = R\left|c_j - \frac{N}{2}\right|.$$

If the parent of s_j is another s_k (satellite point), then its edge has weight $15RN$, so its extra cost is at least

$$15RN - 11RN = 4RN \geq R\left|c_j - \frac{N}{2}\right|.$$

Summing these contributions proves the lemma. □

In Lemma D.4, we explicitly handled the case when $a_j(T) = 1$ and s_j is connected to some other s_k, which is the reason why this "false positive" is not an issue.

Utility formulation. We now formulate the utility of the sign prediction. Let $\overline{Y}_j = c_j/N$, and for $a \in \{-1, +1\}^m$, we define

$$U_Y(a) = \sum_{j=1}^{m} a_j \left(\overline{Y}_j - \frac{1}{2}\right).$$

The optimal sign vector $a^\star$ satisfies

$$U_Y(a^\star) = \sum_{j=1}^{m} \left|\overline{Y}_j - \frac{1}{2}\right|$$

since the sign is always consistent with whether $\overline{Y}_j$ is greater or smaller than $1/2$. For every sign vector a, the induced error are on the coordinates such that a_j does not agree with a_j^*, i.e.,

$$U_Y(a^\star) - U_Y(a) = 2 \sum_{j: a_j \neq a_j^\star} \left|\overline{Y}_j - \frac{1}{2}\right|.$$

Lemma D.4 therefore implies

$$w_{D_Y}(T) - \mathrm{MST}(D_Y) \geq \frac{RN}{2} \cdot \left(U_Y(a^\star) - U_Y(a(T))\right). \tag{1}$$

Equation (1) provides us with a way to bound the MST error as the normalized prediction error for the labels, which is suitable for the following fingerprinting lower bound framework.

Proposition D.5 ((Steinke & Ullman, 2017)). *Let $P_1, \ldots, P_m$ be independent uniform random variables on $[0, 1]$. Conditioned on $P = (P_1, \ldots, P_m)$, let the entries of $Y \in \{0, 1\}^{N \times m}$ be independent with $\Pr[Y_{uj} = 1 \mid P] = P_j$. Suppose that $A : \{0, 1\}^{N \times m} \to [-1, 1]^m$ is $(1, \gamma/N)$-differentially private and satisfies*

$$\mathbb{E}\|A(Y)\|_2^2 = m$$

and

$$\mathbb{E}\left[\sum_{j=1}^{m} A(Y)_j \left(P_j - \frac{1}{2}\right)\right] \geq \gamma m.$$

Then $N \geq \gamma\sqrt{m}$.

We are now ready to prove Theorem 4 using the MST-prediction reduction.

Proof of Theorem 4. Draw $P_1, \dots, P_m$ independently and uniformly from $[0, 1]$. Conditioned on P, draw the entries of Y independently according to $Y_{uj} \sim \text{Bernoulli}(P_j)$. Let ALG be an (ε, δ)-DP algorithm for the MST induced by D_Y, and let

$$A(Y) = a(\mathsf{ALG}(D_Y)) \in \{-1, +1\}^m$$

be the sign vector extracted from the output tree.

We first check privacy. By Lemma D.3, changing one row of Y corresponds to at most r successive ρ-neighboring changes of the metric. Group privacy therefore implies that A is $(r\varepsilon, \delta_r)$-differentially private, where

$$\delta_r = \delta \sum_{\ell=0}^{r-1} e^{\ell\varepsilon}.$$

Since $r\varepsilon \le 1$ and $r \le 1/\varepsilon$, we have

$$\delta_r \le er\delta \le \frac{e\delta}{\varepsilon} \le \frac{e}{n}.$$

For sufficiently large n, $e/n \le \gamma/N$. Thus A is also $(1, \gamma/N)$-differentially private.

We next estimate the utility of the optimal sign vector. Because P_j is uniform on $[0, 1]$ and $c_j \mid P_j \sim \text{Binomial}(N, P_j)$, the beta-binomial identity shows that c_j is uniform on $\{0, 1, \dots, N\}$. Therefore,

$$\mathbb{E}\left[\left|\frac{c_j}{N} - \frac{1}{2}\right|\right] \ge \frac{1}{4},$$

and hence

$$\mathbb{E}[U_Y(a^\star)] \ge \frac{m}{4}. \tag{2}$$

Suppose, toward a contradiction, that the expected excess cost is small than $\frac{RNm}{32}$. Combining (1) with (2) gives

$$\mathbb{E}\left[U_Y(A(Y))\right] \ge \frac{m}{4} - \frac{m}{16} = \frac{3m}{16}. \tag{3}$$

It remains to translate this empirical utility into correlation with the hidden biases P_j. Conditioned on Y, the posterior distribution of P_j is $\text{Beta}(1 + c_j, 1 + N - c_j)$, whose mean is $(1 + c_j)/(N + 2)$. Consequently,

$$\mathbb{E}\left[P_j - \frac{1}{2} \mid Y\right] = \frac{c_j - N/2}{N + 2} = \frac{N}{N + 2}\left(\overline{Y}_j - \frac{1}{2}\right).$$

Using (3), we obtain

$$\mathbb{E}\left[\sum_{j=1}^{m} A(Y)_j \left(P_j - \frac{1}{2}\right)\right] = \frac{N}{N+2}\mathbb{E}[U_Y(A(Y))] > \frac{1}{2} \cdot \frac{3m}{16} > \gamma m.$$

Moreover, $A(Y) \in \{-1, +1\}^m$, so $\mathbb{E}\|A(Y)\|_2^2 = m$. Proposition D.5 now implies

$$N \ge \gamma\sqrt{m} = \frac{\sqrt{m}}{16},$$

contradicting the choice $N = \lfloor\sqrt{m}/64\rfloor$. Therefore,

$$\mathbb{E}\left[w_{D_Y}(M(D_Y)) - \text{MST}(D_Y)\right] \ge \frac{RNm}{32}.$$

For sufficiently large m, $N \ge \sqrt{m}/128$ and $R \ge \rho/(2\varepsilon)$, and hence

$$\mathbb{E}\left[w_{D_Y}(M(D_Y)) - \text{MST}(D_Y)\right] \ge \frac{1}{8192}\frac{\rho m^{3/2}}{\varepsilon} = \Omega\left(\frac{\rho n^{3/2}}{\varepsilon}\right).$$

The expectation is over both Y and the randomness of M. By averaging, there exists a fixed database Y, and hence a fixed n-point metric D_Y, on which the expected excess cost satisfies the same asymptotic lower bound. □

E DP METRIC k-CLUSTERING

We now apply private metric release to k-clustering. We focus on the *facility* version, in which the k centers must be selected from the given set of points. The algorithm will release the *partition*, defined as the cluster membership of each point. For $1 \le k \le n$, a fixed finite exponent $z \ge 1$, and a distance metric D of n points, a *partition* $\mathcal{P} = \{P_1, P_2, \cdots, P_k\}$ of $[n]$ divides $[n]$ into k groups, where each element in $[n]$ is in exactly one group. A *medoid* $c(P)$ of a group $P \in \mathcal{P}$ is a data point $c \in P$ that has the smallest average distance (or in general, the z-th powered distance) to all other points in the same group, $c = \operatorname{argmin}_{c \in P} \sum_{i \in P} D^z(i, c)$. If we fix a partition as the clustering, the best center/facility for each group is the medoid. Now we define the cost of partition P with metric D as

$$F_z(\mathcal{P}, D) = \sum_{P \in \mathcal{P}} \sum_{i \in P} D^z(i, c(P)),$$

and the optimal k-clustering as the best partition into k groups with the minimum cost

$$\mathrm{OPT}_{k,z}(D) = \min_{|\mathcal{P}|=k} F_z(\mathcal{P}, D).$$

Observe that in an optimal solution, the distance from a point i in a group P to the medoid of P is no further than the distance from i to the medoid of other groups – otherwise, i should switch to a different group. We call this problem (k, z)-*clustering* for a constant z. Specifically, the ases $z = 1$ and $z = 2$ are metric k-median and metric k-means, respectively.

We say that a solution is an (α, β)-*approximation* to the optimal solution if the cost is at most $\alpha \cdot \mathrm{OPT} + \beta$. In what follows, we will first show that if we disregard *time efficiency*, it is possible to apply our framework and obtain $(O(1), \widetilde{O}_{\delta,\varepsilon}(\rho n\sqrt{k}))$-approximation for k-median and $(O(1), \widetilde{O}_{\delta,\varepsilon}(\rho^2 nk))$-approximation for $z \ge 2$. We also develop algorithms with polynomial running time but with worse utility (higher additive errors). The approximation factor and running time are presented in Table 4. Due to space constraints, this is deferred to Appendix G.

We give two complementary uses of the metric structure. Averaging distances within a proposed cluster gives a constant-factor surrogate whose Gaussian error can be controlled uniformly over all partitions. This permits accurate but exponential-time post-processing of one release. For polynomial time, we privately form preliminary groups, release their average distances, and cluster the resulting weighted metric. Both the geometric comparison and the privacy analysis allow unequal group sizes, including singletons.

Problem and method	Privacy / time	Factor	Additive upper bound	Lower bound
Median, exponential mechanism	pure / exponential	2	$O(\rho n \log k/\varepsilon)$	$\Omega(\rho n \log k/\varepsilon)$
Median, Gaussian surrogate	approximate / exponential	2	$\widetilde{O}(\rho n\sqrt{k}/\varepsilon)$	same median lower bound
Median, recursive compression	approximate / polynomial	$C_{1,\eta}$	$\widetilde{O}_\eta(\rho n^{1+\eta} k^{1/2-\eta}/\varepsilon)$	same median lower bound
Means, Gaussian surrogate	approximate / exponential	16	$\widetilde{O}(\rho^2 nk/\varepsilon^2)$	$\Omega(\rho^2 n \log^2 k/\varepsilon^2)$
Means, recursive compression	approximate / polynomial	$C_{2,\eta}$	$\widetilde{O}_\eta(\rho^2 n^{1+\eta} k^{1-\eta}/\varepsilon^2)$	same means lower bound

Table 4: Partition-output guarantees. Upper bounds hold with probability $1 - \beta$; the median exponential-mechanism entry is displayed for constant β. Tildes hide logarithms in $n, 1/\delta, 1/\beta$. The lower bounds concern expected cost, allow any finite approximation factor, and hold for $2 \le k \le n/8$ and $\delta \le \varepsilon/(100\sqrt{k})$, including pure privacy. The recursive bounds require fixed $0 < \eta < 1/2$ for median and fixed $0 < \eta < 1$ for means. Their approximation factors depend on η. The Gaussian-surrogate and exponential-mechanism bounds also hold in expectation, with the same parameter dependence.

E.1 A NORMALIZED CLUSTERING SURROGATE

As it will be evident later, directly working with the (k, z)-clustering objective will cause some barrier in terms of the additive error. As such, we use the following surrogate objective function for (k, z)-clustering. For a real symmetric matrix A with zero on the diagonal and a partition $\mathcal{P} = \{P_1, P_2, \cdots, P_k\}$ of $[n]$, let $i \in C$ be a point within cluster C, and define $r_i(\mathcal{P}, A)$ to be the average

distance of i with other points in the same group as i in $\mathcal{P}$, and $Q_z(\mathcal{P}, A)$ to be sum of the z-th power of $r_i(\mathcal{P}, A)$ for all i, i.e.,

$$r_i(\mathcal{P}, A) = \frac{1}{|C|} \sum_{j \in C} A_{ij} \quad i \in C \in \mathcal{P}, \qquad Q_z(\mathcal{P}, A) = \sum_{i=1}^{n} |r_i(\mathcal{P}, A)|^z.$$

When the context is clear, we write r_i and Q_z in place pf $r_i(\mathcal{P}, A)$ and $Q_z(\mathcal{P}, A)$ as short-hand notations. Compared to the vanilla (k, z)-clustering objective, the $Q_z(\mathcal{P}, A)$ objective first calculate the average distance in the cluster before raising to the z-th power. The next lemma shows that $Q_z(\mathcal{P}, A)$ is a good approximation of the cost of the partition $\mathcal{P}$.

Lemma E.1. *For every metric D and partition $\mathcal{P}$,*

$$\frac{1}{2} F_z(\mathcal{P}, D)^{1/z} \le Q_z(\mathcal{P}, D)^{1/z} \le 2 F_z(\mathcal{P}, D)^{1/z}. \tag{4}$$

For $z = 1$, we can have $F_1 \le Q_1 \le 2F_1$. For $z = 2$, we have $\frac{F_2}{2} \le Q_2 \le 4F_2$.

Proof. We first work with one cluster $C \in \mathcal{P}$ of size m, and write $r_i = m^{-1} \sum_{j \in C} D_{ij}$. For any $i, j \in C$, and a third point $k \in C$, we have by triangle inequality $D_{ij} \le D_{ik} + D_{kj}$. If we average the above inequality for all $k \in C$, we have $D_{ij} \le r_i + r_j$. Take a medoid $c \in C$ minimizing r_c. Then $m r_c^z \le \sum_i r_i^z$. Use $D_{ic} \le r_i + r_c$ and apply Minkowski's inequality,

$$F_z(\mathcal{P}, D)^{1/z} = \left(\sum_i D_{ic}^z \right)^{1/z} \le \left(\sum_i r_i^z \right)^{1/z} + m^{1/z} r_c \le 2 \left(\sum_i r_i^z \right)^{1/z} = 2 Q_z(\mathcal{P}, D)^{1/z}.$$

This shows the first part of Equation (4).

For the second part, choose a medoid c and set $s_i = D_{ic}$. By the triangle inequality $D_{ij} \le D_{ic} + D_{jc} = s_i + s_j$. Averaging for all $j \in C$, $r_i \le s_i + m^{-1} \sum_j s_j$. Applying Jensen's and Minkowski's inequalities gives

$$Q_z(\mathcal{P}, D)^{1/z} = \left(\sum_i r_i^z \right)^{1/z} \le 2 \left(\sum_i s_i^z \right)^{1/z} = 2 F_z(\mathcal{P}, D)^{1/z}.$$

For $z = 1$, averaging the cost over all possible medoids gives $F_1(C, D) \le m^{-1} \sum_{i,j \in C} D_{ij} = \sum_i r_i$. For $z = 2$, multiply $D_{ij} \le r_i + r_j$ by D_{ij} and sum over all i, j to obtain

$$\frac{1}{m} \sum_{i,j \in C} D_{ij}^2 \le \frac{1}{m} \sum_{i,j \in C} D_{ij}(r_i + r_j) = 2 \sum_{i \in C} r_i^2.$$

The left side is the average squared cost of a uniformly chosen medoid, and hence upper-bounds the optimal medoid cost. □

E.2 Upper Bound for Post-processing the DP Matrix

We now use the normalized surrogate Q_z to obtain $O(1)$-approximation for (k, z)-clustering with additive error for constant z. The algorithm minimizes the surrogate objective on the released DP distance matrix and outputs a partition, and the guarantee is as follows.

Theorem 5. *Fix $z \ge 1$, $n \ge 2$, $1 \le k \le n$, $\rho > 0$, $0 < \varepsilon \le 1$, and $0 < \delta < 1/2$. Algorithm 2 is (ε, δ)-private under ρ-adjacency. For every $0 < \beta < 1$, with probability at least $1 - \beta$, its output $\widehat{\mathcal{P}}$ satisfies*

$$F_z(\widehat{\mathcal{P}}, D) \le O_z(1) \operatorname{OPT}_{k,z}(D) + O_z \left(\frac{\rho^z}{\varepsilon^z} \left(\log \frac{2}{\delta} \log \frac{2n}{\beta} \right)^{z/2} \right) \cdot H(n, k, z),$$

where the function $H(n, k, z)$ is prescribed as

$$H(n, k, z) = \begin{cases} n k^{z/2}, & 1 \le z \le 2, \\ n^{z/2} k, & z \ge 2. \end{cases}$$

Furthermore, the same guarantee holds in expectation after replacing $\log(2n/\beta)$ by $\log(2n)$.

In particular, the additive errors for k-median and k-means are $\widetilde{O}(\rho n\sqrt{k}/\varepsilon)$ and $\widetilde{O}(\rho^2 nk/\varepsilon^2)$, respectively, if we suppress log factors in n, δ, and β.

Algorithm 2. An algorithm for DP k-clustering

Input: A metric distance matrix D on the public indices $[n]$, a number of clusters k, an exponent $z \geq 1$, and privacy parameters $\varepsilon, \delta, \rho$.

Output: A partition $\widehat{\mathcal{P}}$ of $[n]$ into exactly k nonempty clusters.

1. Compute $\texttt{DP-Release}(D, \texttt{REP} = \texttt{False})$, i.e., adding an $n \times n$ symmetric Gaussian noise matrix Z with $\sigma = O\left(\frac{\rho\sqrt{n\log(1/\delta)(\log(n/\beta))}}{\varepsilon}\right)$.
2. Enumerate all partitions $\mathcal{P}$ into exactly k nonempty clusters. For each partition, compute
$$Q_z(\mathcal{P}, \widetilde{D}) = \sum_{C\in\mathcal{P}} \sum_{i\in C} \left| \frac{1}{|C|} \sum_{j\in C} \widetilde{D}(i,j) \right|^z .$$
3. Return a partition $\widehat{\mathcal{P}}$ minimizing $Q_z(\mathcal{P}, \widetilde{D})$, using a fixed rule to break ties.

Here Z denotes only the added noise, while $\widetilde{D}$ denotes the released matrix. The algorithm uses the signed entries of $\widetilde{D}$ without additional non-negative clipping or metric projection. The absolute value in Q_z makes the objective well-defined for every real $z \geq 1$.

E.2.1 Analysis of the algorithm

The analysis has three parts. We first bound the average-noise vector inside every possible cluster. We then combine these bounds to control the surrogate for every partition. Finally, the comparison between Q_z and F_z converts surrogate accuracy into clustering accuracy.

The effect of noise on a single cluster. For every nonempty $S \subseteq [n]$, define the vector b_S indexed by $i \in S$:
$$b_S(i) = \frac{1}{|S|} \sum_{j\in S} Z(i,j).$$
In other words, $b_S(i)$ is the noise (error) induced by i's average distance to S, and the vector b_S encodes the average distances.

The following lemma controls the Euclidean norm of this vector simultaneously for all possible clusters. Below, maxima over S range over nonempty subsets of $[n]$.

Lemma E.2. *For every $0 < \beta < 1$, with probability at least $1 - \beta$, simultaneously for all nonempty $S \subseteq [n]$, we have*
$$\|b_S\|_2 \leq O\left(\sigma\sqrt{\log\frac{n}{\beta}}\right).$$
Moreover, for every fixed $z \geq 1$,
$$\mathbb{E}\left[\max_S \|b_S\|_2^z\right] \leq O_z\left(\sigma^z \left(\log(n)\right)^{z/2}\right).$$

Proof. Fix a subset S with $m = |S| \geq 2$. Recall that
$$b_S(i) = \frac{1}{m}\sum_{j\in S} Z(i,j) \qquad (i \in S).$$

The coordinates of b_S are not independent because $Z(i,j)$ appears in both $b_S(i)$ and $b_S(j)$. We avoid this dependence by splitting each coordinate into

$$b_S^+(i) = \frac{1}{m}\sum_{\substack{j\in S\\ j>i}} Z(i,j), \qquad b_S^-(i) = \frac{1}{m}\sum_{\substack{j\in S\\ j<i}} Z(i,j).$$

Since the diagonal is zero, $b_S = b_S^+ + b_S^-$. Note that by this trick, for every upper-triangular noise entry $Z(i,j)$, where $i<j$, appears in exactly one coordinate of b_S^+, which is the coordinate indexed by i. Thus distinct coordinates of b_S^+ use disjoint sets of independent noise variables and are independent. Similarly, each such entry appears in exactly one coordinate of b_S^-, namely the coordinate indexed by j, so the coordinates of b_S^- are also independent. The vectors b_S^+ and b_S^- need not be independent of each other.

