Differentially Private Minimum Spanning Tree and Clustering in Euclidean Graphs

Zongrui Zou
Nanjing University

Alessandro Epasto
Google Research

Rudrajit Das
Google Research

Chenglin Fan
Seoul National University

Abstract

Many graph learning applications involve analyzing geometric graphs (e.g., nearest neighbor graphs over embeddings) built over sensitive data, thus requiring formal privacy protections. In this paper, we study benchmark problems in privately analyzing geometric graphs obtained from high dimensional embeddings. We provide several new results for the differentially private approximation of minimum spanning trees and hierarchical clustering in Euclidean graphs. Our algorithms achieve a near optimal privacy-utility trade-off (up to constants), providing a $(1+\eta)$-multiplicative approximation with $\tilde{O}(\rho/\eta^2)$ additive error per edge of the tree under ρ-dist privacy (a generalization of DP in geometric data where neighboring datasets different in a single point moved by at most ρ distance). Furthermore, we establish a separation between Euclidean and general graphs by proving a lower bound of $\Omega(\rho\sqrt{n})$ additive error per edge of the tree for general graphs under a similar privacy notion, demonstrating that better utility is indeed achievable (allowing also multiplicative approximation) for geometric data. Our algorithm can also be directly applied to widely used clustering algorithm based on MST, incurring only a small loss in the approximation guarantee compared to its non-private counterpart.

1 INTRODUCTION

Graphs play a fundamental role in data analysis and machine learning, providing the essential tools to unlock insights from interconnected data. By modeling relationships between entities such as users in a social network, products in a recommendation system, or proteins in a biological network, all kinds of graph allow us to understand and leverage the complex structure within data. Given the importance and wide-range of applications of graph data processing, ensuring the privacy of sensitive data involved in graph algorithms is of particular importance. One of the main privacy notions that has been proposed and used extensively in graph analysis is *differential privacy (DP) Dwork et al. (2006)*, which has been successfully applied across numerous graph problems. Key applications include generating private approximations of graph cuts Dalirrooyfard et al. (2023); Chandra et al. (2024); Aamand et al. (2025), spectral properties Liu et al. (2024), correlation and hierarchical clustering Cohen-Addad et al. (2022); Imola et al. (2023); Deng et al. (2025), as well as in the release of various numerical graph statistics Imola et al. (2022); Eden et al. (2025); Suppakitpaisarn et al. (2025).

A significant line of work in private graph analysis focuses on approximating the *minimum spanning tree (MST)* Sealfon (2016); Pinot et al. (2018); Pagh and Retschmeier (2024); Hladík and Těmek (2024); Pagh et al. (2025); Dietz and Kerschbaum (2025). This problem is not only of inherent interest but also serves as a benchmark task for evaluating the performance of algorithms that generate private synthetic graphs. Prior work in this area focused on addressing the problem for *general graphs* and it is hence affected by the hardness of providing approximation guarantees on arbitrary graphs. For instance, algorithms designed to address privacy in arbitrary graphs under certain privacy notions (e.g., DP with ℓ_∞-sensitivity), have a cost that scales superlinearly with n (which is unavoidable due to the $\Omega(n^{1.5})$ lower bound in Pinot (2018)), rendering them non-trivial only for graphs with very large average edge weights and thus impractical for general use. A natural question is whether these guarantees can be made more practical by focusing on a restricted but still useful class of graphs. In this work we fo-

Proceedings of the 29th International Conference on Artificial Intelligence and Statistics (AISTATS) 2026, Tangier, Morocco. PMLR: Volume 300.

cus on the class of *geometric graphs* where the nodes are associated with an embedding and the edges are defined by similarity between these embeddings, e.g., those based on Euclidean distance (i.e., defining the edge weight between any two points as the ℓ_2 distance between them) or cosine distance. Indeed, such graphs are a common use case for many graph analysis in practice as modern deep learning applications on graphs often operate on graphs that are derived from computing nearest neighbor relationships on high dimensional embeddings Cai et al. (2018); Papernot and McDaniel (2018); Halcrow et al. (2020); Makarov et al. (2021); Chami et al. (2022). For example, in Halcrow et al. (2020), graphs are constructed by fusing multiple similarity measures from high-dimensional embeddings to create task-specific graphs optimized for homophily, which are then used for semi-supervised learning. Also, cosine similarity can be used to measure the similarity between two binary functions in the embedding space Xu et al. (2017). As these graphs have inherent structure properties it is in theory possible to derive better bounds (compared to those for arbitrary graphs) for important private graph problems over them.

In this paper, we mainly focus on similarity graphs based on the Euclidean distance of high dimensional embeddings. These graphs are implicitly defined by n-point, d-dimensional dataset $X \in \mathbb{R}^{d\times n}$ of embeddings. We develop accurate and private schemes for releasing important combinatorial characterizations on such Euclidean graphs, including minimum spanning trees (MSTs) and hierarchical clustering. We formally define the notion of differential privacy here:

Definition 1. *Fix privacy budget parameters $\varepsilon \geq 0$ and $0 \leq \delta \leq 1$, a randomized algorithm $\mathcal{A}$ is (ε, δ)-*differentially private *((ε, δ)-DP) if, for every possible output o and any two "neighboring" inputs $X \in \mathbb{R}^{n\times d}$ and $X' \in \mathbb{R}^{n\times d}$, we have*

$$\mathbf{Pr}[\mathcal{A}(X) = o] \leq e^{\varepsilon} \cdot \mathbf{Pr}[\mathcal{A}(X) = o] + \delta.$$

The definition of "neighboring" datasets is critical, as it determines which information is considered sensitive and should be protected. Standard DP protects against arbitrary changes to a single data point, which in some applications can be sometimes overly conservative. In certain applications (e.g., location data, sensor readings), it can be sufficient to protect against small perturbations (e.g., a few kilometers in GPS coordinates or slight variations in energy usage Xiao and Xiong (2015)). Therefore, for Euclidean embeddings, we adopt the natural notion of (ε, δ)-DP under ρ-dist privacy in Epasto et al. (2023), which is a generalization of the standard (ε, δ)-DP without the ρ-dist adjacency notion. In the ρ-dist adjacency notion, two datasets X and X' are neighboring if one can be transformed into the other by moving a single data point a Euclidean distance of at most ρ. We note that this is a generalization of the standard adjacency notion since this allows us to flexibly provide utilities essentially only depending on the smaller distance ρ instead of the radius Γ of the sample space. As we will show, this allows us to provide more practical privacy-utility trade-off in real-world applications.

Under this privacy notion, which we will call ρ-dist privacy for brevity, we provide a collection of upper and lower bounds for privately approximating the MSTs. This privacy notion allows us to present an near optimal algorithm for Euclidean graphs and, furthermore, to establish a separation between the utility achievable in Euclidean graphs and general graphs (under a similar privacy notion tailored for general graphs). Moreover, we also apply our algorithm to privatize a practical clustering algorithm used in real-world applications, the Affinity clustering Bateni et al. (2017) algorithm, achieving a near optimal approximation ratio with an extra linear-sized additive error.

1.1 Related Works

The study of differentially private minimum spanning tree was initialed by Sealfon (2016). Generally speaking, previous literature mainly focus on DP with two main notions of adjacency: ℓ_1-sensitivity and ℓ_∞-sensitivity. In ℓ_1-sensitivity, two graphs are neighboring if the sum of the edge weights differs by at most the sensitivity parameter Δ_1, this is also known as weight-level privacy. In the ℓ_∞-sensitivity, two graphs are neighboring if the weight of *every* edge differs by at most a sensitivity parameter Δ_∞. In both privacy notions, the edge set is public.