Every coordinate in either vector is a centered Gaussian with $\frac{m\sigma^2}{m^2} = \frac{\sigma^2}{m}$ by m independent Gaussian random variables. Applying Proposition A.2 with $v^2 = \sigma^2/m$ therefore gives, for every $t \geq 0$,

$$\Pr\left(\|b_S^+\|_2^2 > 4\sigma^2\left(1+\frac{t}{m}\right)\right) \leq \exp\left(-t\right),$$
$$\Pr\left(\|b_S^-\|_2^2 > 4\sigma^2\left(1+\frac{t}{m}\right)\right) \leq \exp\left(-t\right).$$

Except on an event of probability at most $2e^{-t}$, both bounds hold, and hence

$$\begin{aligned}\|b_S\|_2 &\leq \|b_S^+\|_2 + \|b_S^-\|_2 && \text{(triangle inequality)}\\ &\leq 4\sigma\sqrt{1+\frac{t}{m}}. && \text{(both squared-norm bounds)}\end{aligned}$$

Therefore,

$$\Pr\left(\|b_S\|_2 > 4\sigma\sqrt{1+\frac{t}{m}}\right) \leq 2e^{-t}.$$

We now move the the "moreover" part of the lemma. Let $t = m\log(4n/\beta)$, where $0<\beta<1$. Since $n \geq 2$, we have $4\sigma\sqrt{1+\log\frac{4n}{\beta}} \leq 8\sigma\sqrt{\log\frac{4n}{\beta}}$. Thus, a union bound over all subsets gives

$$\begin{aligned}&\Pr\left(\text{some nonempty } S\subseteq[n] \text{ satisfies } \|b_S\|_2 > 8\sigma\sqrt{\log\frac{4n}{\beta}}\right)\\ &\quad\leq 2\sum_{m=2}^{n}\binom{n}{m}\exp\left(-m\log\frac{4n}{\beta}\right) && \text{(group subsets by their size)}\\ &\quad\leq 2\sum_{m=2}^{\infty}\left(\frac{\beta}{4}\right)^m && \left(\tbinom{n}{m}\leq n^m\right)\\ &\quad\leq \beta.\end{aligned}$$

Singletons have $b_S = 0$. This proves the high-probability statement, since $\log(4n/\beta) = O(\log(n/\beta))$. Substituting $\beta = \exp(-u)$ into the preceding bound gives, for $u>0$,

$$\Pr\left(\max_S \|b_S\|_2 > 8\sigma\sqrt{\log(4n)+u}\right) \leq e^{-u}.$$

We obtain the moment bound by splitting the possible values into ranges with endpoints $8\sigma\sqrt{\log(4n)+r}$ for positive integers r. Values below the first endpoint contribute at most its z-th power. For each subsequent range, use its upper endpoint to bound the value and the tail probability

at its lower endpoint to bound the probability. This gives

$$\begin{aligned}\mathbb{E}\left[\max_S \|b_S\|_2^z\right] &\le (8\sigma)^z \left(\log(4n)+1\right)^{z/2} + (8\sigma)^z \sum_{r=1}^{\infty} \left(\log(4n)+r+1\right)^{z/2} e^{-r} \\ &\le O_z\left(\sigma^z \left(\log(4n)\right)^{z/2}\right)\left(1+\sum_{r=1}^{\infty}(r+2)^{z/2}e^{-r}\right) \\ &\qquad\qquad (\log(4n)+r+1 \le (r+2)\log(4n)) \\ &= O_z\left(\sigma^z \left(\log(2n)\right)^{z/2}\right). \qquad \text{(the series converges for fixed } z)\end{aligned}$$

This proves the moment statement. □

Combining the noise from all clusters. The way b_S vector is constructed in Lemma E.2 "resembles" the average distances defined by $Q_z(\mathcal{P}, D)$. As such, we can combine the noise of one cluster to bound the error induced by all clusters for the surrogate loss $Q_z(\mathcal{P}, D)$.

Lemma E.3. *For every realization of Z, simultaneously for all partitions $\mathcal{P}$ into exactly k nonempty clusters,*

$$\left|Q_z(\mathcal{P}, \widetilde{D})^{1/z} - Q_z(\mathcal{P}, D)^{1/z}\right| \le \left(\max_S \|b_S\|_2\right) \cdot R(n, k, z),$$

where the term $R(n, k, z)$ is defined as

$$R(n,k,z) = \begin{cases} n^{1/z-1/2}k^{1/2}, & 1 \le z \le 2, \\ k^{1/z}, & z \ge 2. \end{cases}$$

Proof. Fix a partition $\mathcal{P}$. For each $i \in C \in \mathcal{P}$, recall that $r_i(\mathcal{P}, D)$ is the average distance from i to its cluster. Since $\widetilde{D} = D + Z$,

$$\begin{aligned} r_i(\mathcal{P}, \widetilde{D}) - r_i(\mathcal{P}, D) &= \frac{1}{|C|}\sum_{j \in C} Z(i,j) \\ &= b_C(i). \qquad \text{(by definition)}\end{aligned}$$

The quantity $Q_z(\mathcal{P}, D)^{1/z}$ is the ℓ_z norm of $(r_i(\mathcal{P}, D))_{i\in[n]}$. Therefore,

$$\begin{aligned}\left|Q_z(\mathcal{P}, \widetilde{D})^{1/z} - Q_z(\mathcal{P}, D)^{1/z}\right| &= \left|\sum_{i=1}^n |r_i(\mathcal{P}, \widetilde{D})|^z - \sum_{i=1}^n |r_i(\mathcal{P}, D)|^z\right| \qquad \text{(by definition)} \\ &= \left|\|r(\mathcal{P}, \widetilde{D})\|_z - \|r(\mathcal{P}, D)\|_z\right| \\ &= \left(\sum_{C\in\mathcal{P}}\sum_{i\in C} |b_C(i)|^z\right)^{1/z} \qquad (C_i \text{ is the cluster of } i) \\ &= \left(\sum_{C\in\mathcal{P}} \|b_C\|_z^z\right)^{1/z}. \qquad \text{(group coordinates by their cluster)}\end{aligned}$$

For $1 \le z \le 2$, the finite-dimensional norm inequality gives

$$\|b_C\|_z^z \le |C|^{1-z/2}\,\|b_C\|_2^z.$$

Also,

$$\begin{aligned}\sum_{C\in\mathcal{P}} |C|^{1-z/2} &\le k\left(\frac{1}{k}\sum_{C\in\mathcal{P}}|C|\right)^{1-z/2} \qquad \text{(concavity of } x^{1-z/2}) \\ &= n^{1-z/2}k^{z/2}. \qquad (k \text{ clusters with total size } n)\end{aligned}$$

Consequently,

$$\sum_{C\in\mathcal{P}} \|b_C\|_z^z \le \left(\max_S \|b_S\|_2^z\right) \sum_{C\in\mathcal{P}} |C|^{1-z/2}$$
$$\le \left(\max_S \|b_S\|_2^z\right) n^{1-z/2} k^{z/2}.$$

And taking the z-th root proves the first case.

For $z \ge 2$, using $\|b_C\|_z \le \|b_C\|_2$, we can obtain

$$\sum_{C\in\mathcal{P}} \|b_C\|_z^z \le \sum_{C\in\mathcal{P}} \|b_C\|_2^z \le k \max_S \|b_S\|_2^z.$$

Taking the z-th root proves the second case. □

Accurate surrogate values give a low-cost partition. The next lemma shows that if the surrogate loss between D and $\widetilde{D}$ is small enough, then minimizing the $Q_z(\mathcal{P}, \widetilde{D})$ gives good approximation for (k, z)-clustering cost on D.

Lemma E.4. *Let $\widetilde{D}$ be any real symmetric matrix with zero diagonal. Suppose that, for some $a \ge 0$, every partition $\mathcal{P}$ into k nonempty clusters satisfies*

$$\left| Q_z(\mathcal{P}, \widetilde{D})^{1/z} - Q_z(\mathcal{P}, D)^{1/z} \right| \le a.$$

If $\widehat{\mathcal{P}}$ minimizes $Q_z(\mathcal{P}, \widetilde{D})$, then

$$F_z(\widehat{\mathcal{P}}, D)^{1/z} \le 4\,\mathrm{OPT}_{k,z}(D)^{1/z} + 4a.$$

For median and means, respectively, the sharper constants in the bound can be given as

$$F_1(\widehat{\mathcal{P}}, D) \le 2\,\mathrm{OPT}_{k,1}(D) + 2a,$$
$$F_2(\widehat{\mathcal{P}}, D) \le 16\,\mathrm{OPT}_{k,2}(D) + 16a^2.$$

Proof. Let $\mathcal{P}^\star$ minimize the true clustering cost $F_z(\mathcal{P}, D)$. This partition is used only for the analysis. We have

$$\begin{aligned} Q_z(\widehat{\mathcal{P}}, D)^{1/z} &\le Q_z(\widehat{\mathcal{P}}, \widetilde{D})^{1/z} + a && \text{(surrogate accuracy)} \\ &\le Q_z(\mathcal{P}^\star, \widetilde{D})^{1/z} + a && \text{(optimality on the released matrix)} \\ &\le Q_z(\mathcal{P}^\star, D)^{1/z} + 2a. && \text{(surrogate accuracy)} \end{aligned}$$

Applying the comparison between F_z and Q_z from Lemma E.1 gives

$$\begin{aligned} F_z(\widehat{\mathcal{P}}, D)^{1/z} &\le 2Q_z(\widehat{\mathcal{P}}, D)^{1/z} \\ &\le 2Q_z(\mathcal{P}^\star, D)^{1/z} + 4a \\ &\le 4\,\mathrm{OPT}_{k,z}(D)^{1/z} + 4a. \end{aligned}$$

All uses of Lemma E.1 are on the original metric D, not on the possibly nonmetric matrix $\widetilde{D}$.

For $z = 1$, the sharper comparison $F_1 \le Q_1 \le 2F_1$ gives

$$\begin{aligned} F_1(\widehat{\mathcal{P}}, D) &\le Q_1(\widehat{\mathcal{P}}, D) \\ &\le Q_1(\mathcal{P}^\star, D) + 2a \\ &\le 2\,\mathrm{OPT}_{k,1}(D) + 2a. \end{aligned}$$

For $z = 2$, the comparisons $F_2 \le 2Q_2$ and $Q_2 \le 4F_2$ give

$$\begin{aligned} F_2(\widehat{\mathcal{P}}, D)^{1/2} &\le \sqrt{2}\, Q_2(\widehat{\mathcal{P}}, D)^{1/2} \\ &\le \sqrt{2}\, Q_2(\mathcal{P}^\star, D)^{1/2} + 2\sqrt{2}\, a \\ &\le 2\sqrt{2}\ \mathrm{OPT}_{k,2}(D)^{1/2} + 2\sqrt{2}\, a. \end{aligned}$$

Squaring and using $(u+v)^2 \le 2u^2 + 2v^2$ proves the means bound. □

Proof of Theorem 5. Privacy follows from Theorem 1 since Algorithm 2 only conduct post-processing on the released matrix.

For utility, Lemmas E.2 and E.3 give, with probability at least $1-\beta$, a simultaneous bound on the root-surrogate error of

$$O\left(\sigma\sqrt{\log\frac{2n}{\beta}}\right)\cdot R(n,k,z),$$

and recall that

$$R^z(n,k,z)=T(n,k,z)=\begin{cases}n^{1-z/2}k^{z/2}, & 1\le z\le 2,\\ k, & z\ge 2.\end{cases}$$

Apply Lemma E.4 with this bound. Using $(u+v)^z\le 2^{z-1}(u^z+v^z)$, we have

$$\begin{aligned}F_z(\widehat{\mathcal{P}},D)&\le O_z(1)\,\mathrm{OPT}_{k,z}(D)+O_z\left(\sigma^z\left(\log\frac{2n}{\beta}\right)^{z/2}\right)\cdot R^z(n,k,z)\\&=O_z(1)\,\mathrm{OPT}_{k,z}(D)+O_z\left(\sigma^z\left(\log\frac{2n}{\beta}\right)^{z/2}\right)\cdot T(n,k,z).\end{aligned}$$

Substituting $\sigma^z=O(\rho^z n^{z/2}(\log(2/\delta))^{z/2}/\varepsilon^z)$ gives the stated high-probability bound.

For the expectation guarantee, the surrogate-optimization comparison holds on every noise realization. By Lemma E.3, its additive term is bounded by

$$O_z\left(\max_S\|b_S\|_2^z\right)\cdot T(n,k,z)$$

Taking expectations and applying the moment bound in Lemma E.2 therefore gives

$$\mathbb{E}[F_z(\widehat{\mathcal{P}},D)]\le O_z(1)\,\mathrm{OPT}_{k,z}(D)+O_z\left(\sigma^z\,(\log(2n))^{z/2}\right)\cdot T(n,k,z).$$

Substituting σ proves the expectation statement.

Finally, the sharper median and means conclusions of Lemma E.4 give multiplicative factors 2 and 16, with the same respective additive-error rates. □

E.3 A LOWER BOUND FOR DIFFERENTIALLY PRIVATE FACILITY k-CLUSTERING

The lower-bound construction has k possible sites. Each private point chooses one site, and the released partition can be used to guess that choice. Separating the sites far enough forces either many incorrect guesses or large clustering cost. Approximate group privacy limits how accurately the choices can be guessed.

Theorem 6 (k-clustering lower bound)**.** *Fix $z\ge 1$. Suppose $k\le n/8$ is sufficiently large. Let $0<\varepsilon\le 1$, $\delta\le\frac{\varepsilon}{100\sqrt{k}}$, and let* ALG *be an (ε,δ)-private algorithm that outputs a k-clustering with no empty clusters. Suppose for any fixed finite α, the expected cost induced by* ALG *is $\alpha\,\mathrm{OPT}_{k,z}(D)+A$. Then there is an instance D such that*

$$A\ge c_z n\left(\frac{\rho\log k}{\varepsilon}\right)^z.$$

Proof. We first specify the parameter as follows.

$$h=\left\lfloor\frac{\log k}{2\varepsilon}\right\rfloor,\qquad L=h\rho.$$

For sufficiently large k, we have $h\ge 1$, and the above inequality implies $\exp(h\varepsilon)\le\sqrt{k}$.

The hard instance. We define the set of points as

$$V=\{c_1,\dots,c_k\}\cup\{p_1,\dots,p_m\},$$

where $m = n - k$, and all point identities are fixed and public. The points $c_1, \ldots, c_k$ are fixed "reference points", one for each cluster in the optimal clustering. An input is specified by a vector $\theta = (\theta_1, \ldots, \theta_m) \in [k]^m$, where θ_i is the planted cluster label of point p_i.

For each $j \in [k]$, define

$$C_j^\star(\theta) = \{c_j\} \cup \{p_i : \theta_i = j\}, \qquad \mathcal{C}^\star(\theta) = \{C_1^\star(\theta), \ldots, C_k^\star(\theta)\},$$

and it is easy to see that the clusters are non-empty. Fix $\zeta > 0$, for each cluster label specification θ, we define the distance matrix D^θ by

$$D_{uv}^\theta = \begin{cases} 0, & u = v, \\ \zeta, & u \neq v \text{ and } u, v \in C_j^\star(\theta) \text{ for some } j, \\ L + \zeta, & u \in C_j^\star(\theta),\ v \in C_\ell^\star(\theta),\ j \neq \ell. \end{cases}$$

These distances define a strict metric: consider three distinct points u, v, w. If u and v belong to the same $C_j^\star(\theta)$, then $D_{uv}^\theta = \zeta \leq D_{uw}^\theta + D_{wv}^\theta$. Otherwise, at least one of the pairs (u, w) and (w, v) belongs to different planted clusters, so

$$D_{uw}^\theta + D_{wv}^\theta \geq L + \zeta = D_{uv}^\theta.$$

Finally, we can check the optimal clustering cost by choosing c_j as the medoid of $C_j^\star(\theta)$, and the global cost is

$$\mathrm{OPT}_{k,z}(D^\theta) \leq \sum_{j=1}^{k} \sum_{v \in C_j^\star(\theta)} (D_{v,c_j}^\theta)^z \leq n\zeta^z.$$

Changing one planted cluster label. In what follows, we will move a point to another planted cluster to construct datasets with bounded distances. Then, we will use the interpolation of the distance matrices to define $(h+1)$ distance matrices satisfying the ρ-adjacency property. This will allow us to use group privacy as a tool to establish lower bounds later.

Fix an index $i \in [m]$ and all labels other than θ_i. For $a \in [k]$, let $\theta^{(a)}$ be the label vector obtained by setting $\theta_i = a$, and write $D_a = D^{\theta^{(a)}}$ for a short-hand notation. For any $a, b \in [k]$, the matrices D_a and D_b agree outside the row and column corresponding to p_i, and

$$\|D_a - D_b\|_\infty \leq L.$$

Define $h + 1$ matrices explicitly by

$$D^{(t)} = \left(1 - \frac{t}{h}\right) D_a + \frac{t}{h} D_b, \qquad t = 0, \ldots, h.$$

Thus $D^{(0)} = D_a$ and $D^{(h)} = D_b$. Every $D^{(t)}$ is a strict metric: its off-diagonal entries are at least ζ, and each triangle inequality follows by taking the same weighted average of the corresponding triangle inequalities for D_a and D_b.

Moreover, consecutive matrices agree outside the row and column of p_i, and

$$\|D^{(t+1)} - D^{(t)}\|_\infty = \frac{1}{h}\|D_b - D_a\|_\infty \leq \frac{L}{h} = \rho.$$

Hence each consecutive pair is ρ-adjacent.

Let ALG denote the private clustering algorithm, and, with i and θ_{-i} fixed, write

$$\Pr_a(\mathcal{E}) = \Pr(\mathsf{ALG}(D_a) \in \mathcal{E}).$$

Applying differential privacy successively along the sequence $D^{(0)}, \ldots, D^{(h)}$ gives, for every output event $\mathcal{E}$,

$$\Pr_a(\mathcal{E}) \leq \exp(h\varepsilon) \Pr_b(\mathcal{E}) + \delta_h, \qquad \delta_h = \delta \sum_{r=0}^{h-1} e^{r\varepsilon} = \delta \frac{e^{h\varepsilon} - 1}{e^\varepsilon - 1} \leq \frac{\delta\sqrt{k}}{\varepsilon} \leq \frac{1}{100}. \tag{5}$$

Recovering planted cluster labels from an output partition. We now define a new algorithm ALG' that receives the input and the output of ALG, predicts the label of θ_i.

Converting ALG to ALG′

1. Let $\mathcal{P}$ be the output partition. For each $i \in [m]$, we define an estimator of θ_i that is given $\mathcal{P}$ and θ_{-i}, but not θ_i.
2. Let $C_{\mathcal{P}}(p_i)$ be the output cluster containing p_i.
3. For each $j \in [k]$, define the known set
$$C^{\star}_{j,-i} = \{c_j\} \cup \{p_t : t \neq i,\ \theta_t = j\}.$$
4. We define ALG′ that computes the following function.
$$\widehat{\theta}_i(\mathcal{P}; \theta_{-i}) = \min\left(\arg\max_{j \in [k]} \left|C_{\mathcal{P}}(p_i) \cap C^{\star}_{j,-i}\right|\right).$$

That is, among the other points the cluster of p_i outputted by ALG, ALG′ picks the most frequent planted cluster label, breaking ties by the smallest label. If the output cluster is a singleton, all counts are zero and the rule returns 1.

The additional labels θ_{-i} are used only in the lower-bound analysis and as a post-processing step by ALG′: they are not given to ALG and ALG never needs these. Importantly, for fixed θ_{-i}, the rule of ALG′ remains the same for every possible value of θ_i.

Fix i and θ_{-i}, and define

$$\mathcal{E}_a = \{\mathcal{P} : \widehat{\theta}_i(\mathcal{P}; \theta_{-i}) = a\}, \qquad a \in [k],$$

i.e., the partitions that induces the output of ALG′ to be a. These events are disjoint and cover the output space. Let Θ_{-i} be the random variable for θ_{-i}; if θ_i is uniform on $[k]$, then

$$\begin{aligned}
\Pr\left(\widehat{\theta}_i = \theta_i \mid \Theta_{-i} = \theta_{-i}\right) &= \frac{1}{k}\sum_{a=1}^{k} \Pr_a(\mathcal{E}_a) \\
&\leq \frac{1}{k^2}\sum_{a=1}^{k}\sum_{b=1}^{k}\left(\exp(h\varepsilon)\cdot \Pr_b(\mathcal{E}_a) + \delta_h\right) \qquad \text{(Equation (5))} \\
&= \frac{\exp(h\varepsilon)}{k} + \delta_h \\
&\leq \frac{1}{\sqrt{k}} + \frac{1}{100} < \frac{1}{2},
\end{aligned}$$

where the equality in the third line uses $\sum_{a=1}^{k} \Pr_b(\mathcal{E}_a) = 1$ for every b, and the last inequality holds for sufficiently large k.