In Sealfon (2016), the authors showed that by simply using the Laplace mechanism, it is possible to give an $\tilde{O}(n)$ error algorithm for private MST approximation under the ℓ_1-sensitivity. Hladík and Tětek (2024) later showed that $\tilde{O}(n)$ is necessary. Pinot et al. (2018) and Pagh and Retschmeier (2024) gave DP algorithms under the ℓ_∞-sensitivity by privatizing Prim's algorithm, and obtained an approximate-DP algorithm with $\tilde{O}(n^{1.5})$ error. We note that since Prim's algorithm needs to select $n-1$ edges to construct a spanning tree, then the $\tilde{O}(\sqrt{n})$ overhead from advanced composition is not avoidable, and the error is still $\tilde{O}(n^{1.5})$ even for very sparse graphs with bounded maximum degree. This prevents any improvement when considering sparse graphs.

As ρ-dist privacy is a special case of the privacy notion under ℓ_∞-sensitivity (with $\rho = \Delta_\infty$; see Appendix D), prior work on ℓ_∞-privacy is applicable to the privacy

notion used in this work. However, all prior work on ℓ_∞-sensitivity has cost of at least $\Omega(n^{1.5})$ for computing a minimum spanning tree. Moreover, prior work only applied to privatizing one the specific algorithm (Prim's algorithm), rendering it unsuitable for other clustering applications like our work (e.g., the affinity clustering).

1.2 Problem Setting

Although our focus is on geometric graphs, we also include results for general graphs for comparison. We first present our settings for similarity graphs based on Euclidean distances.

ρ-dist privacy in Euclidean graphs. Fix any integers $d, n \in \mathbb{N}_+$. The input of the problem is a d-dimensional, n-instance real-valued dataset $X = \{x_1, x_2, \cdots, x_n\} \in \mathbb{R}^{d\times n}$. Further, for a given edge set $E \subseteq \binom{[n]}{2}$[1], we define the corresponding similarity graph of X with respect to E as a weighted graph $G = ([n], E, w)$. In this graph, each vertex in $[n]$ corresponds to a point in X, and the weight of an edge between vertices x and y is given by the ℓ_2 distance between their corresponding points, that is, $w_{xy} = \|x - y\|_2$ for any $\{x, y\} \in E$. We adopt ρ-dist privacy Epasto et al. (2023):

Definition 2. *(ρ-dist privacy) Let $X, X' \in \mathbb{R}^{n\times d}$ be two d-dimensional datasets with n points. X and X' are neighboring if and only if there exists one integer $i \in [n]$ such that $\|x_i - x'_i\|_2 \le \rho$ and that $x_j = x'_j$ for every $j \in [n]$ and $j \neq i$.*

ρ-node privacy in general graphs. We also explore a variant of ρ-dist privacy that is specifically tailored for general graphs (including other types of similarity graphs on geometric embeddings of datasets), which we refer to as *ρ-node privacy*. In this setting, each input is instead a n-vertex weighted graph $G = ([n], E, w)$. Under ρ-node privacy, two graphs G and G' are neighboring if G' is obtained from G by modifying the edge weights associated with exactly one node, with the change on each edge not exceeding ρ. We write $x \sim y$ if there is an edge between x and y. Formally speaking:

Definition 3. *(ρ-node privacy) Let $G = ([n], E, w)$ and $G' = ([n], E, w')$ be two graphs on n-vertices with the same topology. G and G' are neighboring if and only if there exists a vertex $x \in [n]$ such that $|w_{xy} - w'_{xy}| \le \rho$ for all $y \sim x$ and that $w_{zw} = w'_{zw}$ for every $\{z, w\} \in E$ and $x \notin \{z, w\}$.*

We note that two neighboring graphs share the same edge set. This setting is consistent with prior work on private MST approximation Sealfon (2016); Pinot et al. (2018); Pagh and Retschmeier (2024); Hladík and Tětek (2024); Pagh et al. (2025); Dietz and Kerschbaum (2025). Indeed, outputting a MST reveals a subset of the edge set E, which is impossible when the E itself is private.

In Appendix D, we also illustrate why ρ-node privacy is a practical consideration for the problem that we are studying, instead of standard ℓ_1-sensitivity (i.e., weight-level privacy) or ℓ_∞-sensitivity in previous literature Pagh and Retschmeier (2024); Pagh et al. (2025). Specifically, we consider ρ-node privacy because it encompasses some natural similarity-based privacy notions in geometric data.

1.3 Our Results

We first establish a lower bound for approximating MST under ρ-node privacy (the notion tailored for general graphs). We show that no private algorithm can achieve an expected additive error smaller than $\Omega_\rho(n^{1.5})$, which is matched by prior algorithms Pinot et al. (2018). This shows that for general graphs, ρ-node privacy is no easier than the privacy under ℓ_∞-notion Pagh et al. (2025), and that private analysis must therefore be highly inaccurate.

Theorem 4 (Informal version of Theorem 27)**.** *Fix $0 < \varepsilon < 1$ and $0 < \delta < \Theta(1/\sqrt{n})$, and let M be algorithm for finding a minimum spanning tree that preserves (ε, δ)-DP under ρ-node privacy. Then, there exists a positively weighted graph G such that,*

$$\mathbb{E}_{M(G)\to\widetilde{T}}\left[Cost_G(\widetilde{T})\right] \ge Cost_G(T) + \Omega_\rho(n^{1.5}),$$

where T is the minimum spanning tree in G. Here, Ω_ρ is the asymptotic notation that hides ρ as a constant.

Then, we show that under two relaxations: (1) allowing a small constant multiplicative error and (2) considering Euclidean graph under ρ-dist privacy, we are able to present a private algorithm for approximating Euclidean MSTs that achieves an additive error only linear in the number of points n and, notably, has no dependence on the dimension d. This result demonstrates a separation in the hardness of approximating the MST between general graphs and Euclidean graphs (though under related different privacy notions).

Theorem 5 (Informal version of Theorems 11 and 13)**.** *For any $\varepsilon > 0$, $0 < \delta < 1$, and constant*

[1]Note that without giving E, the data embeddings themselves can be considered as an implicit *complete* similarity graph. However, it is not necessary to explicitly construct this complete similarity graph for each downstream application. For instance, one may efficiently build a sparse k-NN graph from the synthetic data, which scales well even for large n.

$\eta > 0$, there exists an (ε, δ)-ρ-dist DP algorithm that for any Euclidean similarity graph G, outputs with high probability a spanning tree $\widetilde{T}$ such that:

$$Cost_G(\widetilde{T}) \leq (1 + O(\eta))\, Cost_G(T) + \tilde{O}_\delta\left(\frac{n\rho}{\varepsilon \cdot \eta^2}\right).$$

Here, T is one of the minimum spanning trees of G, and $Cost_G(T)$ is the weight of T on G.

The algorithm underpinning Theorem 5 relies on norm preserving projections to reduce data dimensionality. A notable example is the *Johnson-Lindenstrauss Transformation (JLT)*. Indeed, using the JLT to improve the utility of private algorithms is a rather folklore idea Kenthapadi et al. (2012); Nikolov (2023). However, it was never applied in private MST approximation, and our analysis is notably more flexible: it accommodates any norm preserving projection, while the privacy analysis in prior work Kenthapadi et al. (2012) is specific to random Gaussian matrix.[2]

This generality is a key advantage since by decoupling the theoretical guarantees from the specific projection method, we provide a universal and practical framework for private dimensionality reduction. For example, it allows us to select a projection matrix based on computational constraints; for instance, sparse or fast-JLT transformations Dasgupta et al. (2010); Kane and Nelson (2014); Rakhshan and Rabusseau (2020) can offer significant performance benefits over dense Gaussian matrices on large-scale datasets.

More interestingly, we can show that the dependency on the dataset size n in our utility guarantee is optimal, even if allowing a multiplicative factor. This holds even for datasets of moderately high dimension, specifically when $d = O(\log n)$, demonstrating that the upper bound achieved by our simple dimension reduction technique is already tight.

Theorem 6 (Informal version of Theorem 16)**.** *Fix any $\varepsilon = O(1)$ and $\delta \leq \frac{1-e^{-\varepsilon}}{2(1-e^{-2\varepsilon})}$. There exists a Euclidean similarity graph G such that if a mechanism M for finding minimum spanning tree preserves (ε, δ)-ρ-dist privacy, then we have*

$$\mathbb{E}_{M(G)\to\widetilde{T}}[Cost_G(\widetilde{T})] \geq (1+\eta)\, Cost_G(T) + \Omega_\rho\left(\frac{n}{1+\varepsilon}\right)$$

for any $0 \leq \eta \leq O_\varepsilon(1)$. Here, T is the minimum spanning tree in G. Further, G is an Euclidean embedding of $d = O(\log n)$ dimensional data.

Finally, because our algorithm uses a private prepossessing procedure that is applicable to any subsequent non-private MST algorithm, it can be used to privatize many widely used hierarchical clustering algorithms that rely on constructing an MST. In particular, we propose the first private version of the Affinity clustering algorithm Bateni et al. (2017). We analyze the output of our private algorithm's first-round clustering by comparing it to the optimal cluster under the metric from Definition 26.

Theorem 7 (Informal version of Theorem 17)**.** *Fix $\varepsilon > 0$, $0 < \delta < n^{-c}$, and a small constant $\eta > 0$. For any input dataset X with Euclidean similarity graph G, let $\mathcal{C}^*$ be the optimal non-singleton clustering of G. There exists an (ε, δ)-ρ-dist DP hierarchical clustering algorithm which, in its first round, outputs a clustering $\mathcal{C}$ satisfying:*

$$\frac{cost(\mathcal{C}(G))}{2(1+O(\eta))} \leq cost(\mathcal{C}^*(G)) + \tilde{O}_\delta\left(\frac{(n - \#\mathcal{C}(G))}{\varepsilon\eta^2}\right).$$

Here, $cost(\mathcal{C}(G))$ is the cost of clustering $\mathcal{C}$ on graph G (see also Definition 26), and $\#\mathcal{C}(G)$ is the number of clusters in $\mathcal{C}$.

We note that the non-private Affinity clustering algorithm achieves a 2-approximation in its first round. Our private algorithm thus incurs only an additional $1 + O(\eta)$ factor in the approximation ratio, demonstrating that the privacy cost is minimal compared to the inherent approximation factor.

2 PRIVATE MST IN EUCLIDEAN GRAPHS

In section 1.1 we have discussed that previous algorithms on private minimum spanning tree gives $\tilde{O}(n^{1.5})$ purely additive error under ρ-node privacy for constant ρ. In this section, we consider the special but practical case of Euclidean graphs to break this barrier (by also allowing a multiplicative approximation). Before that, we present two warm-up examples. For comparison, in Section 2.1, we first consider general graphs and add noise to their edge weights as a post-processing step. This gives $\tilde{O}(n\sqrt{\Delta})$ purely additive error on finding minimum spanning tree under node privacy, where Δ is the maximum (unweighted) degree. For geometric graphs, we show in Section 2.2 an alternative error bound of $\tilde{O}(n\sqrt{d})$ where d is the dimension of each vertex. This error bound inspires us to use dimension reduction techniques to reduce error. In Section 2.3, we apply a random projection technique to reduce the additive error on finding MST under node-privacy to only $\tilde{O}(n/\eta^2)$ while allowing a multiplicative error at most $1 + O(\eta)$, thus breaking the $\tilde{O}(n^{1.5})$ barrier.

[2]Moreover, the results in Kenthapadi et al. (2012) do not directly yield a multiplicative guarantee on distance preservation.

2.1 Warm up I: A trivial privatization using post-processing over edge weights

As a warm up, in Algorithm 1, we give the simplest approach to privatize an arbitrary MST algorithm by Gaussian mechanism and post-processing. This gives an $\tilde{O}(n\Delta)$ error bound under ρ-node privacy. This shows a separation between ρ-node privacy and ℓ_∞-sensitivity privacy under bounded degree graphs.

Algorithm 1: Post-Processing with Edge-Flipping

Input: A graph $G = ([n], E, w)$ with maximum unweighted degree Δ, privacy budgets (ε, δ) and the sensitivity $\rho \geq 0$.

Output: A tree $T \subseteq E$ as an approximation of the MST in G.

Let $T \leftarrow \varnothing$, $\sigma \leftarrow \frac{\rho\sqrt{2\Delta \log(1.25/\delta)}}{\varepsilon}$ and $r \leftarrow n$;

Adding i.i.d. Gaussian noise from distribution $\mathcal{N}(0, \sigma^2)$ to each edge weight;

Initializing clusters $c_i = \{v_i\}$ for each $i \in [n]$;

while *the number of clusters* $r > 1$ **do**

  for *each cluster* c_i *in* $(c_1, c_2, \cdots, c_r)$ **do**

    Let e be the cheapest edge going out of c_i;

    Add e to T;

  Merge clusters connected by edges added in this iteration;

  Update r to be the new number of clusters after merging;

return T.

Theorem 8. *Algorithm 1 preserves (ε, δ)-DP under ρ-node privacy. Moreover, with high probability, the spanning tree T it outputs has a total weight that exceeds that of the true minimum spanning tree of G by at most $\widetilde{O}\left(\frac{n\rho\sqrt{\Delta}}{\varepsilon}\right)$.*

The proof of Theorem 8 is straightforward and deferred to Appendix C.1.

2.2 Warm up II: Another trivial privatization using post-processing over coordinates

In this section we only consider ρ-dist-privacy in the ℓ_2-Euclidean space. In such privacy notion, for any neighboring dataset X, X', there is only one pair of data points $x \in X$ and $x' \in X'$ such that $\|x - x'\|_2 \leq \rho$. Therefore, there is another obvious way of post-processing, that is to perturb each coordinate of the input dataset by adding independent d-dimensional Gaussian noise of variance $\sim \frac{\rho^2 \log(1/\delta)}{\varepsilon^2}$, instead of adding noise to edge weights as in Algorithm 1. Then, with high probability, for any $\{x, y\} \in E$, the noisy edge weight would be

$$\hat{w}_{xy} = \|x + z - (y + z')\|_2 \leq \|x - y\|_2 + \|z - z'\|_2 \leq w_{xy} + \tilde{O}(\sqrt{d}\rho).$$

Similarly we also have

$$\hat{w}_{xy} \geq w_{xy} - \tilde{O}(\sqrt{d}\rho).$$

By the union bound over $O(n^2)$ edges and the almost identical reasoning in the last section, we directly give the following theorem for **low-dimensional** data:

Theorem 9. *There is an (ε, δ)-ρ-dist DP algorithm that for any d-dimensional, n-instance dataset X, it finds a spanning tree $T \subseteq E$ such that w.h.p, T has a total weight that exceeds that of the true minimum spanning tree of the Euclidean similarity graph of X (in terms of E) by at most $\widetilde{O}\left(\frac{n\rho\sqrt{d}}{\varepsilon}\right)$.*

2.3 Improved utility in Euclidean graph via dimension reduction

In this section, again we consider the Euclidean similarity graph of dataset $X \in \mathbb{R}^{d \times n}$ determined by a fixed edge set $E \subseteq \binom{[n]}{2}$, where edge weights are defined by the ℓ_2 distances between entries in X. We begin by defining the norm preserving projection:

Definition 10. *Fix any $d, r, \in \mathbb{N}_+$ and $0 \leq \eta, \beta \leq 1$. A function P is a (d, r, η, β)-norm preserving projection if $P : \mathbb{R}^d \to \mathbb{R}^r$ and that for any $x \in \mathbb{R}^d$, with probability at least $1 - \beta$,*

$$(1 - \eta)\|x\|_2^2 \leq \|P(x)\|_2^2 \leq (1 + \eta)\|x\|_2^2.$$

We also write $P(X) = [P(x_1), P(x_2), \cdots, P(x_n)]$ for $X \in \mathbb{R}^{d \times n}$. With the definition of such projection, the idea is to first reduce the dimension of dataset X from d to $O(\log n)$, and then use the post-processing scheme in Section 2.2 to generate a private synthetic dataset. Next, we use the given edge set E[3] to generate the similarity graph $\widetilde{G}$ out of this synthetic dataset, where the edge weights are distances between the corresponding embeddings. Finally, we apply the non-private Borůvka's algorithm for finding MST on the similarity graph $\widetilde{G}$ as a post-processing procedure. We first present the algorithm in Algorithm 2:

Theorem 11. *Given a (d, r, η, β)-norm preserving projection P, Algorithm 2 preserves (ε, β)-DP under ρ-dist privacy.*

[3] By default one can consider E as a full graph (i.e., $E = \binom{[n]}{2}$). However, for the sake of efficiency one can also build sparse similarity graphs using non-private LSH Datar et al. (2004) to speed up downstream applications. Therefore our algorithm is by design to handle non-complete graphs. Further, generating a sparse similarity graph using non-private algorithms **does not** compromise privacy since it post-processes a private synthetic dataset $\widetilde{X}$.