Now choose $\theta_1, \ldots, \theta_m$ independently and uniformly from $[k]$. Let

$$W = \sum_{i=1}^{m} \mathbf{1}\{\widehat{\theta}_i(\mathcal{P}; \theta_{-i}) \neq \theta_i\}$$

be the number of incorrectly recovered labels by ALG′. Taking expectations over θ and the algorithm's randomness (only ALG has randomness since ALG′ is a deterministic function of ALG), since the succuss probability for an (ε, δ)-DP algorithm has to be low, we have

$$\mathbb{E}[W] \geq \frac{m}{2}.$$

Connecting the errors of ALG′ to the cost of ALG. We now show that if the output of ALG has small clustering cost, then ALG′ must correctly recover most planted cluster labels. Combined with the lower bound on $\mathbb{E}[W]$ above, this will imply the claimed lower bound on clustering cost.

Fix a label vector θ, and let $\mathcal{P}$ be a partition output by $\mathsf{ALG}(D^\theta)$. For each cluster $C \in \mathcal{P}$, choose a medoid

$$c_C \in \arg\min_{c \in C} \sum_{v \in C} (D^\theta_{vc})^z,$$

and let j_C be its planted cluster label, so that $c_C \in C^\star_{j_C}(\theta)$. These medoids are used only to evaluate the clustering cost; neither ALG nor ALG' needs to output or compute them.

Define

$$B_C = C \setminus C^\star_{j_C}(\theta).$$

Thus, B_C consists of the points in C whose planted cluster label differs from that of the medoid c_C. Every point in B_C is at distance $L + \zeta$ from c_C, and therefore

$$F_z(\mathcal{P}, D^\theta) = \sum_{C \in \mathcal{P}} \sum_{v \in C} (D^\theta_{v,c_C})^z \geq L^z \sum_{C \in \mathcal{P}} |B_C|. \tag{6}$$

We next show that ALG' makes at most $2|B_C| + 1$ errors among the private points in C. Recall that, to predict θ_i, ALG' removes p_i from its output cluster and returns the most frequent planted cluster label among the remaining points.

Let us first assume w.log. that $|C| > 2|B_C| + 1$. Otherwise, if $|C| \leq 2|B_C| + 1$ ALG', cannot make prediction error more than $2|B_C| + 1$ by pigeonhole principle. Consider any private point $p_i \in C \setminus B_C$, whose true label is j_C. After removing p_i, the number of remaining points with label j_C is

$$|C| - |B_C| - 1 > |B_C|.$$

On the other hand, the total number of points with labels different from j_C is only $|B_C|$. Hence, j_C is the unique most frequent label among the remaining points, and ALG' correctly returns $\widehat{\theta}_i = j_C$. Consequently, only points in B_C can be predicted incorrectly, giving at most $|B_C| \leq 2|B_C| + 1$ errors in C.

Summing over the k output clusters and using (6), we obtain

$$W \leq \sum_{C \in \mathcal{P}} (2|B_C| + 1) \leq \frac{2}{L^z} F_z(\mathcal{P}, D^\theta) + k.$$

This inequality holds for every θ and every output partition $\mathcal{P}$.

Taking expectations over the independent uniform labels $\theta_1, \ldots, \theta_m$ and the randomness of ALG, and using the previously established $\mathbb{E}[W] \geq m/2$, gives

$$\begin{aligned}
\mathbb{E}_{\theta, \mathsf{ALG}}\big[F_z(\mathcal{P}, D^\theta)\big] &\geq \frac{L^z}{2}\big(\mathbb{E}[W] - k\big) \\
&\geq \frac{L^z}{2}\left(\frac{n-k}{2} - k\right) \\
&= \frac{n - 3k}{4} L^z = \Omega(nL^z),
\end{aligned}$$

where the last equality uses $k \leq n/8$.

On the other hand, the assumed approximation guarantee holds for every input D^θ. Since $\mathrm{OPT}_{k,z}(D^\theta) \leq n\zeta^z$, averaging this guarantee over θ yields

$$\mathbb{E}_{\theta, \mathsf{ALG}}\big[F_z(\mathcal{P}, D^\theta)\big] \leq \alpha n \zeta^z + A.$$

Combining the two bounds, there is an absolute constant $c > 0$ such that

$$A \geq cnL^z - \alpha n \zeta^z.$$

The guarantee must hold for every $\zeta > 0$ with the same α and A. Letting ζ tend to zero and using $L = \Theta(\rho \log k / \varepsilon)$ gives

$$A = \Omega_z\left(n\left(\frac{\rho \log k}{\varepsilon}\right)^z\right),$$

as claimed. □

E.4 LOWER BOUND FOR PURELY ADDITIVE ERRORS

The upper bounds we obtained for k-clustering in Appendices E.2, F and G contain multiplicative factors on OPT. As such, one may naturally wonder what would happen if we insist on purely additive errors for DP k-clustering. In this section, we will show that such an objective will lead to much worse approximation guarantees, as we can prove additive error of $\Omega(\rho n^{3/2}/\varepsilon)$ for (ε, δ)-DP algorithms and $\Omega(\rho n^2/\varepsilon)$ for pure ε-DP algorithms. The hardness result is given as follows for the simple $k = 2$ case, i.e., 2-median.

Theorem 7. *For any sufficiently large n, $0 < \varepsilon \leq 1$, and $0 \leq \delta \leq \frac{\varepsilon}{n}$, there exists a family of metrics such that for any DP k-cluster algorithm with only additive error, the following statements are true*

1. *If the algorithm is pure ε-private, the additive error is at least $\Omega(\rho n^2/\varepsilon)$.*
2. *If the algorithm is approximate (ε, δ)-DP, the additive error is at least $\Omega(\rho n^{3/2}/\varepsilon)$*

Both lower bounds are attained with $k = 2$.

Proof overview. Both lower bounds use the same family of metrics with a construction similar to the MST lower bound as in Theorem 4. The high-level idea is to use the "double star" idea in the MST lower bound proof, and show that a low-error 2-clustering algorithm will imply a high-precision algorithm that predicts the signs in a database, which has been proved to be an impossible task with a low number of rows. On the other hand, the number of rows is built into the normalization factor of the prediction, which will be counted as a multiplicative factor in the error.

The complication here is that a clustering algorithm may *not* separate the two "planted" centers p and q. In other words, it is possible for the output clustering to have p and q in the same cluster. To enforce metric property, we will need to add large penalties to pairs that belongs to different planted clusters. However, this would create another issue: the metric properties will be violated. To fix the issue, we also add a large *baseline distance* to all pairs of points. This is also crucial for why the lower bound only works for *purely additive* error: once multiplicative error is allowed, the multiplicative factor on the optimal clustering cost would be able to absorb the additive term.

We now proceed to the main proof. Let ALG output exactly two nonempty clusters, and suppose that

$$\mathbb{E}_{\mathsf{ALG}}[F_1(\mathsf{ALG}(D), D)] \leq \mathrm{OPT}_{2,1}(D) + A$$

for every strict metric D, where A does not depend on the input.

The hard instance. Let $m = n - 2 \geq 3$, and let $H > 0$ be a parameter to be specified later. Define

$$L = (m+1)H, \qquad B = 2L, \qquad V = \{p, q, s_1, \ldots, s_m\}.$$

All point identities and the parameters H, L, B are public. An input is specified by a vector $x = (x_1, \ldots, x_m) \in [0, H]^m$. For every such vector, define the symmetric distance matrix D_x by

$$D_x(u,v) = \begin{cases} 0, & u = v, \\ B + x_j, & \{u,v\} = \{p, s_j\} \text{ for some } j \in [m], \\ B + H/2, & \{u,v\} = \{q, s_j\} \text{ for some } j \in [m], \\ B + L, & \text{otherwise.} \end{cases} \tag{7}$$

We intend to force p and q into different clusters, so that each point s_j is assigned to the cluster containing p or to the cluster containing q. Since the distance between p and s_j points are specified by a vector x, we write p_x for such a point p. When x is fixed, we write $D(u,v)$ for $D_x(u,v)$, the entry corresponding to points u, v. The corresponding distances are $B + x_j$ and $B + H/2$, so the cheaper choice depends on whether x_j is below or above $H/2$. The distances between p and q, and between any two distinct satellites, are $B + L$. These distances will ensure that putting p and q together has large cost.

We first verify that D_x is a strict metric. All off-diagonal entries lie in $[B, B + L]$, and $B \geq L > 0$. Thus, for any three distinct points u, v, w,

$$D(u,v) \leq B + L \leq 2B \leq D(u,w) + D(w,v).$$

This is where the large baseline B is used: it is to ensure the construction is a valid metric.

The group privacy trick for the input vector. The change of a row violates the neighboring condition with parameter ρ, but this can be similarly fixed with the group privacy trick as specified by the proof of Theorems 4 and 6. Only the distances $D(p_x, s_j)$ depend on x. Hence, for any $x, x' \in [0, H]^m$, the matrices $D_x, D_{x'}$ agree outside the row and column of p_x, and

$$\|D_x - D_{x'}\|_\infty = \|x - x'\|_\infty.$$

In particular, let $h \geq 1$ be an integer such that $\|x - x'\|_\infty \leq h\rho$. Define

$$x^{(t)} = \left(1 - \frac{t}{h}\right) x + \frac{t}{h} x', \qquad D^{(t)} = D_{x^{(t)}}, \qquad t = 0, \ldots, h.$$

Every $x^{(t)}$ belongs to $[0, H]^m$, so every $D^{(t)}$ is a strict metric. Consecutive matrices agree outside the row and column of p, and

$$\|D^{(t+1)} - D^{(t)}\|_\infty = \frac{\|x - x'\|_\infty}{h} \leq \rho. \tag{8}$$

This gives an explicit sequence of h adjacent steps between D_x and $D_{x'}$.

The optimal 2-clustering vs. other cluster structures. For any partition $\mathcal{P}$ into two nonempty clusters, every choice of medoids pays the common contribution (baseline) B exactly $n - 2$ times. Thus

$$F_1(\mathcal{P}, D_x) = B(n-2) + \sum_{C \in \mathcal{P}} \min_{c \in C} \sum_{v \in C \setminus \{c\}} \big(D(v, c) - B\big).$$

If p and q are separated, write $\mathcal{P} = \{C_p, C_q\}$ with $p \in C_p$ and $q \in C_q$. The point p is a best medoid of C_p: replacing it by a satellite $s_j \in C_p$ leaves the contribution between p, s_j unchanged, and changes the cost by

$$\sum_{\substack{\ell : s_\ell \in C_p \\ \ell \neq j}} (L - x_\ell) \geq 0.$$

A singleton cluster is immediate, and the same argument applies to q in C_q. Therefore,

$$F_1(\mathcal{P}, D_x) = B(n-2) + \sum_{j : s_j \in C_p} x_j + \sum_{j : s_j \in C_q} \frac{H}{2}. \tag{9}$$

The best such partition assigns each s_j to the center with the lower cost.

If p, q are together, the other cluster contains only satellites. If it contains at least two points, every medoid pays $B + L$ to another satellite. Otherwise it is a singleton, and the cluster containing p, q contains $m - 1 \geq 2$ satellites. Every medoid there also pays $B + L$: a hub serves the other hub, and a satellite serves another satellite. Thus, every partition putting p, q together satisfies

$$F_1(\mathcal{P}, D_x) \geq B(n-2) + L. \tag{10}$$

Since $\sum_j \min\{x_j, H/2\} \leq mH/2 < L$, such a partition cannot be optimal. We have therefore proved the exact formula

$$\mathrm{OPT}_{2,1}(D_x) = B(n-2) + \sum_{j=1}^{m} \min\{x_j, H/2\}. \tag{11}$$

We give labels for Equation (10) and Equation (11) since they are going to be used in later proofs as well.

Recovering binary decisions from an output partition. For each j, the decision 0 means choosing p and paying x_j beyond the common contribution; 1 means choosing q and paying $H/2$. Given the partition $\mathcal{P}$ output by $\mathsf{ALG}(D_x)$, define ALG' as follows.

Converting ALG to ALG′

1. If p, q belong to different clusters, let C_p contain p and C_q contain q. Return $\widehat{u} \in \{0,1\}^m$ with
$$\widehat{u}_j = \begin{cases} 0, & s_j \in C_p, \\ 1, & s_j \in C_q. \end{cases}$$
2. Otherwise, return $\widehat{u} = (1, \ldots, 1)$.

This algorithm uses only the output partition and the public point identities. It does not receive x, inspect the original distance matrix, or compute the medoids, which ensures the algorithm can be implemented.

For any prediction vector $\widehat{u}$, define its error by

$$\mathrm{err}_x(\widehat{u}) = \sum_{j=1}^{m} \left((1 - \widehat{u}_j) x_j + \widehat{u}_j \frac{H}{2} - \min\{x_j, H/2\} \right).$$

Each coordinate contributes zero for a correct choice and $|x_j - H/2|$ for an incorrect choice; either choice has zero error at a tie.

Connecting the error of ALG′ to the cost of ALG. For every x and every output partition $\mathcal{P}$, we have

$$\mathrm{err}_x\big(\mathsf{ALG}'(\mathcal{P})\big) \le F_1(\mathcal{P}, D_x) - \mathrm{OPT}_{2,1}(D_x). \tag{12}$$

If p, q are separated, (9) and (11) give equality. Otherwise, ALG′ returns all ones, and $mH/2 < L$ gives

$$\begin{aligned} \mathrm{err}_x(\widehat{u}) &= \frac{mH}{2} - \sum_{j=1}^{m} \min\{x_j, H/2\} \\ &\le L - \sum_{j=1}^{m} \min\{x_j, H/2\} \\ &\le F_1(\mathcal{P}, D_x) - \mathrm{OPT}_{2,1}(D_x), \end{aligned}$$

where the last inequality uses (10) and (11). We are now ready to prove the lower bound in Theorem 7.

Proof of Theorem 7. We prove the lower bounds for approximate and pure DP, respectively.

The lower bound for approximate differential privacy. Let

$$N = \left\lfloor \frac{\sqrt{m}}{64} \right\rfloor, \qquad r = \left\lfloor \frac{1}{\varepsilon} \right\rfloor, \qquad R = r\rho, \qquad H = RN.$$

For sufficiently large n, we have $N \ge 2$, $N \ge \sqrt{m}/128$, and $R \ge \rho/(2\varepsilon)$.

We use an auxiliary database $X \in \{0,1\}^{N \times m}$ with N rows to encode the dataset. For each $j \in [m]$, independently draw P_j uniformly from $[0,1]$. Conditional on $P = (P_1, \ldots, P_m)$, draw all entries of X independently with

$$\Pr[X_{ij} = 1 \mid P] = P_j.$$

Define

$$c_j = \sum_{i=1}^{N} X_{ij}, \qquad \bar{X}_j = \frac{c_j}{N}, \qquad x_j = Rc_j = H\bar{X}_j.$$

In particular, $x \in [0, H]^m$.

To apply the fingerprinting lower bound, define an algorithm $\mathcal{M}$ on binary databases explicitly:

1. Given X, compute the counts c_j, set $x_j = Rc_j$, and construct D_x using (7).
2. Run $\mathsf{ALG}(D_x)$ to obtain $\mathcal{P}$, and run $\mathsf{ALG}'(\mathcal{P})$ to obtain $\widehat{u}$.
3. Return $v = \mathcal{M}(X) \in \{-1, +1\}^m$, where $v_j = 2\widehat{u}_j - 1$.

The values P_j are used only to define the distribution in the analysis, and $\mathcal{M}$ does *not* receive them.

We first verify the privacy of $\mathcal{M}$. If X, X' differ in one row, their count vectors differ by at most one in every coordinate. Thus the corresponding x, x' satisfy $\|x - x'\|_\infty \le R = r\rho$. Applying (8) with $h = r$ connects D_x to $D_{x'}$ by r adjacent steps. Group privacy and the deterministic post-processing by ALG' therefore imply that $\mathcal{M}$ is $(r\varepsilon, \delta_r)$-private under replacement of one database row, where

$$r\varepsilon \le 1, \qquad \delta_r = \delta \sum_{\ell=0}^{r-1} e^{\ell\varepsilon} \le er\delta \le \frac{e\delta}{\varepsilon} \le \frac{e}{n}.$$

For sufficiently large n, $e/n \le 1/(16N)$, so $\mathcal{M}$ is also $(1, 1/(16N))$-private.

Next, for $v \in \{-1, +1\}^m$, define

$$U_X(v) = \sum_{j=1}^{m} v_j \left(\bar{X}_j - \frac{1}{2} \right).$$

Let $v_j^\star = +1$ when $\bar{X}_j \ge 1/2$, and $v_j^\star = -1$ otherwise. Then

$$U_X(v^\star) = \sum_{j=1}^{m} \left| \bar{X}_j - \frac{1}{2} \right|, \qquad \mathrm{err}_x(\widehat{u}) = \frac{H}{2}\big(U_X(v^\star) - U_X(v)\big).$$

The second identity follows by checking the two choices for each coordinate: choosing the wrong sign decreases utility by $2|\bar{X}_j - 1/2|$ and increases prediction error by $H|\bar{X}_j - 1/2|$.

We now lower-bound the expected optimal utility. For every $t \in \{0, \dots, N\}$,

$$\Pr[c_j = t] = \binom{N}{t} \int_0^1 p^t (1-p)^{N-t}\, dp = \frac{1}{N+1}.$$

Hence each count is uniform on $\{0, \dots, N\}$, and

$$\mathbb{E}\left| \bar{X}_j - \frac{1}{2} \right| = \frac{1}{N+1} \sum_{t=0}^{N} \left| \frac{t}{N} - \frac{1}{2} \right| \ge \frac{1}{4}.$$

It follows that $\mathbb{E}[U_X(v^\star)] \ge m/4$.

Suppose, for a contradiction, that $A < Hm/32$. The assumed clustering guarantee holds for every D_x, so averaging it over the random database and using (12) gives

$$\mathbb{E}[\mathrm{err}_x(\widehat{u})] \le A < \frac{Hm}{32}.$$

Therefore,

$$\mathbb{E}[U_X(\mathcal{M}(X))] = \mathbb{E}[U_X(v^\star)] - \frac{2}{H}\mathbb{E}[\mathrm{err}_x(\widehat{u})] > \frac{m}{4} - \frac{m}{16} = \frac{3m}{16}.$$

Here and below, expectations include the hidden probabilities, the database, and the randomness of ALG when relevant.

To connect this utility to the hidden probabilities P_j, condition on the entire database X. Independence across columns gives

$$\mathbb{E}[P_j \mid X] = \frac{\int_0^1 p^{c_j+1}(1-p)^{N-c_j}\, dp}{\int_0^1 p^{c_j}(1-p)^{N-c_j}\, dp} = \frac{c_j + 1}{N+2},$$

which implies $\mathbb{E}[P_j - 1/2 \mid X] = \frac{N}{N+2}\left(\bar{X}_j - \frac{1}{2}\right)$. Conditional on X, the randomness used by $\mathcal{M}$ is independent of P. Consequently,

$$\mathbb{E}\left[\sum_{j=1}^{m} \mathcal{M}(X)_j (P_j - 1/2)\right] = \frac{N}{N+2}\mathbb{E}[U_X(\mathcal{M}(X))] > \frac{1}{2}\cdot\frac{3m}{16} > \frac{m}{16}.$$

We can now apply Proposition D.5 with parameters $\beta = 1$, $\gamma = 1/16$, and output dimension and expected squared norm both equal to m. In this specialization, a $(1, 1/(16N))$-private algorithm with the above correlation must satisfy

$$N \geq \frac{\sqrt{m}}{16}.$$

The norm conditions hold because $\mathcal{M}(X) \in \{-1, +1\}^m$ always, so $\|\mathcal{M}(X)\|_1 = \|\mathcal{M}(X)\|_2^2 = m$. This contradicts $N = \lfloor\sqrt{m}/64\rfloor$. It follows that

$$A \geq \frac{Hm}{32} = \frac{RNm}{32} \geq \frac{\rho m^{3/2}}{8192\varepsilon} = \Omega\left(\frac{\rho n^{3/2}}{\varepsilon}\right).$$

The lower bound for pure differential privacy. We use the same construction and the same algorithm ALG', but choose the input vector differently. Let

$$h = \left\lfloor \frac{m}{32\varepsilon} \right\rfloor, \qquad H = h\rho.$$

For sufficiently large n, $h \geq 1$ and $h \geq m/(64\varepsilon)$. For each $u \in \{0,1\}^m$, define

$$x(u) = Hu, \qquad D_u = D_{x(u)},$$

where $D_{x(u)}$ means the x vector that parametrizes p_x is specified by $x(u)$. The algorithm runs $\mathsf{ALG}(D_u)$, and returns $\widehat{u} = \mathsf{ALG}'(\mathsf{ALG}(D_u))$. Define the Hamming distance between two binary vectors by

$$\mathrm{Ham}(u, y) = \sum_{j=1}^{m} \mathbf{1}\{u_j \neq y_j\}.$$

Because each x_j is either 0 or H, every wrong prediction costs exactly $H/2$. Thus

$$\mathrm{err}_{x(u)}(\widehat{u}) = \frac{H}{2}\,\mathrm{Ham}(\widehat{u}, u). \tag{13}$$

For any two vectors u, u', we have $\|x(u) - x(u')\|_\infty \leq H = h\rho$. By (8), the corresponding metrics are connected by h adjacent steps. This remains true even when all m bits change: only the distance row and column of p change, and all of their entries can change simultaneously.