Algorithm 2: Approximating MSTs in Euclidean Graph With Norm Preserving Projection

Input: A dataset $X \in \mathbb{R}^{d\times n}$, privacy budgets (ε, δ), the sensitivity $\rho \geq 0$, a desired relative factor $\eta \in (0,1)$ and a (d, r, η, β)-norm preserving projection P.

Output: A tree $T \subseteq \binom{[n]}{2}$.

Let $T \leftarrow \varnothing$ and $r \leftarrow n$;

Let $\hat{X} \leftarrow P(X)$;

for *each column* $\hat{x}_i$ *in* $\hat{X}$ **do**

  Sample $z_i \in \mathbb{R}^r$ whose entries are i.i.d. sampled from $\mathrm{Lap}\left(\frac{\rho\sqrt{r(1+\eta)}}{\varepsilon}\right)$;

  Let $\tilde{x}_i = \hat{x}_i + z_i$;

Let $\widetilde{X} = (\tilde{x}_1, \cdots, \tilde{x}_n)$;

Construct the Euclidean similarity graph $\widetilde{G} = ([n], E, \tilde{w})$ of $\widetilde{X}$;

Initializing clusters $c_i = \{v_i\}$ for each $i \in [n]$;

while *the number of clusters* $r > 1$ **do**

  for *each cluster* c_i *in* $(c_1, c_2, \cdots, c_r)$ **do**

    Let e be the cheapest edge in $\widetilde{G}$ going out of c_i;

    Add e to T;

  Merge clusters connected by edges added in this iteration;

  Update r to be the new number of clusters after merging;

return T.

Note that Algorithm 2 adds Laplace noise to the randomly projected data, instead of the raw input. Therefore, before getting into the proof of Theorem 11, we discuss the privacy guarantee of a mechanism when the input is also random.

Let X, Y be two random variables, a coupling of X, Y is a joint distribution $\mathcal{C} = (X', Y')$ such that its marginal distribution X' and Y' follows the same distribution as X and Y. Under some specific privacy notion, for two datasets $x, y \in \mathcal{R}$, we write $x \sim_k y$ if there exists a series $x = x_0, x_1, \cdots, x_k = y$ such that x_i is the neighbor of x_{i+1} for any $0 \leq i \leq k-1$. We present the following lemma in this section. The analysis of of Lemma 12 follows a relatively standard approach, and we include it in Appendix C.4 for the sake of correctness and completeness.

Lemma 12. *Let $\mathcal{S} : \mathcal{R} \to \mathcal{R}$ be a random mapping and $\mathcal{M} : \mathcal{R} \to \mathcal{Y}$ be an $(\varepsilon, 0)$-differentially private algorithm. Given $k > 0$, suppose that for any $x, y \in \mathcal{R}$ that $x \sim y$, there exists a coupling $\mathcal{C}$ of $\mathcal{S}(x)$ and $\mathcal{S}(y)$ such that with probability at least $1 - \delta'$ over the randomness of $\mathcal{C}$, $\mathcal{S}(x) \sim_k \mathcal{S}(y)$. Then, the mechanism $\mathcal{M} \circ \mathcal{S}$ preserves $(k\varepsilon, \delta')$-differential privacy.*

Theorem 11 can be obtained by considering the norm preserving projections as the random mappings in Lemma 12, and we defer the details in Appendix C.2.

We have shown that the privacy of Algorithm 2 is preserved for any abstract norm preserving projection P. Fix $d, r, 0 < \eta < 1/2$, recall that according to Johnson-Lindenstrauss Lemma (Lemma 25), a $(d, r, \eta, 2e^{-\eta^2 r/8})$-norm preserving projection exists, which is simply the random Gaussian matrix $\frac{1}{\sqrt{r}}\Pi$ where each entry in $\Pi \in \mathbb{R}^{r\times d}$ is i.i.d. sampled from $\mathcal{N}(0,1)$. Define $G = ([n], E, w)$ be the Euclidean similarity graph given input $X \in \mathbb{R}^{d\times n}$ and edge set E. Replacing P with such random matrix in Algorithm 2, we have the following utility guarantee.

Theorem 13. *Fix $\varepsilon > 0$, $0 < \delta < n^{-4}$ and small η. There is a random norm preserving P such that Algorithm 2 is (ε, δ)-ρ-dist DP. Further, given any $X \in \mathbb{R}^{d\times n}$, and a fixed edge set $E \subseteq \binom{[n]}{2}$, with probability at least $1 - 1/\mathrm{poly}(n)$ (over the randomness of P and Laplace noise), Algorithm 2 finds a spanning tree $\widetilde{T}$ of the Euclidean similarity graph $G = ([n], E, w)$ of the input dataset X such that*

$$\mathit{Cost}_G(\widetilde{T}) \leq (1 + O(\eta))\,\mathit{Cost}_G(T) + \tilde{O}\left(\frac{n\rho\log(1/\delta)}{\varepsilon\cdot\eta^2}\right).$$

Here, T is one of the minimum spanning trees of G.

We defer the proof of Theorem 13 in Appendix C.3.

3 LOWER BOUND FOR EUCLIDEAN GRAPHS

In this section, we show that for Euclidean graphs, any differentially private algorithm for the minimum spanning tree (MST) must incur an additive error of $\Omega(n)$ under distance privacy (Definition 2). This lower bound holds even for multiplicative approximations exist and when the edge set E is public. This implies that our algorithm is optimal up to constant factors. The construction of our lower bound depends on the concept of *kissing number* in high dimensional space.

Definition 14. *Let $d \in \mathbb{N}_+$. The kissing number in dimension d, $K(d)$, is the maximum number of non-overlapping unit spheres that can touch a single unit sphere in dimension d.*

We apply the following guarantee on the kissing number in d dimensional space:

Theorem 15. *(Jenssen et al. (2018)) The kissing number in d dimension satisfies that $K(d) \geq \Omega((2/\sqrt{3})^d)$.*

We then present the hardness result for privately approximating the minimum spanning tree on Euclidean

graphs. We note that the hard instance is a Euclidean embedding in $O(\log(n))$-dimension space.

Theorem 16. *(Lower bound for Euclidean graph.) Fix any $\varepsilon > 0$ and $0 \leq \delta \leq \frac{1-e^{-\varepsilon}}{2(1-e^{-2\varepsilon})}$. Suppose the mechanism M for finding minimum spanning trees is an (ε, δ)-ρ-dist DP algorithm. Then, there exists an Euclidean graph G such that*

$$\mathbb{E}_{M(G)\to T}[w(T)] \geq (1+\eta)w(T^*) + \Omega\left(\frac{\rho n}{e^{\varepsilon}}\right).$$

for any $0 \leq \eta \leq \frac{e^{-\varepsilon}}{32(1+e^{-\varepsilon})}$. Further, G is an Euclidean similarity graph of $d = O(\log n)$ dimensional data.

Before diving into the proof of Theorem 16, we first present the construction of the "hard graph", which is also based on a star graph. Let $\alpha > 0$ be some parameter to be determined. We consider a n-vertex Euclidean graph $G = (V, E, w)$ where $V = \{v\} \cup_{i=1}^{k} \{u_a, u_b\}$ such that $1 + 2k = n$. Here, let v be the central vertex, and for each pair (u_a, u_b), we place them on a ray emanating from v (with each pair assigned to a different ray), such that u_a lies at a distance α from v, and the distance from u_b to v is 2α. Furthermore, we position the central vertex at the center of a d-dimensional sphere with radius $\alpha/2$. Each vertex u_a is then placed at the center of a distinct d-dimensional sphere, also of radius $\alpha/2$, such that the spheres corresponding to different u_a are neither intersecting nor tangent to each other. Note that such configuration is feasible when $K(d) > k$. By Theorem 15, it is enough to set $d = O(\log n)$. The detailed proof of Theorem 16 is given in Appendix C.5.

4 APPLICATIONS TO HIERARCHICAL CLUSTERING

Here we apply the dimension reduction framework to private clustering for nodes in the Euclidean space. We first introduce the concept of affinity clustering Bateni et al. (2017), which is an example of hierarchical clustering based on Borůvka's algorithm. We believe similar results can be proved for other algorithms.

Affinity Clustering using Borůvka's algorithm. Clustering algorithms can be designed using approaches that construct a minimum spanning tree. Borůvka's algorithm is a well-known algorithm for finding MSTs, and is especially useful in parallel computation scenarios. In Borůvka's algorithm, each vertex is initiated as a cluster. Then, the algorithm picks the cheapest edge going out of each cluster, and use these edges to merge clusters to form largest clusters. This process continues iteratively until a spanning tree of the graph is found and all vertices are connected. This algorithm naturally defines a hierarchy of vertices, and can be equivalently considered as a procedure for hierarchical clustering. We call this kind of clustering as *affinity clustering*.

For the clustering task in this section, we define G as the *complete* Euclidean similarity graph of the input dataset $X \in \mathbb{R}^{d\times n}$ (i.e., letting $E = \binom{[n]}{2}$ in Algorithm 2). We made this relaxation since computing the nearest neighbor in Borůvka's algorithm can be easily speed up with standard Local Sensitivity Hashing tehcniques Datar et al. (2004).

Note that Algorithm 2 is exactly a private version of Borůvka's algorithm for finding minimum spanning trees. Moreover, since the size of each cluster will be doubled at each iteration of Borůvka's algorithm (see step 8 to 14 in Algorithm 2), Algorithm 2 terminates in at most $O(\log n)$ rounds. To measure the utility of the clustering, based on the utility measure (Definition 26) that was firstly proposed in Bateni et al. (2017), we present the following utility guarantee for Algorithm 2 in the *first round* of affinity clustering. The proof of Theorem 17 can be found in Appendix C.6. Notably, a single round of affinity clustering often yields good solutions for various applications.

Theorem 17. *Fix any input dataset X whose Euclidean embedding is G, let $\mathcal{C}^*$ be the optimal non-singleton clustering of G. The first round of Algorithm 2 finds a clustering $\mathcal{C}$ of G such that*

$$\frac{cost(\mathcal{C}(G))}{2(1+O(\eta))} \leq cost(\mathcal{C}^*(G)) + \tilde{O}\left(\frac{(n - \#\mathcal{C}(G)) \cdot \log(1/\delta)}{\varepsilon\eta^2}\right)$$

Here, $cost(\mathcal{C}(G))$ is the cost (defined in Definition 26) of clustering $\mathcal{C}$ on graph G, and $\#\mathcal{C}(G)$ is the number of clusters in $\mathcal{C}$.

Tightness of the multiplicative factor. To understand how good is the utility, we show that there exists an example G such that the cost of the best non-singleton clustering on G is at most $1/2 + o(1)$ fraction of the cost of the clustering obtained from the first round of *non-private* affinity clustering. Therefore, our $(\frac{1}{2+O(\eta)})$-approximation for private algorithms is nearly optimal for small $\eta > 0$. Here, G is a n-vertex cycle in which n is odd and the weight of each edge starts from 1 and increases by 1 clockwise, until the weight of the n-th edge is n. Then, the clustering from the first round of the affinity clustering is clearly the path with edge weights $1, 2, \cdots, n-1$, and the cost of which is $n(n-1)/2$. However, by deleting non-adjacent edges and join the edge with weight n, we obtain a new non-singleton clustering of cost $(1+n)^2/4$. The ratio between the costs of two clustering is $1/2 + o(1)$, concludes our claim.

5 EMPIRICAL RESULTS

This section presents empirical results of our algorithms on both synthetic graphs and a *publicly-available* real-world graphs extracted from LastFM Asia Social Network dataset Asuncion et al. (2007).Using similarity graphs constructed from Euclidean embedding, we benchmark the performance of our algorithm against standard private MST approximation methods including edge-flipping Sealfon (2016).

5.1 Advantage of dimension reduction in Privately Approximating MST

First, we provide empirical evidence for the improvement afforded by a JL-type transformation over its absence. To this end, we define a baseline algorithm, `trivial`, which adds noise directly to each Euclidean embedding. In contrast, our method (Algorithm 2, denoted `JLT`) first applies a JL-transform to preserve norms before perturbing the data.

We evaluate our improvements on datasets with $n = 5 \times 10^3$ instances across varying dimensions, using privacy parameters $\varepsilon = 1$ and $\delta = 10^{-4}$. Crucially, rather than generating synthetic data independently from an identical distribution, we deliberately construct datasets where distances between datapoints are highly *unconcentrated*. This is because that data from an i.i.d distribution (for instance, high dimensional Gaussian distribution), usually produces spanning trees of highly concentrated sizes, making even a random spanning tree already very close to the real MST, which obscure the advantage of our algorithm.

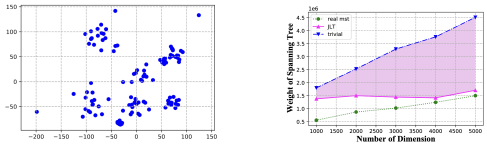

Figure 1: Left: a sample testing dataset in 2D space, with 8 clusters and 4 outliers; Right: private MST approximation on dataset with different dimensions.

Our dataset combines tight clusters (from multivariate Gaussians with random centers and variable spreads) with distant outliers (sampled uniformly from a wide range). This forces the MST to include both very long and very short edges, creating the high variance in the sizes of spanning trees. A sample dataset of $n = 150$ points generated with this strategy is illustrated in Figure 1. With such a distribution of the synthetic data, we test the improvement of `JLT` over `trivial` with dimensions from $d = 10^3$ to $d = 10^5$. The results are also shown in Figure 1. Each datapoint is an average over 50 independent runs. Indeed, as the dimension increases, the advantage of the `JLT` becomes more apparent, as its theoretical guarantee is independent of the dimension (unlike that of `trivial`), a key property that we have established in Theorem 13.

5.2 Private Affinity Clustering

As our primary application, we evaluate our framework on graph clustering. We focus on Affinity Clustering Bateni et al. (2017), a hierarchical method based on Borůvka's algorithm for constructing a minimum spanning tree (MST). For this practical task, we use the real-world LastFM Asia Social Network dataset Asuncion et al. (2007) (with $n = 7624$ instances, $p = 7842$ features). We apply PCA to project the data into a range of dimensions from $d = 1 \times 10^3$ to $d = 5 \times 10^3$ to obtain different datasets with varying dimensionality. To establish a more comprehensive baseline, we also compare against the `edge-flip` algorithm which directly adds noise to edge weights. This is a standard and well-known technique from prior work on privately approximating minimum spanning trees Sealfon (2016); Pagh et al. (2025)[4].

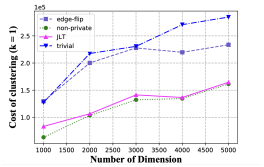

Figure 2: $k = 1$

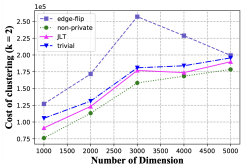

Figure 3: $k = 2$

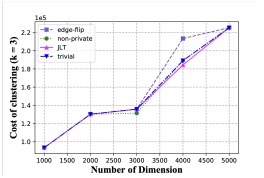

Figure 4: $k = 3$

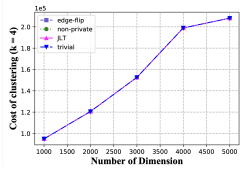

Figure 5: $k = 4$

Figure 6: Private Affinity Clustering.