For any event $\mathcal{E} \subseteq \{0,1\}^m$, write

$$\Pr_u(\mathcal{E}) = \Pr\left(\mathsf{ALG}'(\mathsf{ALG}(D_u)) \in \mathcal{E}\right).$$

Pure group privacy gives

$$\Pr_u(\mathcal{E}) \leq \exp(h\varepsilon)\cdot\Pr_{u'}(\mathcal{E}) \leq \exp(m/32)\cdot\Pr_{u'}(\mathcal{E}).$$

For each u, let

$$\mathcal{E}_u = \{y \in \{0,1\}^m : \mathrm{Ham}(y, u) \leq m/4\}.$$

This is the event that the estimator makes at most $m/4$ errors on input u.

We compare every input with the fixed reference input $u' = \mathbf{0}$. For any fixed output vector y, the number of vectors u with $y \in \mathcal{E}_u$ is

$$\sum_{j=0}^{\lfloor m/4 \rfloor} \binom{m}{j}.$$

Indeed, one chooses at most $m/4$ coordinates in which u differs from y. Averaging over the output on the reference input therefore gives

$$\begin{aligned}
\frac{1}{2^m} \sum_{u \in \{0,1\}^m} \Pr_{\mathbf{0}}(\mathcal{E}_u) &= \frac{1}{2^m} \sum_{u \in \{0,1\}^m} \sum_{y \in \{0,1\}^m} \Pr\left(\mathsf{ALG}'(D_{\mathbf{0}}) = y\right) \cdot \mathbf{1}\{y \in \mathcal{E}_u\} \\
&= \frac{1}{2^m} \sum_{j=0}^{\lfloor m/4 \rfloor} \binom{m}{j} \cdot \sum_{y \in \{0,1\}^m} \Pr\left(\mathsf{ALG}'(D_{\mathbf{0}}) = y\right) \\
&= \frac{1}{2^m} \sum_{j=0}^{\lfloor m/4 \rfloor} \binom{m}{j}.
\end{aligned}$$

The events $\mathcal{E}_u$ need not be disjoint; the displayed identity counts how many of them contain each fixed output.

Combining the above calculation with group privacy yields

$$\begin{aligned}
\frac{1}{2^m} \sum_{u \in \{0,1\}^m} \Pr_{u}(\mathcal{E}_u) &\le \exp(m/32) \cdot \frac{1}{2^m} \sum_{u \in \{0,1\}^m} \Pr_{\mathbf{0}}(\mathcal{E}_u) && \text{(group privacy)} \\
&= \exp(m/32) \cdot \frac{1}{2^m} \sum_{j=0}^{\lfloor m/4 \rfloor} \binom{m}{j} && \text{(by the earlier calculation)} \\
&\le \exp(m/32 - m/8) \le \frac{1}{2}.
\end{aligned}$$

The second last inequality is Hoeffding's bound for $\Pr(\mathrm{Binomial}(m, 1/2) \le m/4)$, and the last inequality holds for sufficiently large m.

Now choose u uniformly from $\{0,1\}^m$. The probability that the estimator makes more than $m/4$ errors is at least $1/2$, so

$$\mathbb{E}_{u,\mathsf{ALG}}[\mathrm{Ham}(\widehat{u}, u)] \ge \frac{m}{8}.$$

Using (12) and (13), we obtain

$$\begin{aligned}
&\mathbb{E}_{u,\mathsf{ALG}}\left[F_1(\mathsf{ALG}(D_u), D_u) - \mathrm{OPT}_{2,1}(D_u)\right] \\
&\quad \ge \frac{H}{2} \mathbb{E}_{u,\mathsf{ALG}}[\mathrm{Ham}(\widehat{u}, u)] \ge \frac{Hm}{16}.
\end{aligned}$$

The assumed guarantee bounds the left-hand side by A on average, because it holds on every input. Therefore,

$$A \ge \frac{Hm}{16} = \frac{h\rho m}{16} \ge \frac{\rho m^2}{1024\varepsilon} = \Omega\left(\frac{\rho n^2}{\varepsilon}\right).$$

This completes both parts of the theorem. □

F Pure-DP for k-clustering with Exponential Mechanism

We give an algorithm using exponential mechanism for k-clustering in this section. This result can be treated as a "baseline" for our k-clustering algorithm to compare with. Our algorithms matches the bound for exponential mechanism when $k = O(1)$.

Proposition F.1 (Pure-private exponential mechanism)**.** *For every fixed $z \geq 1$, the exponential mechanism with score $-Q_z(P, D)^{1/z}$ is ε-private and, with probability at least $1 - \beta$, returns a partition satisfying*

$$F_z(\widehat{P}, D)^{1/z} \leq 4\,\mathrm{OPT}_{k,z}(D)^{1/z} + \frac{8\rho}{\varepsilon}\left(n \log k + \log(1/\beta)\right).$$

For $z \geq 2$, the coefficient 8 in the additive term can be replaced by 4. For k-median ($Z = 1$), the sharper statement is

$$F_1(\widehat{P}, D) \leq 2\,\mathrm{OPT}_{k,1}(D) + \frac{4\rho}{\varepsilon}\left(n \log k + \log(1/\beta)\right).$$

Proof. Fix a partition $\mathcal{P}$ and suppose the element with index i moves. Only the average-distance value $r_j(\mathcal{P}, D)$ of a point j in the same cluster as i changes. If that cluster has size m, $r_i(\mathcal{P}, D)$ changes by at most $\rho(m-1)/m$, while $r_j(\mathcal{P}, D)$ for each of the other $m-1$ element j in the same cluster changes by at most ρ/m. For every $z \geq 1$, the z-norm of the change to vector $(r_1, r_2, \cdots, r_n)$ is at most 2ρ; for $z \geq 2$, it is at most ρ. These also bound the sensitivity of $Q_z(\mathcal{P}, D)^{1/z}$.

There are at most k^n labeled partitions. The standard exponential mechanism guarantee bounds the selected root surrogate by its minimum plus at most

$$\frac{4\rho}{\varepsilon}\left(n \log k + \log(1/\beta)\right).$$

For $z \geq 2$, the numerator is 2ρ instead. Applying Lemma E.1 proves both statements. Treating unlabeled partitions as labeled only enlarges the output count. □

For k-median, the mechanism samples a partition with probability proportional to $\exp(-\varepsilon Q_1(\mathcal{P}, D)/(4\rho))$. Integrating its tail bound gives an expected cost at most

$$2\,\mathrm{OPT}_{k,1} + 4\rho(n \log k + 1)/\varepsilon.$$

This is an exact exponential mechanism over all partitions, not a polynomial-time sampler. The general-z statement is a baseline; for k-means, its squared entropy term is worse than the Gaussian result below.

G Recursive k-clustering

In order to get a polynomial running time algorithm, we will perform metric release and clustering in a *recursive* manner. Specifically, we partition the input points into groups (arbitrarily, e.g., just by index), so that recursively we compute private k-clustering for each group. Now we take the union of all the clusters and perform another round of clustering on top of these groups, during which we add a fresh set of random noise. To control the total additive error, we treat each cluster as a weighted "super node" and evaluate the distance between two clusters using the average pairwise distance between their points. The fresh noise is imposed on top of these average distances. Then we run a constant-approximate k-clustering algorithm on these super nodes by using local search methods. We use composition to derive the final differential privacy bound.

In the following, we first present an outline of the recursive clustering algorithm (Appendix G.1), followed by the modules for k-clustering by local search (Appendix G.2), the creation of supernodes and the distances between them (Appendix G.3), and the DP metric release for each layer (Appendix G.4). We then analyze the running time and privacy guarantee (Appendix G.5).

In the following discussion, we focus on $z = 1$, i.e., the k-median problem. The general case of other values of z is similar and omitted.

G.1 Recursive Clustering Algorithm

The total depth in the recursive algorithm is an integer $t \geq 0$. Define

$$a_s = \frac{1}{1 - (1/3)^{s+1}} \quad (s = 0, 1, \ldots, t). \tag{14}$$

In particular $a_0 = 3/2$. Every Gaussian release in the recursion uses

$$\varepsilon_0 = \frac{\varepsilon}{t+1}, \qquad \delta_0 = \frac{\delta}{t+1}, \qquad \beta_0 = \frac{\beta}{n(t+1)}.$$

The procedure on a public set S of m original indices and remaining depth s is as follows.

1. If $m \le k$, return the singleton partition.
2. If $s = 0$, apply the weighted distance release of Lemma G.4 to singleton groups of S, with unit weights. Return its k-partition.
3. If $s \ge 1$, let
$$\theta_s = \frac{1}{2a_{s-1}+1}, \qquad b = \min\left\{m, \left\lceil k(m/k)^{\theta_s} \right\rceil\right\}.$$
Partition S into public blocks of size at most b, using their fixed index order. Recursively cluster each block with remaining depth $s-1$.
4. Take the union $\mathcal{A}$ of the resulting groups. If its number of groups is $N = k$, return those groups. Otherwise, compute their average-distance matrix (by Equation (20)) from the *original* submatrix $D_{S,S}$. Apply Lemma G.4, using fresh Gaussian noise, to cluster the N weighted groups into k clusters.
5. Expand the group clusters into unions of original indices.

The top-level call has $S = [n]$ and $s = t$.

G.2 Local Search Algorithm

A pseudometric (X, d) is a metric except that the idempotent property can be dropped, i.e., $d(x, x)$ may not be zero. We first show that we can still perform local search on a pseudometric and obtain a constant-factor approximation for k-median, as in the metric setting (Arya et al., 2004). The main argument in the local search algorithm only requires that the distances are symmetric and satisfy the triangle inequality. Further, we ask that all local swaps make sufficient progress, thereby bounding the number of swaps.

Lemma G.1. *Let H be a pseudometric on N points Q with positive integer weights $w_1, \dots, w_N$ and total weight m. Consider the weighted k-median problem, in which for a fixed $k \le N$ we aim to select k centers C among Q such that $\sum_{i \in Q} \min_{c \in C} w_i H(i, c)$ is minimized. The algorithm below finds k centers C with weighted cost at most $20/3$ times the optimal weighted cost, in $O(N^2 k^2 \log m)$ arithmetic operations.*

1. *Initialize k centers by farthest-first traversal, where the first center is chosen arbitrarily and the ith center is selected as a distinct point that is furthest from the $i-1$ centers selected so far.*
2. *If replacing one existing center in C by a point not in C reduces the current weighted cost by more than a $1/(4k)$ fraction, make such a swap. Repeat until no such swap exists.*
3. *Assign points to nearest centers, with each center assigned to itself.*

Proof. Denote by L the set of k centers returned by the algorithm and by O a set of optimal centers. Let $L(j)$ and $O(j)$ denote the center serving j in the local and optimal solutions. For a point i, let $r_i = H(i, L(i))$ and $r_i^* = H(i, O(i))$. The clustering cost of L and O is

$$C = \sum_{i \in Q} w_i r_i, \text{ and } C^* = \sum_{i \in Q} w_i r_i^*$$

respectively. Our goal is to show that $C \le (20/3) C^*$.

We map each optimal facility $o \in O$ to its closest open facility in L via $\pi(o) = \arg\min_{\ell \in L} H(o, \ell)$. Specifically, if $o \in L$, $\pi(o) = o$. We construct a bijective function $\phi : O \to L$ such that if ℓ has any optimal facilities mapped to it, $\phi^{-1}(\ell)$ is exactly one of those mapped optimal facilities (a representative).

We evaluate k potential swaps of the form $(\phi(o), o)$. Let $\Delta(o)$ be the change in total cost resulting from the swap $(\phi(o), o)$. Because the algorithm terminated, no single swap reduces the cost by more than $\frac{C}{4k}$. Summing the minimum cost changes over all k swaps:

$$\sum_{o \in O} \Delta(o) \geq \sum_{o \in O} \left(-\frac{C}{4k} \right) = -\frac{C}{4} \tag{15}$$

When a swap $(\phi(o), o)$ occurs, we consider a theoretical reassignment:

1. Any client j with $O(j) = o$ is optimally reassigned to o.
2. Any client j with $L(j) = \phi(o)$ and $O(j) \neq o$ is orphaned.
   - If $\phi(O(j)) = \pi(O(j))$ (meaning $O(j)$ is the unique representative for $\pi(O(j))$), we reassign j to $\pi(O(j))$.
   - If $\phi(O(j)) \neq \pi(O(j))$, we reassign j to the unique representative $\bar{o} \in O_{\pi(O(j))}$ that is guaranteed to be opened when $\pi(O(j))$ is closed.

Following the structural analysis of Arya et al. (2004), we sum these reassignment costs across all k swaps for each client j:

- The optimal facility $O(j)$ is opened in exactly one swap. In this swap, client j's weighted cost changes by $w_j H(j, O(j)) - w_j H(j, L(j))$.
- The current facility $L(j)$ is closed in exactly one swap. In this swap, the distance to the backup facility (either $\pi(O(j))$ or $\bar{o}$) is bounded using the triangle inequality. The aggregate increase in weighted cost over all clients is bounded by $4w_j H(j, O(j))$ due to the properties of the mapping π. (Note: This structural bounding is similar to (Arya et al., 2004) and relies entirely on the triangle inequality and symmetry, remaining valid even if $H(x, x) > 0$).

Summing these structural bounds over all clients and all k swaps yields:

$$\sum_{o \in O} \Delta(o) \leq \sum_{j \in C} (w_j H(j, O(j)) - w_j H(j, L(j)) + 4w_j H(j, O(j))) \tag{16}$$

$$\leq \sum_{j \in C} 5w_j H(j, O(j)) - \sum_{j \in C} w_j H(j, L(j)) \tag{17}$$

$$\leq 5C^* - C \tag{18}$$

Combined with Equation (15) we have,

$$C \leq (20/3)C^* \tag{19}$$

For the running time, farthest-first traversal has maximum service distance at most twice the optimal k-center radius. This follows by considering the k selected points and a farthest remaining point: two of these $k + 1$ points share an optimal k-center ball. Since every weight is at least one, the optimal k-center radius is at most C^*. Thus the initial cost is at most $2mC^*$. When $C^* = 0$, the initial cost is already zero. Otherwise every accepted swap reduces cost by a factor smaller than $1 - 1/(4k)$, so there are at most $O(k \log m))$ swaps. Enumerating and evaluating all possible single swaps in one round takes $O(N^2 k)$. Thus the total running time is at most $O(N^2 k^2 \log m)$. □

G.3 Replacing groups by a weighted metric

Let $\mathcal{A} = \{A_1, \ldots, A_N\}$ be any partition of $[m]$ into nonempty groups. Define $a_j = |A_j|$ as the weight of the group A_j. Define

$$B_{uv} = \frac{1}{a_u a_v} \sum_{i \in A_u} \sum_{j \in A_v} D_{ij} \quad (u \neq v), \qquad B_{uu} = 0. \tag{20}$$

We can verify that the matrix B is a metric. Recall that $D_{ij} = D_{ji}$. Thus $B_{uv} = B_{vu}$. For three distinct groups A_u, A_v, A_w, average $D_{ij} \le D_{ih} + D_{hj}$ over $i \in A_u$, $h \in A_v$, and $j \in A_w$. This gives $B_{uw} \le B_{uv} + B_{vw}$. The cases involving repeated group indices follow from $B_{uu} = 0$.

The partition $\mathcal{A}$ can be understood as a set of weighted points with metric B. For a partition of the partition $\mathcal{A}$, denoted by $\mathcal{P}$, its cost can be defined as the cost of the weighted points $\mathcal{A}$ with metric B,

$$G(\mathcal{P}, B, a) = \sum_{J \in \mathcal{P}} \sum_{u \in J} a_u B_{uc(J)}, \text{ where } c(J) = \operatorname*{argmin}_{c \in J} \sum_{u \in J} a_u B_{uc}.$$

Similarly, denote the optimal k-median cost with metric B and weight a by $\mathrm{OPT}_k(B, a)$. Recall that the cost of the partition $\mathcal{A}$ is $F_1(\mathcal{A}, D)$, using one medoid per group. The following lemma says that the optimal k-clustering of the weighted points for $\mathcal{A}$ can be bounded by the optimal k-clustering of the original metric D and the cost of the partitioning $\mathcal{A}$.

Lemma G.2 (Compression). *For $k \le N$,*

$$\mathrm{OPT}_k(B, a) \le 2\,\mathrm{OPT}_k(D) + 4F_1(\mathcal{A}, D). \tag{21}$$

Proof. Suppose c_u is the medoid of group A_u, and define $s_i = D_{i,c_u}$ for $i \in A_u$ and $\bar{s}_u = a_u^{-1} \sum_{i \in A_u} s_i$. Then $\sum_i s_i = F_1(\mathcal{A}, D)$ and $\sum_u a_u \bar{s}_u = \sum_u \sum_{i \in A_u} s_i = \sum_i s_i = F_1(\mathcal{A}, D)$.

Let S^* be the set of k centers in an optimal k-median solution for D. For any $i \in A_u$, we have $D(c_u, S^*) \le D(c_u, i) + D(i, S^*)$, by triangle inequality. Sum the inequality over all $i \in A_u$ for all u, we have

$$\sum_u a_u D(c_u, S^*) \le \sum_u \sum_{i \in A_u} s_i + \sum_u \sum_{i \in A_u} D(i, S^*) = F_1(\mathcal{A}, D) + \mathrm{OPT}_k(D).$$

Define $r_u = D(c_u, S^*) + \bar{s}_u$. Now we have

$$\sum_u a_u r_u = \sum_u a_u D(c_u, S^*) + \sum_u a_u \bar{s}_u \le \mathrm{OPT}_k(D) + 2F_1(\mathcal{A}, D). \tag{22}$$

Assign each c_u to its nearest center in S^*. Two groups A_u, A_v assigned to the same center $s \in S^*$ satisfy $B_{uv} \le r_u + r_v$. To see why, multiplied both sides by $a_u a_v$ and with triangle inequality,

$$\sum_{i \in A_u} \sum_{j \in A_v} D_{ij} \le \sum_{i \in A_u} \sum_{j \in A_v} (s_i + D(c_u, s) + D(c_v, s) + s_j) = a_u a_v (r_u + r_v).$$

Now we partition the groups in $\mathcal{A}$ by the center of S^* their medoid is assigned to. This way we form a partition of the partition $\mathcal{A}$: $\mathcal{P} = \{P_1, P_2, \cdots, P_L\}$, $P_i \subseteq \mathcal{A}$. For all the groups that are assigned to the same center in S^*, we draw a group center A_u with probability proportional to its weight a_u. Define m_j as the total weight of the groups in P_j. The resulted weighted cost becomes

$$\sum_j \sum_{A_u \in P_j} \frac{a_u}{m_j} \sum_{A_v \in P_j} a_v B_{uv} \le \sum_j \sum_{A_u \in P_j} \frac{a_u}{m_j} \sum_{A_v \in P_j} a_v (r_u + r_v) \tag{23}$$

$$= \sum_j \sum_{A_u \in P_j} \frac{a_u}{m_j} \cdot m_j r_u + \sum_j \sum_{A_u \in P_j} \frac{a_u}{m_j} \sum_{A_v \in P_j} a_v r_v \tag{24}$$

$$= 2 \sum_u a_u r_u \tag{25}$$

Thus, the optimal k-median with metric B has cost at most this value. Combined with Equation (22), we have the claim Equation (21). □

Given a partition $\mathcal{A} = \{A_1, \ldots, A_N\}$ of $[m]$ and a partition $\mathcal{P}$ of $\mathcal{A}$, let $\mathcal{P}^\uparrow$ be the partition of the original points obtained by taking unions. Essentially, $\mathcal{P}^\uparrow = \{P_1^\uparrow, P_2^\uparrow, \cdots, P_L^\uparrow\}$, where $P_i^\uparrow \subseteq [m]$, $i \in P_i^\uparrow$ if and only if $i \in A_j$ and $A_j \in P_i$. $\mathcal{P}^\uparrow$ is called the *lifting* of $\mathcal{P}$. The following lemma says that the cost of the lifting $\mathcal{P}^\uparrow$ is bounded by the cost of the partition $\mathcal{A}$ and the cost of partition $\mathcal{P}$ operating on the metric B.

Lemma G.3 (Lifting). *For a partition $\mathcal{P}$ of $\mathcal{A}$,*

$$F_1(\mathcal{P}^\uparrow, D) \le 4F_1(\mathcal{A}, D) + 2G(\mathcal{P}, B, a). \tag{26}$$

Proof. For $\mathcal{A}$, define $t_i = s_i + \bar{s}_u$ for $i \in A_u$, where s_i and $\bar{s}_u$ are defined in the same way as in the proof of Lemma G.2. Again, $\sum_i t_i = 2F_1(\mathcal{A}, D)$.

By triangle inequality, we have, including when $u = v$, for $i \in A_u,\ j \in A_v$, $D(c_u, c_v) \le s_i + s_j + D_{ij}$. Sum this inequality for all $i \in A_u,\ j \in A_v$ and take average, we have $D(c_u, c_v) \le \bar{s}_u + \bar{s}_v + B_{uv}$. Now apply triangle inequality again,

$$D_{ij} \le s_i + D(c_u, c_v) + s_j \le s_i + \bar{s}_u + \bar{s}_v + B_{uv} + s_j = t_i + B_{uv} + t_j.$$

For a collection J of groups, define $m_J = \sum_{u \in J} a_u$. The expected cost with a uniformly random chosen point in the union of the groups in J as the medoid upper-bounds the cost of the best medoid. Thus,

$$F_1(P^{\uparrow}, D) = \sum_{J \in \mathcal{P}} \sum_{A_u \in J} \sum_{i \in A_u} \frac{1}{m_J} \cdot \sum_{A_v \in J} \sum_{j \in A_v} D_{ij} \tag{27}$$

$$\le 2 \sum_i t_i + \sum_{J \in \mathcal{P}} \frac{1}{m_J} \sum_{A_u, A_v \in J} a_u a_v B_{uv}. \tag{28}$$

For a medoid of J in the metric B, the triangle inequality bounds the second term in the above formula by $2G(\mathcal{P}, B, a)$. This proves (26). □

G.4 Private average distances with unequal group sizes

Suppose first that a partitioning $\mathcal{A}$ is fixed, and let the privacy budget for this release be $(\varepsilon_0, \delta_0)$. Let

$$\kappa = \frac{\sqrt{2\log(2/\delta_0)}}{\varepsilon_0}.$$

For $u < v$, we add independent Gaussian random noise to B_{uv} and release the perturbed value

$$Y_{uv} = B_{uv} + Z_{uv}, \qquad Z_{uv} \sim N\left(0,\, \kappa^2 \rho^2 (N-1)(a_u^{-1} + a_v^{-1})^2\right). \tag{29}$$

We set $Y_{vu} = Y_{uv}$ and set the diagonal Y_{ii} to zero. For a chosen failure probability β_0, set

$$\tau = \kappa \rho \sqrt{N-1}\, \sqrt{2\log(2N^2/\beta_0)}.$$

Construct nonnegative edge weights

$$W_{uv} = \max\{0, Y_{uv} + \tau(a_u^{-1} + a_v^{-1})\} \quad (u \ne v),$$

and let $\widehat{B}$ be their all-pairs shortest-path metric using weight W. Run the weighted local-search algorithm of Lemma G.1 on $\widehat{B}$ with weights a. We will show that the output clustering satisfies $(\varepsilon_0, \delta_0)$-DP and we can bound the cost of the obtained clustering.

Lemma G.4 (Privacy and utility of the weighted release)**.** *For any fixed partition $\mathcal{A}$, the preceding release and clustering procedure is $(\varepsilon_0, \delta_0)$-private under the original row adjacency. With probability at least $1 - \beta_0$,*

$$B_{uv} \le \widehat{B}_{uv} \le B_{uv} + 2\tau(a_u^{-1} + a_v^{-1}) \qquad (u \ne v), \tag{30}$$

and the output partition $\widehat{\mathcal{P}}$ by the weighted local-search algorithm of Lemma G.1 on $\widehat{B}$ with weights a satisfies

$$G(\widehat{\mathcal{P}}, B, a) \le \frac{20}{3}\left(2\,\mathrm{OPT}_k(B, a) + 4\tau \sum_{u=1}^{N} a_u\right). \tag{31}$$

The procedure is polynomial time.

Proof. If the changed index lies in A_u, only the row and column u of B change, and each changed entry differs by at most ρ/a_u. A query of the upper-triangular part of B with

$$\frac{B_{uv}}{\rho\sqrt{N-1}(a_u^{-1} + a_v^{-1})}$$

has squared Euclidean sensitivity at most

$$\sum_{v \neq u} \frac{a_u^{-2}}{(N-1)(a_u^{-1} + a_v^{-1})^2} \leq 1.$$

Thus (29) is a Gaussian mechanism. Every subsequent step is post-processing.

A union bound over the standardized Gaussian noises gives $|Z_{uv}| \leq \tau(a_u^{-1} + a_v^{-1})$ simultaneously, with probability at least $1 - \beta_0$. Conditioned on this, we have $B_{uv} \leq W_{uv} \leq B_{uv} + 2\tau(a_u^{-1} + a_v^{-1})$. Every path has W-length at least its B-length, and hence at least the direct B-distance. The direct edge gives the upper bound in (30).

To prove (31), we take an optimal weighted k-partition in B, which has cost $\mathrm{OPT}_k(B, a)$. In each cluster J of $\widehat{\mathcal{P}}$, choose a replacement center V_J with probability a_v / M_J, where $M_J = \sum_{u \in J} a_u$.

$$\mathbb{E}[\sum_J \sum_{u \in J} a_u \widehat{B}_{u,V_J}] \leq 2\,\mathrm{OPT}_k(B, a) + 4\tau \sum_u a_u.$$

For the second term, each of the two random variables a_u^{-1} and $a_{V_J}^{-1}$ has the first moment, under the weighted pair measure, summing to $\sum_u a_u$. The first term follows by going through the original medoid of each J. Consequently, some replacement-center set attains the displayed bound. Lemma G.1 loses at most the root factor $20/3$. Its assignments in $\widehat{B}$ have at least their true cost in B, by the lower bound in (30). This proves the result.

The expectation in this argument is only for exhibiting a good comparison center set. The actual algorithm is the deterministic local-search procedure applied to the released metric. □

Remark 2 (Adaptive preliminary groups)**.** In the recursive algorithm, $\mathcal{A}$ is itself a private output. For every fixed possible value of $\mathcal{A}$, the preceding privacy proof holds for every adjacent pair D, D' using that same partition on both inputs. Adaptive composition therefore applies. The groups are not treated as public before they are privatized. Their weights are their publicly visible sizes once the private partition has been output.

G.5 Analysis of Utility and Running Time

Theorem 8. *The recursive clustering algorithm in Appendix G.1 is (ε, δ)-private with running time $O\left((t+1) \cdot \left(n^3 + k^2 n^2 \log n\right)\right)$. Further, with probability at least $1 - \beta$, the output satisfies*

$$F_1(\widehat{P}, D) \leq C^{\,t+1} \left(\mathrm{OPT}_k(D) + L\left(n^{a_t} k^{3/2 - a_t}\right)\right), \tag{32}$$

where C is a constant $C \geq 1$,

$$L = \frac{\rho(t+1)}{\varepsilon} \sqrt{\log\left(\frac{2(t+1)}{\delta}\right) \log\left(\frac{2n^3(t+1)}{\beta}\right)},\ a_t = \frac{1}{1 - \left(\frac{1}{3}\right)^{t+1}}. \tag{33}$$

Proof. The recursion tree's original index sets and block boundaries are public and depend only on n, k, t. At each fixed tree depth, these original index sets are disjoint. A row move changes the induced submatrix of at most one such set. Conditioned on all outputs at deeper levels, each node release is $(\varepsilon_0, \delta_0)$-private by Lemma G.4. The privacy guarantee of (ε, δ) then comes from adaptive sequential composition applied to at most $t + 1$ levels from leaves to root.

At a node on m original indices and $N \leq m$ groups, computing all group averages costs $O(m^2)$ arithmetic operations, and all-pairs shortest paths costs $O(N^3)$. By Lemma G.1 the running time of the local search algorithm at one node is $O\left(m^2 + N^3 + k^2 N^2 \log m\right)$. Fixing a depth level, the original node sets are disjoint. Their sizes m_j satisfy $\sum_j m_j \leq n$, and hence $\sum_j m_j^2 \leq n^2$ and $\sum_j m_j^3 \leq n^3$. Summing the running time for all nodes over at most $t + 1$ depths proves the stated running time.

There are at most $n(t+1)$ releases. Conditional on all the previously computed private groups, each confidence event in Lemma G.4 fails with probability at most β_0. A union bound gives a probability

of at least $1-\beta$ that all these events hold. Work deterministically on that event. At every node, one can check that $\tau \le 2L\sqrt{N}$. Consequently, the root noise term in the weighted release is at most $8\alpha LN^{3/2}$, where $\alpha = 20/3$.

In the following, we will prove by induction that at the remaining depth s on m original points, the root cost is at most $K_s\left(\mathrm{OPT}_k(D_{S,S}) + Lm^{a_s}k^{3/2-a_s}\right)$ for a constant K_s specified below. If $m \le k$, the output cost is zero. At $s = 0$ and $m > k$, singleton groups have internal partition cost $E = 0$, $N = m$, and Lemma G.4 proves the claim with $K_0 = 2\alpha$.

Now let $s \ge 1$, and take $a = a_{s-1}$. For public blocks S_j of sizes $m_j \le b$, let their optimal costs use $\min(k, m_j)$ centers. Then

$$\sum_j \mathrm{OPT}_{\min(k,m_j)}(D_{S_j,S_j}) \le 2\,\mathrm{OPT}_k(D_{S,S}). \tag{34}$$

To see this, restrict an optimal global center assignment to a block. For each nonempty intersection with a global cluster, choose a uniformly random member of that intersection as an available medoid. With the triangle inequality, one can bound the expected cost by twice its original assigned cost. Thus some choice attains that bound. number of centers in the block.

Let E be the sum of the true medoid costs of the child outputs. Using induction hypothesis and (34) give

$$E \le 2K_{s-1}\,\mathrm{OPT}_k(D_{S,S}) + K_{s-1}L(mb^{a-1}k^{q-a}).$$

Blocks of size at most k have zero output cost and can be omitted from this bound. The total number of child groups satisfies $N \le k\lceil m/b\rceil \le 2mk/b$.

If $N > k$, applying the compression and weighted-release lemmas and then lifting yields

$$F_1(\widehat{P}, D_{S,S}) \le 8\alpha\,\mathrm{OPT}_k(D_{S,S}) + (4+16\alpha)E + 16\alpha LN^q.$$

If $N = k$, the output cost equals E, so this upper bound remains valid without making a release.

Write $b_* = k(m/k)^{\theta_s}$. Since $b_* \le b \le 2b_*$, the two noise expressions are bounded, respectively, by

$$mb^{a-1}k^{q-a} \le 2^{a-1}m^{a_s}k^{q-a_s}, \qquad N^q \le 2^q m^{a_s}k^{q-a_s}.$$

These exponents agree because

$$a_s = \frac{\frac{3}{2}a}{a+1/2} = 1+\theta_s(a-1) = \frac{3}{2}(1-\theta_s). \tag{35}$$

The closed form (14) solves this recurrence.

For explicit constants put $X = 4+16\alpha$ and take

$$K_s = \left(2+\sqrt{2}\right)XK_{s-1} + 8\alpha + 16\alpha 2^{3/2}.$$

This dominates both the optimum and noise coefficients above, since $a \le q$. For example, it is enough to choose

$$C = \max\left\{8\alpha,\ \left(2+\sqrt{2}\right)X + 8\alpha + 16\alpha 2^{3/2},\ 2\right\}.$$

Then $K_s \le C^{s+1}$ by induction. Taking $s = t$ proves (32). □

Corollary G.5 (Fixed approximation factor and almost linear error)**.** *For every fixed $z \ge 1$ and every fixed $0 < \eta < 1/2$, there is an (ε,δ)-private polynomial-time algorithm returning exactly k clusters such that, with probability at least $1-\beta$,*

$$F_1(\widehat{\mathcal{P}}, D) \le C_\eta\,\mathrm{OPT}_k(D) + \widetilde{O}_\eta\left((\rho/\varepsilon)n^{1+\eta}k^{1/2-\eta}\right).$$

Here the tilde hides logarithmic factors in $n, 1/\delta, 1/\beta$. The constant C_η is independent of $n, k, \rho, \varepsilon, \delta, \beta$, but can grow as η decreases.

Proof. Choose fixed t so that $(1/3)^{t+1} \le \eta/(1+\eta)$. Then $a_t \le 1+\eta$, and, because $n \ge k$, $n^{a_t}k^{3/2-a_t} \le n^{1+\eta}k^{1/2-\eta}$. Apply Theorem 8 we have the claimed result. □

Depth, utility, and running time. Only fixed depths are used for constant approximation. For k-means problem, depths $t = 0, 1, 2, 3$ give additive powers

$$n^2, \qquad n^{4/3}k^{2/3}, \qquad n^{8/7}k^{6/7}, \qquad n^{16/15}k^{14/15},$$

each multiplied by $\widetilde{O}_t(\rho^2/\varepsilon^2)$. Higher depth increases the approximation constant and the number of private levels, but decreases the additive error. The runtime bound gains a factor $t+1$; its polynomial exponent does not grow. For fixed η, using a depth of $O(1 + \log(1 + 1/\eta))$ suffices.

H DP HIERARCHICAL CLUSTERING

We investigate the application of DP metric release to hierarchical clustering (HC) using our algorithmic framework. Hierarchical clustering aims to partition the dataset by a dendrogram with increasing levels of granularity. We study two types of hierarchical clustering: *hierarchical agglomerative clustering (HAC)* and *HC with objective functions*. In particular, we obtain bounds for:

- Differentially private HAC with a per-merge error for single linkage, average linkage, and complete linkage functions.
- Differentially private HAC with a merge height error for single linkage function.
- Differentially private HC with a dissimilarity cost function proposed by (Cohen-Addad et al., 2019).

We will discuss these results in order for the rest of the section.

H.1 LINKAGE-BASED DIFFERENTIALLY PRIVATE HAC: PER-MERGE ERROR

We start with HAC objectives with a per-merge error notion. HAC is a bottom-up meta-algorithm such that at each round, we recursively merge clusters C_i and C_j with the minimum *linkage function criteria*. Depending on the actual linkage function, the criteria can be categorized as follows.

Definition H.1 (Linkage functions)**.** Let C_i and C_j be two clusters and D be a metric distance matrix. The linkage function between them, $\lambda_D(C_i, C_j)$, is defined as follows:

- **Single Linkage:** $\lambda_D(C_i, C_j) = \min_{a\in C_i, b\in C_j} D(a, b)$.
- **Complete Linkage:** $\lambda_D(C_i, C_j) = \max_{a\in C_i, b\in C_j} D(a, b)$.
- **Average Linkage:** $\lambda_D(C_i, C_j) = \frac{1}{|C_i||C_j|} \sum_{a\in C_i} \sum_{b\in C_j} D(a, b)$.

The error of a merge. We use the following error notion to measure the quality of an HAC merge in this section. Let $\mathcal{P}_t$ be the partition immediately before the t-th merge, and let $A_t, B_t \in \mathcal{P}_t$ be the two clusters merged by an algorithm. We define the best available true linkage and the additive error by

$$\mathrm{err}_t(D) = \lambda_D(A_t, B_t) - \min_{\substack{A,B\in\mathcal{P}_t \\ A\neq B}} \lambda_D(A, B). \tag{36}$$

In other words, the error measures the additive gap between the distance created by the algorithm (in terms of the linkage criteria) and the best possible linkage-based distance. In the rest of this section, we write

$$g_t(D) \min_{\substack{A,B\in\mathcal{P}_t \\ A\neq B}} \lambda_D(A, B)$$

as a short-hand notation.

Our error notion draws analogy with HAC with linkage function studied by previous work in the literature. The most popular notion is to compare the *multiplicative* factor between $\lambda_D(A_t, B_t)$ and $g_t(D)$ (Lavastida et al., 2021; Dhulipala et al., 2022). Here, we use the additive version of this error notion.

The algorithm. For any $n \geq 2$, $\rho > 0$, $0 < \varepsilon \leq 1$, and $0 < \delta < 1/2$, the algorithm for the linkage-based clustering is as follows.

Algorithm 3. A DP algorithm for HAC by post-processing the DP metric

1. Run Algorithm 1 to obtain $Y = \texttt{DP-Release}(D, \texttt{REP} = \texttt{false})$.
2. Let $W(i,j) = \max\{0, Y(i,j)\}$ for $i \neq j$ and $W(i,i) = 0$.
3. Initialize $\mathcal{P}_1 = \{\{1\}, \ldots, \{n\}\}$.
4. For $t = 1, \ldots, n-1$, choose
$$(A_t, B_t) \in \operatorname*{arg\,min}_{\substack{A,B \in \mathcal{P}_t \\ A \neq B}} \lambda_W(A, B),$$
using a fixed rule to break ties. Record $\widehat{\ell}_t = \lambda_W(A_t, B_t)$ and set
$$\mathcal{P}_{t+1} = (\mathcal{P}_t \setminus \{A_t, B_t\}) \cup \{A_t \cup B_t\}.$$
5. Return the sequence of merges and their recorded linkage values.

The privacy and utility of the above algorithm on HAC objectives are straightforward to prove. To make the proofs modular, we first give the following lemma for the error bound between $\lambda_W(A, B)$ and $\lambda_D(A, B)$ for two fixed clusters A and B.

Lemma H.1. *For any real symmetric matrices D, W with $\|W - D\|_\infty \leq a$, each of the three linkage functions satisfies*
$$|\lambda_W(A, B) - \lambda_D(A, B)| \leq a$$
simultaneously for every disjoint nonempty $A, B \subseteq [n]$.

Proof. For single and complete linkage, taking the minimum or maximum of entries that each change by at most a changes the result by at most a. For average linkage,
$$\begin{aligned} |\lambda_W^{\mathrm{AL}}(A, B) - \lambda_D^{\mathrm{AL}}(A, B)| &\leq \frac{1}{|A||B|} \sum_{i \in A} \sum_{j \in B} |W(i,j) - D(i,j)| \\ &\leq a. \end{aligned}$$
The statements are deterministic and simultaneous over all A, B. □

Theorem 9 (Private per-merge guarantees)**.** *For each of single, complete, and average linkage functions, the algorithm in this section is (ε, δ)-private under ρ-adjacency. For every $\beta \in (0, 1)$, with probability at least $1 - \beta$ all merges simultaneously satisfy*
$$\mathrm{err}_t(D) = O\left(\frac{\rho}{\varepsilon}\sqrt{n \log(2/\delta) \log(n/\beta)}\right).$$
Moreover, we have
$$\mathbb{E}\left[\max_{1 \leq t \leq n-1} \mathrm{err}_t(D)\right] = O\left(\frac{\rho}{\varepsilon}\sqrt{n \log(2/\delta) \log(en)}\right).$$

Proof. The privacy guarantees simply follow from Theorem 1 since we only conduct post-processing on the distance matrix. As such, we focus on the error bound proof.