We evaluate the clustering error (Definition 26) for the first four iterations (where k is the number of iterations) of Affinity Clustering. The results, averaged over 50 independent runs (Figure 6), demonstrate a dramatic advantage for the `JLT` method in the first round. We note that by the 4-th round, all points are grouped by each method into a single cluster, so the

[4]We note that while more sophisticated in-place perturbation algorithms exist for private MST Pinot (2018); Pagh and Retschmeier (2024); Pagh et al. (2025), they are unsuitable for hierarchical clustering tasks like Affinity Clustering, as they rely on privatizing sequential algorithms.

clustering error is identical for every method.

Acknowledgements

The work of Chenglin Fan was supported in part by the National Research Foundation of Korea (NRF) under Grant No. RS-2026-25484051.

References

Aamand, A., Chen, J. Y., Dalirrooyfard, M., Mitrović, S., Nevmyvaka, Y., Silwal, S., and Xu, Y. (2025). Breaking the $n^{1.5}$ additive error barrier for private and efficient graph sparsification via private expander decomposition. In *Forty-second International Conference on Machine Learning*.

Asuncion, A., Newman, D., et al. (2007). Uci machine learning repository.

Bateni, M., Behnezhad, S., Derakhshan, M., Hajiaghayi, M., Kiveris, R., Lattanzi, S., and Mirrokni, V. (2017). Affinity clustering: Hierarchical clustering at scale. *Advances in Neural Information Processing Systems*, 30.

Cai, H., Zheng, V. W., and Chang, K. C.-C. (2018). A comprehensive survey of graph embedding: Problems, techniques, and applications. *IEEE transactions on knowledge and data engineering*, 30(9):1616–1637.

Chami, I., Abu-El-Haija, S., Perozzi, B., Ré, C., and Murphy, K. (2022). Machine learning on graphs: A model and comprehensive taxonomy. *Journal of Machine Learning Research*, 23(89):1–64.

Chandra, R., Dinitz, M., Fan, C., and Zou, Z. (2024). Differentially private algorithms for graph cuts: A shifting mechanism approach and more. *arXiv preprint arXiv:2407.06911*.

Cohen-Addad, V., Fan, C., Lattanzi, S., Mitrovic, S., Norouzi-Fard, A., Parotsidis, N., and Tarnawski, J. (2022). Near-optimal correlation clustering with privacy. In Koyejo, S., Mohamed, S., Agarwal, A., Belgrave, D., Cho, K., and Oh, A., editors, *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*.

Dalirrooyfard, M., Mitrovic, S., and Nevmyvaka, Y. (2023). Nearly tight bounds for differentially private multiway cut. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, pages 24947–24965.

Dasgupta, A., Kumar, R., and Sarlós, T. (2010). A sparse johnson: Lindenstrauss transform. In *Proceedings of the forty-second ACM symposium on Theory of computing*, pages 341–350. ACM.

Datar, M., Immorlica, N., Indyk, P., and Mirrokni, V. S. (2004). Locality-sensitive hashing scheme based on p-stable distributions. In *Proceedings of the twentieth annual symposium on Computational geometry*, pages 253–262.

Deng, C., Gao, J., Upadhyay, J., Wang, C., and Zhou, S. (2025). On the price of differential privacy for hierarchical clustering. In *The Thirteenth International Conference on Learning Representations*.

Dietz, M. and Kerschbaum, F. (2025). Private shared random minimum spanning forests. *Proceedings on Privacy Enhancing Technologies*.

Dwork, C., McSherry, F., Nissim, K., and Smith, A. (2006). Calibrating Noise to Sensitivity in Private Data Analysis. In *TCC*, pages 265–284.

Eden, T., Liu, Q. C., Raskhodnikova, S., and Smith, A. (2025). Triangle counting with local edge differential privacy. *Random Structures & Algorithms*, 66(4):e70002.

Epasto, A., Mirrokni, V., Narayanan, S., and Zhong, P. (2023). k-means clustering with distance-based privacy. *Advances in Neural Information Processing Systems*, 36:19570–19593.

Halcrow, J., Mosoi, A., Ruth, S., and Perozzi, B. (2020). Grale: Designing networks for graph learning. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 2523–2532.

Hladík, R. and Tětek, J. (2024). Near-universally-optimal differentially private minimum spanning trees. *arXiv preprint arXiv:2404.15035*.

Imola, J., Epasto, A., Mahdian, M., Cohen-Addad, V., and Mirrokni, V. (2023). Differentially private hierarchical clustering with provable approximation guarantees. In Krause, A., Brunskill, E., Cho, K., Engelhardt, B., Sabato, S., and Scarlett, J., editors, *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pages 14353–14375. PMLR.

Imola, J., Murakami, T., and Chaudhuri, K. (2022). Differentially private triangle and 4-cycle counting in the shuffle model. In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*, pages 1505–1519.

Jenssen, M., Joos, F., and Perkins, W. (2018). On kissing numbers and spherical codes in high dimensions. *Advances in Mathematics*, 335:307–321.

Kane, D. M. and Nelson, J. (2014). Sparser johnson-lindenstrauss transforms. *Journal of the ACM (JACM)*, 61(1):4.

Kenthapadi, K., Korolova, A., Mironov, I., and Mishra, N. (2012). Privacy via the johnson-lindenstrauss transform. *arXiv preprint arXiv:1204.2606*.

Liu, J., Upadhyay, J., and Zou, Z. (2024). Optimal bounds on private graph approximation. In *Proceedings of the 2024 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1019–1049. SIAM.

Makarov, I., Kiselev, D., Nikitinsky, N., and Subelj, L. (2021). Survey on graph embeddings and their applications to machine learning problems on graphs. *PeerJ Computer Science*, 7:e357.

Nikolov, A. (2023). Private query release via the Johnson-Lindenstrauss transform. In *Proceedings of the 2023 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 4982–5002. SIAM.

Pagh, R. and Retschmeier, L. (2024). Faster private minimum spanning trees. *arXiv preprint arXiv:2408.06997*.

Pagh, R., Retschmeier, L., Wu, H., and Zhang, H. (2025). Optimal bounds for private minimum spanning trees via input perturbation. *Proceedings of the ACM on Management of Data*, 3(2):1–26.

Papernot, N. and McDaniel, P. (2018). Deep k-nearest neighbors: Towards confident, interpretable and robust deep learning. *arXiv preprint arXiv:1803.04765*.

Pinot, R. (2018). Minimum spanning tree release under differential privacy constraints. *arXiv preprint arXiv:1801.06423*.

Pinot, R., Morvan, A., Yger, F., Gouy-Pailler, C., and Atif, J. (2018). Graph-based clustering under differential privacy. *arXiv preprint arXiv:1803.03831*.

Rakhshan, B. and Rabusseau, G. (2020). Tensorized random projections. In *International Conference on Artificial Intelligence and Statistics*, pages 3306–3316. PMLR.

Sealfon, A. (2016). Shortest paths and distances with differential privacy. In *Proceedings of the 35th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems*, pages 29–41.

Steinke, T. and Ullman, J. (2017). Tight lower bounds for differentially private selection. In *2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 552–563. IEEE.

Suppakitpaisarn, V., Ponnoprat, D., Hirankarn, N., and Hillebrand, Q. (2025). Counting graphlets of size k under local differential privacy. *arXiv preprint arXiv:2505.12954*.

Xiao, Y. and Xiong, L. (2015). Protecting locations with differential privacy under temporal correlations. In *Proceedings of the 22nd ACM SIGSAC conference on computer and communications security*, pages 1298–1309.

Xu, X., Liu, C., Feng, Q., Yin, H., Song, L., and Song, D. (2017). Neural network-based graph embedding for cross-platform binary code similarity detection. In *Proceedings of the 2017 ACM SIGSAC conference on computer and communications security*, pages 363–376.