Fix any noise realization with $a = \|W - D\|_\infty$, since $D(i,j) \geq 0$, clipping negative entries cannot increase the entry-wise error, i.e., $a \leq \|Z\|_\infty$. At round t, let $A_t^\star, B_t^\star \in \mathcal{P}_t$ attain the minimum linkage criteria $g_t(D)$. By Lemma H.1 and the choice of A_t, B_t,
$$\begin{aligned} \lambda_D(A_t, B_t) &\leq \lambda_W(A_t, B_t) + a \\ &\leq \lambda_W(A_t^\star, B_t^\star) + a \\ &\leq \lambda_D(A_t^\star, B_t^\star) + 2a = g_t(D) + 2a. \end{aligned}$$
Although the clusters depend on the noise, the linkage bound holds for all possible clusters simultaneously. Therefore both conclusions hold for every round on this same noise realization. Applying the error bound in Theorem 1 gives us the statement on the additive error bound. □

H.2 SINGLE-LINKAGE CLUSTERING: MERGE-HEIGHT APPROXIMATION

For the single-linkage clustering criteria, we also study a cost notion defined by the *minimum merge threshold*. In the literature, this is also called "merge height" of a pair of vertices.

Definition H.2 (Merge height of vertex pairs)**.** For a symmetric nonnegative matrix W with zero diagonal and a pair of points (i, j), the *merge height* (or single-linkage ultrametric) of (i, j) is defined as

$$u_W(i,j) = \min_{\pi: i \rightsquigarrow j} \max_{\{v,w\} \in \pi} W(v,w), \qquad u_W(i,i) = 0,$$

where the minimum is over simple paths in the complete graph on $[n]$. Equivalently, $u_W(i, j)$ is the smallest threshold at which i, j become connected using edges of weight at most that threshold.

Note that for a pair (i, j), the merge height in a metric space may *not* be attained by $d(i, j)$ since we only account for the *maximum* weight on the path. In other words, although we have $d(i, j) \leq d(i, k) + d(k, j)$, we might have $d(i, k) \leq d(i, j)$.

The algorithm. Since single-linkage clustering is equivalent to MST, we simply run the algorithm in Section 4 with one modification to use $Y^+(i, j) = \max\{0, Y(i, j)\}$ so that we can obtain Y^+ as non-negative. The guarantee of the algorithm is as follows.

Proposition H.2. *The algorithm is (ε, δ)-private and has maximum merge-height error of at most $O(\rho\sqrt{n \log(2/\delta) \log(n)}/\varepsilon)$ conditioning on the high-probability event of Theorem 1.*

Proof. Let $u_{Y^+}(i, j)$ be the merge height of (i, j) in Y^+, and let $a = \|Y^+ - D\|_\infty \leq \|Z\|_\infty$. For every fixed path, its largest edge weight changes by at most a. Taking the minimum over paths preserves this bound, so $|u_{Y^+}(i, j) - u_D(i, j)| \leq a$ for every pair, and $a = O(\rho\sqrt{n \log(2/\delta) \log(n)}/\varepsilon)$ conditioning on the high probability event of Theorem 1. Privacy follows from post-processing. □

This is the standard stability property of single linkage (see, e.g. (Carlsson & Mémoli, 2010)), applied to the private release. Nevertheless, in what follows, we will show that this simple algorithm surprisingly achieves near-tight bound for DP merge height for single-linkage clustering.

Lemma H.3. *For any sufficiently large n, $0 < \varepsilon \leq 1$, and $0 < \delta \leq \varepsilon/\sqrt{n}$, the expected maximum merge-height error of (ε, δ)-DP single-linkage clustering under ρ-neighboring metric is $\Omega(\rho\sqrt{n}/\varepsilon)$.*

Proof. The proof essentially follows the same argument of the lower bound in Theorem 2, and we only give the sketch to avoid unnecessary repetition. Let $d = n - 1$, $N = \lceil\sqrt{d}/\varepsilon\rceil$, and $L = 2N\rho$. For a database $X \in \{0, 1\}^{N \times d}$, let $q_j(X) = \sum_{r=1}^{N} X_{rj}$. Construct a metric D_X on the points $p, s_1, \ldots, s_d$ by

$$D_X(p, s_j) = L - \rho q_j(X), \qquad D_X(s_j, s_\ell) = L \quad (j \neq \ell).$$

The diagonal is zero, and all off-diagonal distances lie in $[L/2, L]$, which implies the metric construction. Changing one database row changes only the row and column of p, by at most ρ. Every non-direct simple path from p to s_j contains an edge with L. Consequently,

$$u_{D_X}(p, s_j) = D_X(p, s_j) = L - \rho q_j(X).$$

Given a private algorithm for MST, the following algorithm answers the marginals using the merge height: construct D_X, run the algorithm, and return

$$\widehat{q}_j = \frac{L - \widehat{u}(p, s_j)}{\rho} \qquad (j \in [d]).$$

It is private under database-row replacement, and its maximum query error is at most the maximum merge-height error divided by ρ. Applying Proposition 3.1 gives us the desired lower bound statement. □

H.3 Differentially private HC with objective functions

We now switch our attention to an *objective function* for HC. Compared to agglomerative clustering, objective functions provide a metric to evaluate the *quality* of clustering. Since we are studying distance-based HC, we will use the following dissimilarity objective studied by (Cohen-Addad et al., 2019).

Definition H.3. Let $\mathcal{T}_n$ be the family of binary trees whose leaves are labeled by V. For a tree $T \in \mathcal{T}_n$ and distinct $i, j \in V$, let $C_T(i,j)$ be the set of leaves induced by their lowest common ancestor, and let $c_T(i,j) = |C_T(i,j)|$ be the number of leaves. The objective function for T on distance matrix D is defined as

$$R_D(T) = \sum_{1 \le i < j \le n} D(i,j) c_T(i,j). \tag{37}$$

We use $\mathrm{OPT}_{\mathrm{HC}}(D) = \max_{T \in \mathcal{T}_n} R_D(T)$ to denote the optimal HC objective.

The objective should be understood as a "score" as opposed to "cost" since we want to maximize, and the intuition here is that the data points with larger distances are split early in the tree to increase the score. Note that in contrast with Dasgupta's cost ((Dasgupta, 2016)), in which we can assume w.log. that the tree is binary, here the we *enforce* the binary restriction. Without it, a star gives $c_T(i,j) = n$ to every pair and maximizes the objective trivially without giving any meaningful structure.

We have the following immediate observation for the set $C_T(i,j)$ for any i and j.

Observation H.4. *$C_T(i,j)$ is the smallest cluster in T containing both points.*

For an algorithm $\mathsf{ALG}(D)$ that outputs a tree T, we measure the additive error between the score induced by the tree computed by the algorithm and the optimal score, i.e.,

$$\mathrm{OPT}_{\mathrm{HC}}(D) - \mathbb{E}_{\mathsf{ALG}}[R_D(\mathsf{ALG}(D))].$$

Furthermore, the privacy guarantees use the same ρ-adjacency by the input distance matrix D.

We will use the following cost characterization for complete graphs.

Lemma H.5 (Dasgupta (Dasgupta, 2016))**.** *A tree over the input of all pairs of points, i.e., a complete graph, induces a score of $\frac{n^3-n}{3}$ for every $T \in \mathcal{T}_n$. In other words, for every T, there is*

$$\sum_{i<j} c_T(i,j) = \frac{n^3 - n}{3}.$$

H.3.1 An upper bound from input perturbation

We now show the error guarantee by post-processing the metric computed by Theorem 1 as in Theorem 10.

Theorem 10. *For every $n \ge 2$, $\rho > 0$, $0 < \varepsilon \le 1$, and $0 < \delta < 1/2$, there is an (ε, δ)-private algorithm under ρ-adjacency that returns $\widehat{T} \in \mathcal{T}_n$ and satisfies*

$$\mathrm{OPT}_{\mathrm{HC}}(D) - \mathbb{E}\left[R_D(\widehat{T})\right] = O\left(\frac{\rho n^3}{\varepsilon}\sqrt{\log(1/\delta)\log(n)}\right).$$

The algorithm for Theorem 10 is as follows.

Algorithm 4. A DP algorithm for dissimilarity HC objective in Definition H.3

1. Run Algorithm 1 to obtain $Y = \texttt{DP-Release}(D, \texttt{REP} = \texttt{false})$.
2. Enumerate the rooted full binary trees $T \in \mathcal{T}_n$ and compute $R_Y(T) = \sum_{i<j} Y(i,j) c_T(i,j)$ for each tree.
3. Return a maximizing tree $\widehat{T}$ (breaking ties arbitrarily).

In other words, the algorithm runs Algorithm 1 to obtain Y without the metric repairing step. The notation R_M is valid for any M that is a symmetric distance matrix, although the metric property might be violated.

The key lemma to establish the upper bound is to prove for any tree, a matrix constructed from random Gaussian noise will *not* induce large score. The statement is given as follows.

Lemma H.6. *For a $n \times n$ Gaussian noise matrix Z constructed by adding independent noise with $\sigma = \Theta\left(\frac{\rho\sqrt{n\log(1/\delta)}}{\varepsilon}\right)$ and any tree T, we have*

$$\mathbb{E}\left[\max_{T\in\mathcal{T}_n} |R_Z(T)|\right] = O\left(\sigma n^{5/2}\sqrt{\log(n)}\right).$$

Proof. For any fixed tree T, the score $R_Z(T)$ is a centered Gaussian, with

$$\begin{aligned}
\mathrm{Var}(R_Z(T)) &= \sigma^2 \sum_{i<j} c_T(i,j)^2 && \text{(independent noise entries)}\\
&\le \sigma^2 n^4. && (c_T(i,j) \le n \text{ and at most } n^2 \text{ pairs})
\end{aligned}$$

There are at most n^{2n} trees in $\mathcal{T}_n$. Let $M = \max_{T\in\mathcal{T}_n} |R_Z(T)|$. For every $s > 0$,

$$\begin{aligned}
\mathbb{E}\left[\exp(sM)\right] &\le \sum_{T\in\mathcal{T}_n} \left(\mathbb{E}[\exp(sR_Z(T))] + \mathbb{E}[\exp(-sR_Z(T))]\right) && \text{(maximum bounded by a sum)}\\
&\le 2n^{2n}\cdot \exp\left(\frac{s^2\sigma^2 n^4}{2}\right). && \text{(Gaussian exponential moments)}
\end{aligned}$$

Consequently, we have

$$\begin{aligned}
\mathbb{E}[M] &\le \frac{\log \mathbb{E}[\exp(sM)]}{s} && \text{(Jensen's inequality)}\\
&\le \frac{\log(2n^{2n})}{s} + \frac{s\sigma^2 n^4}{2}\\
&= O\left(\frac{n\log n}{s} + s\sigma^2 n^4\right).
\end{aligned}$$

To minimize the sum, we balance the terms and choose

$$s = \frac{\sqrt{n\log n}}{\sigma n^2},$$

leading to the desired $O\left(\sigma n^{5/2}\sqrt{\log n}\right)$ bound. □

The next lemma transforms the scored induced on the noise matrix to the error introduced by adding such Z matrices to distance matrix D.

Lemma H.7. *Let D be the input metric and W any symmetric matrix. If $\widehat{T}$ maximizes R_W (among the trees in $\mathcal{T}_n$), then*

$$\mathrm{OPT}_{\mathrm{HC}}(D) - R_D(\widehat{T}) \le 2\max_{T\in\mathcal{T}_n} |R_W(T) - R_D(T)|.$$

Proof. Let $T^\star$ maximize R_D, and let $M = \max_{T\in\mathcal{T}_n} |R_W(T) - R_D(T)|$. We can bound

$$\begin{aligned}
R_D(T^\star) &\le R_W(T^\star) + M && (|R_W(T^\star) - R_D(T^\star)| \le \max_{T\in\mathcal{T}_n} |R_W(T) - R_D(T)|)\\
&\le R_W(\widehat{T}) + M && \text{(optimality on the released matrix)}\\
&\le R_D(\widehat{T}) + 2M. && (|R_W(\widehat{T}) - R_D(\widehat{T})| \le \max_{T\in\mathcal{T}_n} |R_W(T) - R_D(T)|)
\end{aligned}$$

□

Proof of Theorem 10. Privacy follows from Theorem 1 and post-processing. For the additive error, since $R_Y(T) - R_D(T) = R_Z(T)$, applying Lemmas H.6 and H.7 gives

$$\begin{aligned}\mathrm{OPT}_{\mathrm{HC}}(D) - \mathbb{E}[R_D(\widehat{T})] &\le 2\mathbb{E}\left[\max_{T\in\mathcal{T}_n} |R_Z(T)|\right] \\ &= O\left(\sigma n^{5/2}\sqrt{\log(n)}\right) \\ &= O\left(\frac{\rho n^3}{\varepsilon}\sqrt{\log(1/\delta)\log(n)}\right), \text{ (plugging in } \sigma = \Theta\left(\tfrac{\rho\sqrt{n\log(1/\delta)}}{\varepsilon}\right)\text{)}\end{aligned}$$

as desired. □

H.3.2 A LOWER BOUND FOR PURELY ADDITIVE ERROR

In this section, we show that despite the simplicity of our HC algorithm, it achieves a near-optimal additive error due to the existence of the following lower bound.

Theorem 11. *There is an absolute constant $c > 0$ such that, for all sufficiently large n, $\rho > 0$, $0 < \varepsilon \le 1$, and $0 \le \delta \le c\varepsilon$, every (ε, δ)-private algorithm ALG under ρ-adjacency returning a tree in $\mathcal{T}_n$ has a strict metric input D with*

$$\mathrm{OPT}_{\mathrm{HC}}(D) - \mathbb{E}_{\mathsf{ALG}}[R_D(\mathsf{ALG}(D))] = \Omega\left(\frac{\rho n^3}{\varepsilon}\right).$$

The lower bound proof uses a standard argument to control the additive error induced by the DP constraint. At a high level, the hard instance samples a "double planted group" structure in which the distances inside each group are ρ/ε, and the distances for pairs between the groups are $2\rho/\varepsilon$. Here, the constant matters since we will get separations of the costs using $\Omega(\rho n/\varepsilon)$ additive error per pair of points. The hard instance is prescribed as follows.

The hard instance for Theorem 11.

1. Parameter:
$$h = \left\lfloor \frac{1}{\varepsilon} \right\rfloor, \qquad L = h\rho = \Theta\left(\frac{\rho}{\varepsilon}\right).$$
2. Uniformly at random sample vector $\theta \in \{0,1\}^n$, where $\theta_i \in \{0,1\}$ is the planted label of point i.
3. Construct the distance matrix as follows:
$$D_\theta(i,j) = \begin{cases} 0, & i = j, \\ L, & i \ne j \text{ and } \theta_i = \theta_j, \\ 2L, & \theta_i \ne \theta_j. \end{cases}$$

We will argue that the optimal clustering is to first partition the points with different θ_i values before partitioning the points inside each planted group. In what follows, we will use $S_n := \frac{n^3-n}{3}$ to denote the n^3 error for unit-weight score in Lemma H.5. We first verify the hard instance satisfies the metric property and use the group privacy trick to bound the error induced by instances with *neighboring* θ, i.e., differing in one coordiante θ_i.

Lemma H.8. *Every D_θ is a valid metric. Furthermore, if θ, θ' differ in one label, every (ε, δ)-private algorithm satisfies*

$$\Pr\left(\mathsf{ALG}(D_\theta) \in \mathcal{E}\right) \le e \cdot \Pr\left(\mathsf{ALG}(D_{\theta'}) \in \mathcal{E}\right) + O\left(\frac{\delta}{\varepsilon}\right)$$

for every output event $\mathcal{E} \subseteq \mathcal{T}_n$.

Proof. Observe that all distances between distinct points lie in $[L, 2L]$. Hence, the distances are are positive, and every triangle inequality holds because one distance is at most $2L$ while the other two sum to at least $2L$.

Changing label i changes only row and column i, by at most L. For $t = 0, \ldots, h$, define

$$D^{(t)} = \left(1 - \frac{t}{h}\right) D_\theta + \frac{t}{h} D_{\theta'}.$$

Every intermediate matrix still has off-diagonal entries in $[L, 2L]$ and is therefore a strict metric. Consecutive matrices agree outside row and column i and satisfy

$$\|D^{(t+1)} - D^{(t)}\|_\infty = \frac{1}{h}\|D_{\theta'} - D_\theta\|_\infty \leq \frac{L}{h} = \rho.$$

Applying group privacy along these h adjacent steps gives

$$\begin{aligned}
\Pr\left(\mathsf{ALG}(D_\theta) \in \mathcal{E}\right) &\leq \exp\left(h\varepsilon\right) \Pr\left(\mathsf{ALG}(D_{\theta'}) \in \mathcal{E}\right) + \delta \sum_{t=0}^{h-1} e^{t\varepsilon} \\
&\leq e \cdot \Pr\left(\mathsf{ALG}(D_{\theta'}) \in \mathcal{E}\right) + O\left(\frac{\delta}{\varepsilon}\right), \qquad (h\varepsilon \leq 1 \text{ and } h \leq 1/\varepsilon)
\end{aligned}$$

as claimed. □

We now establish the structure and score of the optimal clustering. Intuitively, the point pairs with $2L$ distances can only be maximized if they are partitioned on the top level. To make our life easier, we only need to *lower bound* the optimal score.

Lemma H.9 (Score of the optimal clustering)**.** *For every label vector θ,*

$$\mathrm{OPT}_{\mathrm{HC}}(D_\theta) \geq LS_n + Ln \sum_{i<j} \mathbf{1}\{\theta_i \neq \theta_j\}.$$

Consequently, for θ sampled uniformly at random, we have

$$\mathbb{E}_\theta[\mathrm{OPT}_{\mathrm{HC}}(D_\theta)] \geq LS_n + \frac{Ln}{2} \cdot \binom{n}{2}.$$

Proof. To prove a lower bound, we only need to fix one way of partitioning, which is to first partition the points into $\{i : \theta_i = 0\}$ and $\{i : \theta_i = 1\}$, and then partition the points inside arbitrarily. The induced cost for this partitioning is

$$R_{D_\theta}(T) = L \cdot S_n + L \cdot \sum_{i<j} \mathbf{1}\{\theta_i \neq \theta_j\} c_T(i, j).$$

Since we partition points with different θ labels at the first level, we have

$$R_{D_\theta}(T) = L \cdot S_n + L \cdot \sum_{i<j} \mathbf{1}\{\theta_i \neq \theta_j\} n.$$

Since we take θ uniformly at random, we have

$$\mathbb{E}\left[\mathbf{1}\{\theta_i \neq \theta_j\}\right] = \frac{\binom{n}{2}}{2}.$$

Therefore, we have

$$\begin{aligned}
\mathbb{E}\left[\mathrm{OPT}_{\mathrm{HC}}(D_\theta)\right] &\geq \mathbb{E}\left[R_{D_\theta}(T)\right] \qquad (T \text{ is one of the valid trees}) \\
&= LS_n + \frac{Ln}{2} \cdot \binom{n}{2},
\end{aligned}$$

as desired. □

The next lemma bounds how much of the total coefficient a private tree can assign to pairs with different planted labels. Intuitively, the lemma says that an algorithm with high scores will "leak" the identity of the points, which is what DP asks to protect.

Lemma H.10 (Score of DP clustering)**.** *Let θ have independent uniform labels and $T = \mathsf{ALG}(D_\theta)$, where ALG is (ε, δ)-private. Then*

$$\mathbb{E}_{\theta,\mathsf{ALG}}[R_{D_\theta}(T)] \le LS_n + \frac{e}{1+e} LS_n + O\left(\frac{\delta}{\varepsilon} Ln^3\right).$$

Proof. Fix $i < j$ and all labels except θ_i. Let D_1 denote the input with $\theta_i \neq \theta_j$, and D_0 the input with $\theta_i = \theta_j$. Write

$$a = \mathbb{E}_{\mathsf{ALG}}[c_{\mathsf{ALG}(D_1)}(i,j)], \qquad b = \mathbb{E}_{\mathsf{ALG}}[c_{\mathsf{ALG}(D_0)}(i,j)]$$

as the expected number of leaves induced by i and j. For each integer $r \in [n]$, the event $\{T : c_T(i,j) \ge r\}$ is the same output event under both inputs. Therefore,

$$\begin{aligned} a &= \sum_{r=1}^{n} \Pr\left(c_{\mathsf{ALG}(D_1)}(i,j) \ge r\right) && \text{(integer coefficient)} \\ &\le e \cdot \sum_{r=1}^{n} \Pr\left(c_{\mathsf{ALG}(D_0)}(i,j) \ge r\right) + O\left(\frac{n\delta}{\varepsilon}\right) && \text{(applying Lemma H.8)} \\ &= e \cdot b + O\left(\frac{n\delta}{\varepsilon}\right). \end{aligned}$$

Adding ea to both sides and dividing by $(1+e)$ gives

$$a \le \frac{e}{1+e} \cdot (a+b) + O\left(\frac{n\delta}{\varepsilon}\right).$$

Now choose θ_i uniformly from $\{0,1\}$, keeping θ_{-i} fixed, and let θ_j be the fixed label on coordinate j. With probability $1/2$, the two labels differ, and the algorithm runs on D_1, giving expected coefficient a. With probability $1/2$, the labels agree, and the algorithm runs on D_0, giving expected coefficient b. The returned tree can differ between these cases.