Checklist

1. For all models and algorithms presented, check if you include:
   (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. [Yes]
   (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. [Yes]
   (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. [Yes]

2. For any theoretical claim, check if you include:
   (a) Statements of the full set of assumptions of all theoretical results. [Yes]
   (b) Complete proofs of all theoretical results. [Yes]
   (c) Clear explanations of any assumptions. [Yes]

3. For all figures and tables that present empirical results, check if you include:
   (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). [Yes]
   (b) All the training details (e.g., data splits, hyperparameters, how they were chosen). [Yes]
   (c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). [Yes]
   (d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). [Yes]

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:

(a) Citations of the creator If your work uses existing assets. [Yes]

(b) The license information of the assets, if applicable. [Yes]

(c) New assets either in the supplemental material or as a URL, if applicable. [Yes]

(d) Information about consent from data providers/curators. [Yes]

(e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. [Yes]

5. If you used crowdsourcing or conducted research with human subjects, check if you include:

(a) The full text of instructions given to participants and screenshots. [Not Applicable]

(b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. [Not Applicable]

(c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. [Not Applicable]

Differentially Private Minimum Spanning Tree in Euclidean Graphs: Supplementary Materials

A Preliminaries

In this section, we introduce some basic technical definitions and lemmas that underpin our proofs.

Lemma 18 (Post processing Dwork et al. (2006)). *Let $\mathcal{M} : \mathcal{X} \to \mathcal{O}$ be a (ε, δ)-differentially private algorithm. Let $f : \mathcal{O} \to \mathcal{O}'$ be an arbitrary randomized mapping, then $f \circ \mathcal{A} : \mathcal{X} \to \mathcal{O}'$ is also (ε, δ)-differentially private.*

Lemma 19 (Basic composition Dwork et al. (2006)). *Let $d \in \mathcal{X}$ be a dataset and $\mathcal{M}_1, \cdots, \mathcal{M}_k$ be k algorithms such that $\mathcal{M}_i : \mathcal{X} \to \mathcal{O}_i$ preserves $(\varepsilon_i, \delta_i)$-DP. Then the composed algorithm $\mathcal{M} : \mathcal{X} \to \prod_{i=1}^{k} \mathcal{O}_i$ defined as $\mathcal{M}(d) = (\mathcal{M}_1(d), \cdots, \mathcal{M}_k(d))$ preserves $(\sum_{i=1}^{k} \varepsilon_i, \sum_{i=1}^{k} \delta_i)$-DP.*

The following introduces the Laplace distribution and its standard tail bound:

Definition 20 (Laplace distribution). *Given parameter b, Laplace distribution (with scale b) is the distribution with probability density function*

$$Lap(x|b) = \frac{1}{2b} \exp\left(-\frac{|x|}{b}\right).$$

We use $\mathrm{Lap}(b)$ to denote the Laplace distribution with scale b.

Lemma 21. *Let x be a random variable with $Lap(b)$ distribution. Then, $\mathbf{Pr}[|x| \geq tb] \geq \exp(-t)$.*

We also describe mechanisms that are commonly used in differential privacy. We begin by defining the sensitivity of a function.

Definition 22 (ℓ_p-sensitivity). *Let $f : \mathcal{X} \to \mathbb{R}^k$ be a query function on datasets. The sensitivity of f (with respect to $\mathcal{X}$) is*

$$\mathsf{sens}_p(f) = \max_{\substack{D, D' \in \mathcal{X}, \\ D \sim D'}} \|f(D) - f(D')\|_p.$$

Lemma 23 (Laplace mechanism). *Fix any $\varepsilon > 0$. Suppose $f : \mathcal{X} \to \mathbb{R}^k$ is a query function with ℓ_1-sensitivity $\mathsf{sens}_1(f)$. Then the mechanism*

$$\mathcal{M}(D) = f(D) + (Z_1, \cdots, Z_k)^\top$$

is $(\varepsilon, 0)$-differentially private, where $Z_1, \cdots, Z_k$ are i.i.d random variables drawn from $Lap(\mathsf{sens}_1(f)/\varepsilon)$.

Lemma 24 (Gaussian Mechanism). *Fix any $\varepsilon \in (0, 1)$ and $\delta \in (0, 1)$. Suppose $f : \mathcal{X} \to \mathbb{R}^k$ is a query function with ℓ_2-sensitivity $\mathsf{sens}_2(f)$. Then the mechanism the Gaussian Mechanism $\mathcal{M}$, defined by*

$$\mathcal{M}(D) = f(D) + \mathcal{N}(0, \sigma^2 \mathbf{I}_d),$$

is (ε, δ)-differentially private if the noise scale σ satisfies

$$\sigma \geq \frac{\mathsf{sens}_2(f)}{\varepsilon} \sqrt{2 \ln\left(\frac{1.25}{\delta}\right)}.$$

We recall the standard JLT transformation using random Gaussian matrix:

Lemma 25. *Fix any $0 < \eta < \frac{1}{2}$. Let M be a $r \times d$ matrix whose entries are i.i.d. sampled from $\mathcal{N}(0, 1)$. Then $\forall x \in \mathbb{R}^d$,*

$$\mathbf{Pr}\left[(1-\eta)\|x\|_2^2 \leq \frac{1}{r}\|Mx\|_2^2 \leq (1+\eta)\|x\|_2^2\right] \geq 1 - 2e^{-\eta^2 r/8}.$$

For the clustering task, we recall the following measure of utility introduced in Bateni et al. (2017):

Definition 26 (Bateni et al. (2017))**.** *The cost of a a cluster is the sum of edge weights of a minimum Steiner tree connecting all vertices of the cluster. The cost of a clustering is the sum of the costs of its clusters. Finally, a non-singleton clustering of a graph partitions its vertices into clusters of size at least* 2*.*

B Hardness of Approximating MST Under ρ-Node Privacy for General Graphs

In this section we instead consider the lower bound of finding the *maximum* spanning tree privately. This implies a lower bound for private minimum spanning tree approximation by simply flipping the edge weights: setting new edge weights $w'_e = W - w_e$, where $W = O(\sqrt{n})$ is the maximum edge weight in the graphs used to establish the lower bound. We prove the following lower bound:

Theorem 27. *Fix $0 < \varepsilon < 1$ and $0 < \delta < O(1/\sqrt{n})$. Suppose algorithm $\mathcal{A}$ for finding maximum spanning tree preserves (ε, δ)-differential privacy under ρ-node privacy (Definition 3). There exists a positively weighted graph G (which may depend on $\mathcal{A}$) such that*

$$\mathbb{E}_{\mathcal{A}}\left[\sum_{e\in\mathcal{A}(G)} w_e\right] \leq \sum_{e\in T^*} w_e - \Omega\left(\frac{\rho n^{1.5}}{\varepsilon}\right),$$

where T^* is the maximum spanning tree in G.

Definition 28. *A beta distribution denoted by $\mathsf{Beta}(\alpha, \beta)$ is a continuous distribution on $[0, 1]$ with two parameters $\alpha, \beta > 0$ and the probability density at $p \in [0, 1]$ is proportional to $p^{\alpha-1}(1-p)^{\beta-1}$. More formally,*

$$\mathbf{Pr}_{P\sim\mathsf{Beta}(\alpha,\beta)}[P \leq p] = \int_0^p \frac{t^{\alpha-1}(1-t)^{\beta-1}}{B(\alpha,\beta)}dt$$

where $B(\alpha, \beta)$ is the normalization factor.