Write $T = \mathsf{ALG}(D_\theta)$ for the tree returned on whichever input is selected. The indicator $\mathbf{1}\{\theta_i \neq \theta_j\}$ equals one in the first case and zero in the second. Therefore,

$$\begin{aligned} \mathbb{E}\left[\mathbf{1}\{\theta_i \neq \theta_j\} c_T(i,j) \,|\, \theta_{-i}\right] &= \frac{1}{2}a + \frac{1}{2} \cdot 0 = \frac{a}{2}, \\ \mathbb{E}[c_T(i,j) \mid \theta_{-i}] &= \frac{1}{2}a + \frac{1}{2}b = \frac{a+b}{2}. \end{aligned}$$

As such, we can substitute the variables a and b with the costs two new random variables as follows.

$$\begin{aligned} &\mathbb{E}\left[\mathbf{1}\{\theta_i \neq \theta_j\} c_T(i,j) \,|\, \theta_{-i}\right] \\ &\qquad \le \frac{e}{1+e} \mathbb{E}[c_T(i,j) \mid \theta_{-i}] + O\left(\frac{n\delta}{\varepsilon}\right). \end{aligned}$$

This inequality holds for any fixed choice of θ_{-i}. We can then average over the choices of θ_{-i} and use the law of total expectation to obtain

$$\begin{aligned} &\mathbb{E}_{\theta_{-i}}\left[\mathbb{E}\left[\mathbf{1}\{\theta_i \neq \theta_j\} c_T(i,j) \,|\, \theta_{-i}\right]\right] \\ &\qquad = \mathbb{E}_{\theta,\mathsf{ALG}}[\mathbf{1}\{\theta_i \neq \theta_j\} c_T(i,j)]. && \text{(law of total expectation)} \end{aligned}$$

The same equality holds without the indicator. Indeed, each complete label vector receives probability $2^{-(n-1)} \cdot (1/2) = 2^{-n}$ under this two-stage choice, exactly as under a uniform choice of θ. Averaging the conditional inequality therefore gives

$$\begin{aligned} &\mathbb{E}_{\theta,\mathsf{ALG}}[\mathbf{1}\{\theta_i \neq \theta_j\} c_T(i,j)] \\ &\qquad \le \frac{e}{1+e} \mathbb{E}_{\theta,\mathsf{ALG}}[c_T(i,j)] + O\left(\frac{n\delta}{\varepsilon}\right). \end{aligned} \tag{38}$$

Recall from Lemma H.5 that every rooted full binary tree on the same n leaves satisfies

$$\sum_{i<j} c_T(i,j) = S_n = \frac{n^3 - n}{3}.$$

This statement holds separately for every possible output tree. Although changing θ or the algorithm's randomness may change individual coefficients, their sum is always the same number S_n. Consequently,

$$\begin{aligned}
\sum_{i<j} \mathbb{E}_{\theta,\mathsf{ALG}}[c_T(i,j)] &= \mathbb{E}_{\theta,\mathsf{ALG}}\left[\sum_{i<j} c_T(i,j)\right] && \text{(linearity of expectation)} \\
&= \mathbb{E}_{\theta,\mathsf{ALG}}[S_n] && \text{(Lemma H.5 holds for every output)} \\
&= S_n. && (S_n \text{ depends only on } n)
\end{aligned}$$

Therefore, we can bound the cost for any (ε, δ)-DP tree as

$$\begin{aligned}
\mathbb{E}_{\theta,\mathsf{ALG}}\left[R_{D_\theta}(T)\right] &= LS_n + L \cdot \mathbb{E}_{\theta,\mathsf{ALG}}\left[\sum_{i<j} \mathbf{1}\{\theta_i \neq \theta_j\} c_T(i,j)\right] \\
&\leq LS_n + L \cdot \left(\frac{e}{1+e}\mathbb{E}_{\theta,\mathsf{ALG}}[c_T(i,j)] + O\left(\frac{n\delta}{\varepsilon}\right)\right) && \text{(Equation (38))} \\
&= LS_n + \frac{e}{1+e} LS_n + O\left(\frac{\delta}{\varepsilon} L n^3\right),
\end{aligned}$$

as desired. □

Proof of Theorem 11. Let $T = \mathsf{ALG}(D_\theta)$, and since θ is chosen uniformly at random, we can apply Lemmas H.9 and H.10 to obtain

$$\begin{aligned}
&\mathbb{E}_{\theta,\mathsf{ALG}}[\mathrm{OPT}_{\mathrm{HC}}(D_\theta) - R_{D_\theta}(T)] \\
&\geq L\left(\frac{n}{2}\binom{n}{2} - \frac{e}{1+e} S_n\right) - O\left(\frac{\delta}{\varepsilon} L n^3\right) \\
&\geq L n^3 \left(\frac{1}{4} - \frac{e}{3(1+e)} - O\left(\frac{1}{n} + \frac{\delta}{\varepsilon}\right)\right). && (S_n = (n^3 - n)/3)
\end{aligned}$$

Since $e < 3$, the constant $1/4 - e/(3(1+e))$ is strictly positive. The subtraction term vanishes with sufficiently large n and small c, which means $\mathbb{E}_{\theta,\mathsf{ALG}}[\mathrm{OPT}_{\mathrm{HC}}(D_\theta) - R_{D_\theta}(T)] = \Omega(Ln^3) = \Omega(\rho n^3/\varepsilon)$, proving the theorem. □

I DP CORESET

In this section, we apply our framework to (ε, δ)-DP coresets. At a high level, a strong coreset is a weighted set of points whose clustering cost approximates the original cost for every center sets (Feldman & Langberg, 2011). This is slightly different from the case in Appendix E: here, we are interested in a set of centers C rather than a partition $\mathcal{P}$. We will use the notation F_z in a slightly different way: for $C \subseteq [n]$, write

$$D(i, C) = \min_{c \in C} D(i, c), \qquad F_z(C, D) = \sum_{i=1}^{n} D^z(i, C).$$

Coreset is usually evaluated on sets of weighted points. For $S \subseteq [n]$, let D_S be a nonnegative matrix with rows indexed by S and columns by $[n]$. Given weights $w_i \geq 0$ for $i \in S$, we can define the weighted clustering cost as

$$F_z(C, D_S, w) = \sum_{i \in S} w_i \min_{c \in C} D_S^z(i, c).$$

Using the above objectives, we can define (k, η, A)-strong coresets as follows.

Definition I.1 ((k, η, A)-strong coresets)**.** For a set of points $[n]$ with metric distances given by D and a fixed $z \geq 1$, a weighted distance-row summary (S, w, D_S) is a (k, η, A)-strong coreset for D if every center set $C \subseteq [n]$ with $|C| = k$ satisfies

$$|F_z(C, D) - F_z(C, D_S, w)| \leq \eta F_z(C, D) + A.$$

When D_S contains the original distance rows and $A = 0$, this is an ordinary weighted subset strong coreset for discrete center-set queries. Our private output instead uses rows of a released metric.

Our main result for (ε, δ)-DP coreset using the framework we provided is as Theorem 12.

Theorem 12. *There exists a (ε, δ)-private algorithm such that given a dataset of $n \geq 2$ points and parameters $z \geq 1$, $1 \leq k \leq n$, $0 < \varepsilon \leq 1$, and $0 < \delta < 1/2$, under ρ-neighboring for $\rho > 0$, with probability at least $1 - \beta$, returns a (k, η, A)-strong coreset with a number of points*

$$|S| = O_z\left(\frac{k}{\eta^2}\left(k \log n + \log \frac{1}{\beta}\right)\right)$$

multiplicative error η, and additive error

$$A = O_z\left(\frac{\rho^z n^{1+z/2}}{\eta^{z-1}\varepsilon^z}\left(\log \frac{2}{\delta} \log \frac{2n}{\beta}\right)^{z/2}\right).$$

The coreset stores $O(n|S|)$ distance entries from the distance matrix D.

For median and means, the additive terms are respectively $\widetilde{O}(\rho n^{3/2}/\varepsilon)$ and $\widetilde{O}(\rho^2 n^2/(\eta\varepsilon^2))$. These are uniform guarantees over candidate center sets, not over all arbitrary partitions with separately optimized medoids. The algorithm is given as follows.

Algorithm 5. An algorithm for (ε, δ)-DP coreset
Parameters: $n \geq 2$, $\rho > 0$, $0 < \varepsilon \leq 1$, $0 < \delta < 1/2$, and $0 < \eta, \beta \leq 1/2$. Let $\xi = \eta/4$ and $\kappa = 2^{z-1}$.

1. Run Algorithm 1 to obtain $\widehat{D} = \texttt{DP-Release}(D, \texttt{REP} = \texttt{True})$, i.e., input perturbation with metric repair.
2. Apply the existing constant-factor metric clustering routine to $\widehat{D}$, obtaining a set B of k centers with
$$F_z(B, \widehat{D}) \leq \alpha_z \, \mathrm{OPT}_{k,z}(\widehat{D}),$$
where α_z depends only on z. Assign each point to the nearest center in B, and each center to itself. Write V_b for the nonempty group assigned to b.
3. Compute a sensitivity score $s(i)$ using Equation (39) with $\widehat{D}$ as its input for each $i \in [n]$.
4. Sampling: Let $Q = \sum_i s(i)$ and $p(i) = s(i)/Q$, and draw t independent indices $I_1, \ldots, I_t$ following the distribution p over $[n]$, where
$$t = \frac{48Q}{\eta^2} \log \frac{4n^k}{\beta}.$$
5. Let S contain the distinct sampled indices and give each $i \in S$ weight
$$w_i = \frac{|\{r : I_r = i\}|}{tp(i)}.$$
6. Return $(S, w, \widehat{D}_S)$, where $\widehat{D}_S(i, j) = \widehat{D}(i, j)$ for $i \in S, j \in [n]$.

When the coreset is queried by an input set of centers C, we evaluate the cost using $F_z(C, \widehat{D}_S, w)$. We also assume w.log. that $F_z(B, \widehat{D})$ is non zero, as otherwise we can simply return the center set B.

I.1 Analysis of the algorithm

At a high level, our algorithm follows the sensitivity sampling argument of (Feldman & Langberg, 2011), and we will argue that the additive error in the metric can be absorbed into the additive error of the coreset. We start with defining the sensitivity score function and establishing its property.

Lemma I.1. *Let D be any (pseudo)metric and let B be k reference centers with $F_z(B, D) \leq \alpha_z \operatorname{OPT}_{k,z}(D)$ for $\alpha_z \cdot \geq 1$. Let V_b be the nearest-center group of b, and for each point $i \in V_b$, define*

$$s(i) = 2^{z-1} \cdot \alpha_z \cdot \frac{D^z(i,b)}{F_z(B,D)} + 2^{2z-2} \cdot \frac{\alpha_z + 1}{|V_b|}. \tag{39}$$

Then, every k-center set C satisfies the following statements

- $D^z(i, C) \leq s(i) F_z(C, D)$ *for* $i \in [n]$*; and*
- $\sum_i s(i) = 2^{z-1}\alpha_z + 2^{2z-2}(\alpha_z + 1)k = O_z(k)$.

Proof. Fix a candidate center set $C \subseteq [n]$ with $|C| = k$. For every point v, let

$$C(v) = \arg\min_{c \in C} D(v, c), \qquad D(v, C) = D(v, C(v)).$$

The set C is fixed throughout, but $C(v)$ may depend on v. For $i \in V_b$, we have

$$\begin{aligned} D^z(i,C) &= D^z(i, C(i)) \\ &\leq D^z(i, C(b)) && \text{(nearest-center optimality)} \\ &\leq 2^{z-1} D^z(i,b) + 2^{z-1} D^z(b, C). && \text{(generalized triangle inequality)} \end{aligned}$$

The last line uses the triangle inequality along $i, b, C(b)$ and $(u+v)^z \leq 2^{z-1}(u^z + v^z)$ for $u, v \geq 0$. Similarly, for every $j \in V_b$,

$$\begin{aligned} D^z(b,C) &= D^z(b, C(b)) \\ &\leq D^z(b, C(j)) && \text{(nearest-center optimality)} \\ &\leq 2^{z-1} \left(D^z(j,b) + D^z(j,C) \right). && \text{(generalized triangle inequality)} \end{aligned}$$

Substituting this bound and averaging over $j \in V_b$ gives

$$\begin{aligned} D^z(i,C) &\leq 2^{z-1} D^z(i,b) + \frac{2^{2z-2}}{|V_b|} \sum_{j \in V_b} \left(D^z(j,b) + D^z(j,C) \right) && \text{(averaging over the group)} \\ &\leq 2^{z-1} D^z(i,b) + \frac{2^{2z-2}}{|V_b|} \left(F_z(B,D) + F_z(C,D) \right). && \text{(nonnegative partial sums)} \end{aligned}$$

Here $D(j,b) = D(j,B)$ for $j \in V_b$, whereas $D(j,C)$ uses j's nearest center in C. Thus, the two sums are bounded by $F_z(B,D)$ and $F_z(C,D)$, respectively.

Since

$$F_z(C,D) \geq \operatorname{OPT}_{k,z}(D) \geq \frac{F_z(B,D)}{\alpha_z} > 0,$$

we can divide by $F_z(C,D)$ to obtain

$$\begin{aligned} \frac{D^z(i,C)}{F_z(C,D)} &\leq \frac{2^{z-1} D^z(i,b)}{F_z(C,D)} + \frac{2^{2z-2}}{|V_b|} \left(\frac{F_z(B,D)}{F_z(C,D)} + 1 \right) \\ &\leq \frac{2^{z-1}\alpha_z D^z(i,b)}{F_z(B,D)} + \frac{2^{2z-2}(\alpha_z+1)}{|V_b|} && \text{(reference approximation guarantee)} \\ &= s(i). && \text{(definition of the score)} \end{aligned}$$

Finally, summing the scores over the points gives

$$\begin{aligned} \sum_{i=1}^n s(i) &= \frac{2^{z-1}\alpha_z}{F_z(B,D)} \sum_{b \in B} \sum_{i \in V_b} D^z(i,b) + 2^{2z-2}(\alpha_z+1) \sum_{b \in B} \sum_{i \in V_b} \frac{1}{|V_b|} \\ &= 2^{z-1}\alpha_z + 2^{2z-2}(\alpha_z+1)k && \text{(reference cost and number of groups)} \\ &= O_z(k), \end{aligned}$$

where the last equality uses that α_z depends only on z. □

The next lemma is standard and follows from (Feldman & Langberg, 2011) for coreset sampling and approximation. We present the lemma and the proof for the sake of completeness.

Lemma I.2. *Suppose $s(i) > 0$ satisfies*

$$D^z(i, C) \le s(i) F_z(C, D) \qquad \text{for all } i \in [n] \text{ and all } |C| = k.$$

Let $Q = \sum_i s(i)$ and $p(i) = s(i)/Q$; for $0 < \eta, \beta \le 1/2$, draw t independent indices $I_1, \dots, I_t$ following the distribution of p, where

$$t \ge \frac{3Q}{\eta^2} \log \frac{2n^k}{\beta}.$$

Let S be the set of distinct sampled indices and let

$$w_i = \frac{|\{r \in [t] : I_r = i\}|}{tp(i)} \qquad (i \in S).$$

With probability at least $1 - \beta$, every k-center set C simultaneously satisfies

$$|F_z(C, D_S, w) - F_z(C, D)| \le \eta F_z(C, D).$$

Proof. Fix C with $F_z(C, D) > 0$. For each sample, define

$$X_r = \frac{D^z(I_r, C)}{p(I_r) F_z(C, D)}, \qquad r \in [t].$$

These variables are independent and satisfy

$$\begin{aligned} \mathbb{E}[X_r] &= \sum_{i=1}^{n} p(i) \frac{D^z(i, C)}{p(i) F_z(C, D)} && \text{(sampling distribution)} \\ &= \frac{\sum_{i=1}^{n} D^z(i, C)}{F_z(C, D)} = 1. && \text{(definition of the clustering cost)} \end{aligned}$$

Furthermore, we have

$$0 \le X_r \le \frac{s(I_r)}{p(I_r)} = Q,$$

where the second inequality holds due to the importance bound condition $D^z(i, C) \le s(i) F_z(C, D)$. The sampled cost can be related to these variables by

$$\begin{aligned} F_z(C, D_S, w) &= \sum_{i \in S} \frac{|\{r \in [t] : I_r = i\}|}{tp(i)} D^z(i, C) && \text{(sampling weights)} \\ &= \frac{1}{t} \sum_{r=1}^{t} \frac{D^z(I_r, C)}{p(I_r)} && \text{(count each sampled occurrence)} \\ &= \frac{F_z(C, D)}{t} \sum_{r=1}^{t} X_r. \end{aligned}$$

In particular, the relative error is controlled by $\left| t^{-1} \sum_r X_r - 1 \right|$. Since $0 \le X_r/Q \le 1$ and $\mathbb{E}[\sum_r X_r/Q] = t/Q$, the multiplicative Chernoff bound gives

$$\begin{aligned} &\Pr\left(|F_z(C, D_S, w) - F_z(C, D)| > \eta F_z(C, D)\right) \\ &\quad = \Pr\left(\left| \frac{1}{t} \sum_{r=1}^{t} X_r - 1 \right| > \eta \right) \\ &\quad \le 2 \exp\left(-\frac{t\eta^2}{3Q}\right) && \text{(Chernoff for bounded independent variables)} \\ &\quad \le \frac{\beta}{n^k}. && \text{(choice of sample count)} \end{aligned}$$

There are $\binom{n}{k} \le n^k$ candidate center sets, so a union bound gives simultaneous success probability at least $1 - \beta$. A query of zero cost is preserved exactly because all its point contributions are zero. □

So far, we have only discussed standard coreset algorithm with slight variants to (Feldman & Langberg, 2011). In the next lemma, we formally establish the bound between the additive error on the metric and the additive error of the coreset. This is the main lemma we used for our DP coreset case.

Lemma I.3. *Let $D, \widehat{D}$ be pseudometrics on $[n]$ with $\|\widehat{D} - D\|_\infty \le a$, and let $0 < \eta \le 1/2$. If $(S, w, \widehat{D}_S)$ is a $(k, \eta/4, 0)$-strong coreset for $\widehat{D}$, then every k-center set C satisfies*

$$\left|F_z(C, \widehat{D}_S, w) - F_z(C, D)\right| \le \eta F_z(C, D) + K_z \eta^{1-z} n a^z,$$

where K_z depends only on z.

Proof. Fix a candidate center set C. For every i and every $c \in C$, by the bound on $\|\widehat{D} - D\|_\infty$, we have

$$D(i,c) - a \le \widehat{D}(i,c) \le D(i,c) + a.$$

We want to claim that

$$D(i,C) - a \le \widehat{D}(i,C) \le D(i,C) + a$$

even though the nearest centers in D and $\widehat{D}$ may not be the same. To see this, for a fixed center set C and point i, let

$$c_i \in \arg\min_{c \in C} D(i,c), \qquad \widehat{c}_i \in \arg\min_{c \in C} \widehat{D}(i,c).$$

We have

$$\begin{aligned}
\widehat{D}(i,C) &= \widehat{D}(i, \widehat{c}_i) \\
&\le \widehat{D}(i, c_i) && \text{(nearest center under the repaired metric)} \\
&\le D(i, c_i) + a = D(i,C) + a. && \text{(entrywise accuracy)}
\end{aligned}$$

Conversely, we also have

$$\begin{aligned}
\widehat{D}(i,C) &= \widehat{D}(i, \widehat{c}_i) \\
&\ge D(i, \widehat{c}_i) - a && \text{(entrywise accuracy)} \\
&\ge D(i, c_i) - a = D(i,C) - a, && \text{(nearest center under the original metric)}
\end{aligned}$$

which gives us the desired bound between $\widehat{D}(i,C)$ and $D(i,C)$.

The reverse triangle inequality in ℓ_z now implies

$$\begin{aligned}
&\left|F_z(C, \widehat{D})^{1/z} - F_z(C, D)^{1/z}\right| \\
&\quad\le \left(\sum_{i=1}^{n} \left|\widehat{D}(i,C) - D(i,C)\right|^z\right)^{1/z} && \text{(reverse triangle inequality)} \\
&\quad\le n^{1/z} a.
\end{aligned}$$

Moving terms around, we have

$$\begin{aligned}
F_z(C, \widehat{D}) &\le \left(F_z(C, D)^{1/z} + n^{1/z} a\right)^z, \\
F_z(C, D) &\le \left(F_z(C, \widehat{D})^{1/z} + n^{1/z} a\right)^z.
\end{aligned}$$

To compare the two costs, for $u, v \ge 0$ and $q > 0$, we use convexity to obtain

$$(u+v)^z \le (1+q)^{z-1} u^z + (1+1/q)^{z-1} v^z.$$

Choose $q = \eta/(8z)$, and we have $(1+q)^{z-1} \le \exp(\eta/8) \le 1 + \eta/4$ and $(1+1/q)^{z-1} \le c_z \eta^{1-z}$, where $c_z = (1+8z)^{z-1}$ depends only on z. Applying this inequality in both directions of the root-cost bound gives

$$\begin{aligned}
F_z(C, \widehat{D}) &\le (1 + \eta/4) F_z(C, D) + c_z \eta^{1-z} n a^z, \\
F_z(C, D) &\le (1 + \eta/4) F_z(C, \widehat{D}) + c_z \eta^{1-z} n a^z.
\end{aligned} \tag{40}$$

Using the first inequality and the coreset guarantee,

$$\begin{aligned} F_z(C, \widehat{D}_S, w) &\le (1+\eta/4) F_z(C, \widehat{D}) && \text{(coreset guarantee)} \\ &\le (1+\eta/4)^2 F_z(C, D) + (1+\eta/4) c_z \eta^{1-z} n a^z && \text{(by Equation (40))} \\ &\le (1+\eta) F_z(C, D) + 2 c_z \eta^{1-z} n a^z. \end{aligned}$$

Similarly, the second inequality gives

$$\begin{aligned} F_z(C, \widehat{D}_S, w) &\ge (1-\eta/4) F_z(C, \widehat{D}) && \text{(coreset guarantee)} \\ &\ge \frac{1-\eta/4}{1+\eta/4} \left(F_z(C, D) - c_z \eta^{1-z} n a^z \right) && \text{(by Equation (40))} \\ &\ge (1-\eta) F_z(C, D) - c_z \eta^{1-z} n a^z. \end{aligned}$$

Here we used $(1+\eta/4)^2 \le 1+\eta$ and $1-\eta \le (1-\eta/4)/(1+\eta/4)$ for $0 < \eta \le 1/2$. Combining the two bounds proves the lemma with $K_z = 2c_z$. □

Proof of Theorem 12. By Theorem 1, the Gaussian release and its ℓ_∞ metric repair give

$$\|\widehat{D} - D\|_\infty \le a, \qquad a = O\left(\frac{\rho}{\varepsilon} \sqrt{n \log \frac{1}{\delta} \log \frac{n}{\beta}} \right)$$

with probability at least $1 - \beta/2$. Condition on any fixed released metric $\widehat{D}$ and its reference clustering B. By Lemma I.1, the total score is $Q = O_z(k)$. Applying Lemma I.2 with accuracy $\eta/4$ and failure probability $\beta/2$ requires

$$\begin{aligned} t &= \left\lceil \frac{3Q}{(\eta/4)^2} \log \frac{2n^k}{\beta/2} \right\rceil \\ &= \left\lceil \frac{48Q}{\eta^2} \log \frac{4n^k}{\beta} \right\rceil. && \text{(parameters used by the algorithm)} \end{aligned}$$

Thus, conditionally on the released metric and reference clustering, the sampling step returns a $(k, \eta/4, 0)$-strong coreset summary for $\widehat{D}$ with probability at least $1 - \beta/2$. This bound holds for every such conditioning, so a union bound gives probability at least $1 - \beta$ that both the metric-accuracy event and the sampling event hold. Independence of these events is not needed.

On their intersection, Lemma I.3 gives additive error

$$\begin{aligned} A &= K_z \eta^{1-z} n a^z \\ &= O_z \left(\frac{\rho^z n^{1+z/2}}{\eta^{z-1} \varepsilon^z} \left(\log \frac{2}{\delta} \log \frac{2n}{\beta} \right)^{z/2} \right). && \text{(substitute the metric error bound)} \end{aligned}$$

The number of distinct sampled points satisfies

$$|S| = t = O_z \left(\frac{k}{\eta^2} \left(k \log n + \log \frac{1}{\beta} \right) \right). \qquad \text{(total score bound)}$$

Finally, for privacy, since all the sampling and point-weighting operations are just post-processing, the (ε, δ)-DP property follows from Theorem 1. □

J More Details on Experiments

J.1 Datasets

Table 2 lists the 12 datasets and their metrics. MNIST (LeCun et al., 1998) and CIFAR-10 (Krizhevsky, 2009) use balanced W_1 on pooled images. ModelNet10 (Wu et al., 2015) uses Hausdorff distance on sampled mesh surfaces, while Characters (Williams, 2006) uses discrete Fréchet distance on resampled pen trajectories. Sushi (Kamishima, 2016), Swiss-Prot (The UniProt Consortium, 2025), Mushroom (UCI Machine Learning Repository, 1981), and MovieLens (Harper

& Konstan, 2015) use Kendall-τ, edit (Levenshtein), Hamming, and Jaccard distances, respectively. For the Email (Yin et al., 2017) and Facebook (McAuley & Leskovec, 2012) networks, we compute shortest paths in the largest connected component before restricting to the sampled vertices.

The two synthetic datasets use unit-edge graph distances. Expander is the 256-vertex 16×16 Margulis–Gabber–Galil graph (Gabber & Galil, 1981). Diamond is the 172-vertex level-four diamond graph, formed by repeatedly replacing each edge with two parallel paths of length two. Both are classical hard instances for Euclidean embedding: the distortion grows as $\Omega(\log n)$ for constant-degree expanders (Linial et al., 1995) and $\Omega(\sqrt{\log n})$ for diamond graphs (Newman & Rabinovich, 2003). Neither is therefore close to a Euclidean metric.

J.2 Metric Repair Implementations

We first explain our implementations of the ℓ_∞ and ℓ_2 solvers for metric nearness. Both seek symmetric, nonnegative distances with zero diagonal that satisfy the triangle inequality; distinct points may have zero distance. Write

$$\mathcal{M}_n = \{X = X^\top :\ X_{ii} = 0,\ X_{ij} \geq 0,\ X_{ij} \leq X_{ik} + X_{kj} \text{ for all } i, j, k\}.$$

All norms act on the unordered pairs $i < j$.

ℓ_∞ repair. The optimization problem is the linear program

$$t^* = \min_{X,t}\{t :\ X \in \mathcal{M}_n,\ t \geq 0,\ -t \leq X_{ij} - Y_{ij} \leq t \quad (i < j)\}.$$

We consider three implementations of this problem.

(i) Regularized triangle fixing (∞-2008; Brickell et al., 2008). Brickell et al. (2008) apply cyclic Bregman projections to a quadratic reformulation of the linear program. Its specialization to ℓ_∞ is

$$\min_{X,t}\ t + \frac{\gamma}{2}\big(\|X - Y\|_2^2 + t^2\big) \quad \text{subject to } X \in \mathcal{M}_n,\ \|X - Y\|_\infty \leq t.$$

The algorithm cycles through triangle, nonnegativity, and interval constraints, retaining dual corrections. Their pseudocode is given for ℓ_1; we implement the analogous ℓ_∞ version. For sufficiently small $\gamma > 0$, its limit is the LP optimum with the smallest squared correction (Brickell et al., 2008, Theorem 3.1). Since this needs a small γ and exact convergence, our stopping rule also certifies the LP radius.

(ii) Shortest-path radius search (∞-SP). For radius t, the lower bounds are $L_{ij}(t) = \max\{0, Y_{ij} - t\}$; the upper bounds are $U_{ij}(t) = Y_{ij} + t$ off the diagonal, with zero diagonal. If $U(t) \geq 0$, let $C(t) = \mathrm{SP}(U(t))$ be its all-pairs shortest-path closure. The greatest-metric property of closure (Brickell et al., 2008, Lemma 4.1) implies

$$\exists X \in \mathcal{M}_n :\ L(t) \leq X \leq U(t) \quad \Longleftrightarrow \quad C(t) \geq L(t).$$

Our search starts at $\max\{0, -\min_{i<j} Y_{ij}\}$ and tightens radius bounds. A violated path P from i to j supplies the lower bound $t \geq (Y_{ij} - \sum_{e \in P} Y_e)/(|P| + 1)$. It returns $C(t)$ only after closing the radius certificate. At $t = t^*$ it returns the componentwise greatest optimal pseudometric, and a radius gap τ certifies $\|C(t) - Y\|_\infty \leq t^* + \tau$ up to roundoff.

(iii) Shortest-path radius with least-squares selection (∞-LS). After finding t^* by SP, we solve the additional QP

$$X_{\mathrm{LS}} = \arg\min_{X \in \mathcal{M}_n} \frac{1}{2}\|X - Y\|_2^2 \quad \text{subject to } L(t^*) \leq X \leq U(t^*).$$

We adapt Project-and-Forget by adding the interval constraints and their dual variables. For a feasible certificate, close the bounded iterate and, if a lower bound is violated, take the smallest convex combination with the SP solution that restores all lower bounds. Both matrices are pseudometrics below U, so the combination remains so. The QP has a unique solution, which is also an ℓ_∞ optimum; we accept the output only after certifying both the radius gap and the QP gap.

Thus SP and LS can return different matrices despite having the same optimal maximum correction. With sufficiently small regularization and exact convergence, (i) and (iii) select the same matrix,

since $t = t^*$ makes the t^2 term constant. LS avoids tuning γ but adds a QP to the radius search. Because the least-squares selection is defined relative to Y, any lower error against the true metric is an empirical finding rather than a guarantee.

ℓ_2 **repair.** Here the strictly convex QP

$$X_2^* = \arg\min_{X \in \mathcal{M}_n} f(X), \qquad f(X) = \frac{1}{2}\sum_{i<j}(X_{ij} - Y_{ij})^2$$

has a unique solution. Brickell et al. (2008, Algorithm 3.1) give a cyclic triangle-fixing solver that converges to this solution. Project-and-Forget (Sonthalia & Gilbert, 2022) instead maintains an active set: find violated constraints, project with dual corrections, and forget constraints whose dual variables become zero. To make the update explicit, vectorize the unordered pairs as x, y. For $a_r^\top x \le b_r$, it uses

$$\lambda_r^+ = \max\left\{0, \lambda_r + \frac{a_r^\top x - b_r}{\|a_r\|_2^2}\right\}, \qquad x \leftarrow x - (\lambda_r^+ - \lambda_r)a_r.$$

Nonnegativity is enforced as a constraint; the target y is not clipped. We compare three versions of Project-and-Forget (PF).

(i) PF with triangle separation and raw output (2-Tri). It finds violated triangle inequalities, including a maximally violated one, applies the active-set updates, and converges to X_2^*. Unlike the cyclic solver of Brickell et al. (2008), it only updates a changing active set of constraints.

(ii) PF with shortest-path cycle separation and raw output (2-Raw). Algorithm 5 of Sonthalia & Gilbert (2022) finds a shortest path P between each pair and, when violated, adds the constraint

$$X_{ij} \le \sum_{e \in P} X_e.$$

On a complete graph, these cycle inequalities and the triangle inequalities define the same metric cone. It converges to the same X_2^*; only the way violated constraints are found changes. As in Sonthalia & Gilbert (2022), it returns the iterate, and we add an objective-gap certificate and a feasibility check. At nonzero stopping tolerances, their returned iterates are only approximately feasible, not exact pseudometrics.

(iii) PF with cycle separation and closure output (2-Cyc). We keep the updates in (ii), but return $C = \mathrm{SP}(X)$, where the iterate is nonnegative. For constraints $Ax \le b$ and dual variables $\lambda \ge 0$, a dual lower bound is

$$q(\lambda) = \lambda^\top(Ay - b) - \frac{1}{2}\|A^\top\lambda\|_2^2.$$

We stop when $G = f(C) - q(\lambda)$ satisfies $G/\max\{1, f(C)\} \le \tau_{\mathrm{gap}}$. The output is feasible and comes with an optimality certificate:

$$C \in \mathcal{M}_n, \qquad 0 \le f(C) - f(X_2^*) \le G, \qquad \|C - X_2^*\|_2 \le \sqrt{2G}.$$

These bounds hold up to floating-point roundoff. The closure is not itself the ℓ_2 projection; the gap bounds its distance to the projection. The raw cycle rule additionally requires the iterate to be close to C; returning C can avoid iterations spent meeting that requirement. All three PF versions target the same unique optimum, so their finite-run differences concern convergence and stopping, unlike the different optimal selections possible for ℓ_∞ repair.

Table 5 compares six implementations on two datasets under different privacy budgets. On both, ∞-LS has much lower error against the original matrix than ∞-SP, but takes substantially longer. The ℓ_2 solvers give similar errors; 2-Tri is faster on MNIST, while 2-Cyc is faster on Characters. These timings come from one noise seed per dataset.

J.3 Metric Release at Other Privacy Budgets

We repeat metric release with $\varepsilon \in \{0.1, 0.5, 1, 1.5, 2\}$, $\delta = 10^{-4}$, and three noise seeds. Table 6 reports mean ℓ_∞ entrywise error for all seven methods on six representative datasets.

Errors fall quickly when ε increases from 0.1 to 0.5. After $\varepsilon = 1$, the repair methods improve more slowly. ∞-SP remains worse than ∞-LS at every budget, although the gap narrows as noise decreases (on Expander, 225 versus 16.8 at 0.1, and 15.6 versus 4.14 at 2). The three repair methods stay close throughout the sweep, and one of them has the lowest mean in every setting.

Table 5: Six implementations. Each block reports ℓ_∞ error, ℓ_2 error, and runtime in seconds. † marks a time-limited output without an optimality certificate.

	MNIST, $\varepsilon = 0.1$			Characters, $\varepsilon = 1.5$		
Solver	ℓ_∞	ℓ_2	Time (s)	ℓ_∞	ℓ_2	Time (s)
∞-2008†	0.568	114.1	300	0.346	36.45	300
∞-SP	7.14	1754	0.030	1.24	266	0.040
∞-LS	0.567	113.5	75.3	0.343	36.06	171
2-Tri	0.568	114.1	1.07	0.346	36.47	28.7
2-Raw	0.568	114.1	7.13	0.346	36.47	123
2-Cyc	0.568	114.1	8.55	0.346	36.45	23.4

Table 6: Mean metric-release ℓ_∞ error over three seeds across privacy budgets. Bold marks the smallest unrounded mean in each setting. A dagger marks a mean containing one feasible ∞-LS output stopped at 300 seconds without an optimality certificate.

Dataset	Method	$\varepsilon = 0.1$	0.5	1	1.5	2
Characters	∞-SP	14.0	3.38	1.91	1.30	0.989
	∞-LS	**1.19**	**0.683**†	**0.481**	**0.383**	**0.328**
	2-Tri	1.19	0.687	0.485	0.387	0.328
	2-Cyc	1.19	0.688	0.485	0.387	0.328
	SP-Closure	19.2	4.85	2.69	1.91	1.51
	MDS	12.5	3.21	2.12	1.73	1.51
	Fréchet	21.4	5.03	2.64	1.80	1.39
CIFAR-10	∞-SP	15.0	3.42	1.79	1.23	0.955
	∞-LS	1.15†	0.490	**0.407**	0.357	0.328
	2-Tri	1.15	0.487	0.408	**0.353**	**0.322**
	2-Cyc	**1.15**	**0.485**	0.410	0.353	0.322
	SP-Closure	20.2	5.00	2.83	2.04	1.61
	MDS	13.0	3.23	1.85	1.46	1.25
	Fréchet	22.3	5.19	2.70	1.82	1.36
Diamond	∞-SP	163	37.8	20.3	14.9	11.8
	∞-LS	**21.2**	**11.7**	**7.69**†	**6.20**	**5.28**
	2-Tri	21.4	12.1	7.92	6.27	5.30
	2-Cyc	21.4	12.0	7.93	6.27	5.30
	SP-Closure	244	62.4	34.7	24.7	19.3
	MDS	134	36.6	23.6	19.1	16.6
	Fréchet	220	51.8	27.9	19.6	15.2
Expander	∞-SP	225	53.3	27.8	19.8	15.6
	∞-LS	**16.8**	**6.34**	**4.83**	**4.32**	4.14
	2-Tri	16.8	6.35	4.85	4.32	**4.13**
	2-Cyc	16.8	6.36	4.86	4.34	4.14
	SP-Closure	299	74.3	41.4	29.4	23.2
	MDS	172	42.4	23.6	17.0	13.9
	Fréchet	301	69.9	36.3	25.0	19.2
MNIST	∞-SP	7.07	1.62	0.857	0.590	0.458
	∞-LS	**0.569**	0.246	0.200	0.181	0.168
	2-Tri	0.569	**0.242**	0.199	**0.178**	0.164
	2-Cyc	0.569	0.242	**0.199**	0.178	**0.164**
	SP-Closure	9.64	2.40	1.33	0.955	0.762
	MDS	6.23	1.54	0.862	0.642	0.545
	Fréchet	10.7	2.51	1.31	0.893	0.678
ModelNet10	∞-SP	38.6	8.72	4.47	3.09	2.32
	∞-LS	**2.91**	1.15†	**0.881**†	**0.787**†	0.749†
	2-Tri	2.92	1.15	0.882	0.792	0.739
	2-Cyc	2.92	**1.15**	0.882	0.791	**0.739**
	SP-Closure	51.9	12.9	7.28	5.27	4.19
	MDS	33.8	8.40	4.75	3.50	2.86
	Fréchet	57.8	13.4	6.96	4.68	3.50

J.4 Private k-Median

We evaluate k-median at $\varepsilon = 1$ and $\delta = 10^{-4}$ on six datasets. We set $k = 10$ for MNIST, CIFAR-10, and ModelNet10, $k = 20$ for Characters, $k = 2$ for Mushroom, and $k = 42$ for Email. Each method

clusters its private matrix, and we evaluate the resulting centers on the original distances. Gaussian + clip clusters the noisy matrix clipped at zero, without repair, and Recursive is the recursive algorithm of Appendix G. The table reports excess cost over the best of 50 nonprivate local-search starts, which is an empirical reference rather than a certified optimum.

Table 7: Private k-median excess cost at $\varepsilon = 1$, $\delta = 10^{-4}$. Entries are means over three seeds; bold marks the lowest mean in each dataset. Time is the median end-to-end time across the 18 dataset–seed runs.

Method	MNIST	CIFAR-10	ModelNet10	Characters	Mushroom	Email	Time (s)
Gaussian + clip	16.14	25.73	86.59	54.16	1880	737	0.00864
∞-SP	16.08	25.45	85.35	52.75	1910	862	0.0197
∞-LS	16.06	24.94	84.99	31.05	1903	868	132
2-Tri	16.33	24.46	86.61	31.00	1905	861	3.57
2-Cyc	16.39	**24.03**	85.53	**27.75**	1870	854	4.72
SP-Closure	16.18	25.38	84.78	50.13	1902	866	0.0164
MDS	**14.95**	24.58	**82.85**	60.87	**1846**	**664**	0.0252
Fréchet	16.55	25.02	86.66	41.22	1916	859	0.00734
Recursive (depth 2)	15.76	25.01	85.22	61.31	1850	787	0.0240

MDS has the lowest mean excess cost on MNIST, ModelNet10, Mushroom, and Email; 2-Cyc leads on CIFAR-10 and Characters. Thus, the method with the smallest entrywise release error need not give the best centers. The depth-two recursive algorithm has a polynomial-time guarantee, but MDS has lower mean cost on all six datasets. Its median runtime is 0.024 seconds: much faster than 2-Cyc (4.72 seconds), though slower than simpler methods such as Gaussian + clip (0.00864 seconds). On these datasets, the added complexity of recursion gives no clear practical gain, despite its theoretical guarantee.