The following proposition describes the anti-concentration property of beta distributions:

Proposition 29. *(Steinke & Ullman Steinke and Ullman (2017)) Fix $\beta > 0$ and $d \in \mathbb{N}$. Let $P_1, \cdots, P_d$ be independent samples from $\mathsf{Beta}(\beta, \beta)$. If $1 \leq \beta \leq \frac{1}{2}\log\left(\frac{d}{224}\right)$ then*

$$\mathop{\mathbb{E}}_{P_1,\cdots,P_d}\left[\max_{i\in[d]} P_i\right] \geq \frac{3}{4}.$$

Definition 30. *(The construction of the hard distribution based on star graphs.) Fix $d, n \in \mathbb{N}_+$ and $\beta > 0$. Let G be a star graph on n vertices, such that between the central vertex and each non-central vertex v, there are d multi-edges. For each edge e, first independently sample P_e from $\mathsf{Beta}(\beta, \beta)$, then let $X \in \{0, 1\}^{s\times m}$ be a random matrix in which $X_{i,e}$ is an independent Bernoulli random variable with mean P_e. Here $m = (n-1)d$ is the number of edges. We define the edge weight for each edge e as $w_e := \sum_{i=1}^{s} X_{i,e}$, which follows binomial distribution with mean sP_e.*

Remark 31. *This construction can be transformed into a simple graph without multi-edges through the following steps. Let G be the graph described as above. First, replace each non-central vertex v into d new vertices $v'_1, \cdots, v'_d$. Each original multi-edge incident to v is then reassigned to connect to one of these new vertices. Next connect $v'_1, \cdots, v'_d$ in sequence to form a path, where each edge in the path has weight $s + 1$. We denote the resulting graph as G'. It is straightforward to verify that all path edges* must *be included in any maximum spanning tree, as all other edges in G' have weights at most s by the construction. Consequently, computing the maximum spanning tree in the simple graph G' is equivalent to doing so in G as all $v_1, \cdots v_d$ are already connected in the maximum spanning tree.*

Let $\mathcal{G}$ be the "hard" distribution of graphs that we described in Definition 30. We prove the following property associated with $\mathcal{G}$:

Lemma 32. *Let $G \sim \mathcal{G}$, and T^* be the maximum spanning tree in G. Suppose we chose a proper β such that $1 \leq \beta \leq \frac{1}{2}\log(d/224)$, then*

$$\mathop{\mathbb{E}}_{\mathcal{G}}\left[\sum_{e\in T^*} w_e\right] \geq \frac{3}{4}\cdot s\cdot(n-1).$$

Proof. This proof mainly follows the anti-concentration property of random variables from Beta distribution. Notice that for each non-central vertex v, the maximum spanning tree of G must contain the edge incident to v with the largest weight. In particular, let $E(v)$ be the collection of edges connecting the central vertex to v, then finding the maximum spanning tree T^* equivalents selecting the maximum-weight edge for each non-central vertex v.

Given a non-central vertex $v \in [n-1]$ and an implementation of $\{P_e\}_{e\in E(v)}$, we let $e^*(v)$ be the edge e with largest P_e in $E(v)$. By the construction of the hard distribution $\mathcal{G}$, we have that

$$\begin{aligned}
\mathop{\mathbb{E}}_{\mathcal{G}}\left[\sum_{e\in T^*} w_e\right] &= \sum_{v\in[n-1]} \mathop{\mathbb{E}}_{\mathcal{G}}\left[\max_{e\in E(v)} : w_e\right] \\
&\geq \sum_{v\in[n-1]} \int_{p=0}^{1} \mathbb{E}\left[w_e : e = e^*(v) \mid \max_{e\in E(v)} P_e = p\right] \cdot \mathbf{Pr}\left[\max_{e\in E(v)} P_e = p\right] \\
&= \sum_{v\in[n-1]} \int_{p=0}^{1} sp \cdot \mathbf{Pr}\left[\max_{e\in E(v)} P_e = p\right] \\
&= (n-1)\cdot s \cdot \mathop{\mathbb{E}}_{e\in E(v)}\left[\max_{e\in E(v)} P_i\right] \\
&\geq \frac{3}{4}\cdot s\cdot (n-1).
\end{aligned}$$

Here, the last inequality comes from Proposition 29. □

The following lemma reduces the lower bound for (ε,δ)-differential privacy to $(1,\delta)$-differential privacy by re-weighting the edges:

Lemma 33. *(Pagh et al. Pagh et al. (2025)) Fix any $0 < \varepsilon, \delta < 1$. Suppose that there is an $(\varepsilon, \frac{e^\varepsilon - 1}{e-1}\cdot\delta)$-deferentially private MST algorithm M such that for every input graph G with MST denoted by T,*

$$\mathbb{E}_M\left[\sum_{e\in M(G)} w_e\right] \leq \left(\sum_{e\in T} w_e\right) + o\left(\frac{n^{1.5}}{\varepsilon}\right).$$

Then, there exists a $(1,\delta)$-deferentially private algorithm M' such that for every input graph G' with MST denoted by T',

$$\mathbb{E}_{M'}\left[\sum_{e\in M(G')} w'_e\right] \leq \left(\sum_{e\in T'} w'_e\right) + o\left(n^{1.5}\right).$$

Our lower bound relies on the following conclusion from Steinke and Ullman (2017):

Lemma 34. *(Theorem 3 in Steinke and Ullman (2017).) Let $\beta,\gamma,\rho,k > 0$ and $s,d \in \mathbb{N}^+$ be a fixed set of parameters. Let $P = (P_1,\ldots,P_m)$ be independent samples from the beta distribution $\mathsf{Beta}(\beta,\beta)$, and let $X \in \{0,1\}^{s\times m}$ be a random matrix such that every $X_{i,j}$ is an independent sample from Bernoulli distribution with mean P_j for every $i \in [s]$ and $j \in [m]$. Let $\mathcal{A} : \{0,1\}^{s\times m} \to \mathbb{R}^m$ be a $(1, \beta\gamma k/(s\rho))$-differentially private algorithm (where two datasets X and X' are neighboring if and only if they differ by at most one row). Assume that $\mathbb{E}_{P,X,\mathcal{A}}[\|\mathcal{A}(X)\|_2^2] = k$, $\|\mathcal{A}(X)\|_1 \leq \rho$ and*

$$\mathbb{E}_{P,X,\mathcal{A}}\left[\textstyle\sum_{j\in[m]} \mathcal{A}(X)_j \cdot \left(P_j - \frac{1}{2}\right)\right] \geq \gamma k\ , \tag{1}$$

then $s \geq \beta\gamma\sqrt{k}$.

We note that an algorithm for finding the maximum spanning tree in a weighted graph with public edge set can be considered as algorithm $\mathcal{A}$ in Lemma 34, by formulating its output as a m-dimensional Boolean vector that indicates which edges are chosen in the maximum spanning tree. In this case $\mathbb{E}_{P,X,\mathcal{A}}[\|\mathcal{A}(X)\|_2^2] = \|\mathcal{A}(X)\|_1 = n-1$. Further, the privacy notion in Lemma 34 aligns with the node-privacy under our construction of $\mathcal{G}$, as all edges are incident on the central vertex, so changing these edge weights by 1 equivalents changing one row of X. We are now ready to present the proof of Theorem 27:

Proof. (Of Theorem 27.) With Lemma 33, we now only consider the lower bound under $(1,\delta)$-DP algorithms. Let $\gamma = 0.02$, $s = \sqrt{n}/10$, $k = \rho = n-1$ and d be the least constant such that $\log(d/224) \geq 20$, and let $\beta = \frac{1}{2}\log(d/224)$. Let $\delta = \beta\gamma k/(s\rho) = O(1/\sqrt{n})$. For the sake of contradiction, we assume that there exists a $(1,\delta)$-deferentially private maximum spanning tree algorithm $\mathcal{A}$ such that

$$\mathbb{E}_{G\sim\mathcal{G},\mathcal{A}}\left[\sum_{e\in\mathcal{A}(G)} w_e\right] > \mathbb{E}_{G\sim\mathcal{G}}\left[\sum_{e\in T_G} w_e\right] - \frac{n^{1.5}}{1000},$$

where T_G is the maximum spanning tree of G. To apply Lemma 34, notice that

$$\begin{aligned}
&\mathbb{E}_{G,\mathcal{A}}\left[\sum_{e\in[m]} (\mathcal{A}(G))_e \left(s\cdot P_e - \frac{s}{2}\right)\right] \\
&= \mathbb{E}_{G,\mathcal{A}}\left[\sum_{e\in[m]} (\mathcal{A}(G))_e w_e\right] + \mathbb{E}_{G,\mathcal{A}}\left[\sum_{e\in[m]} (\mathcal{A}(G))_e \left(s\cdot P_e - w_e\right)\right] - \mathbb{E}_{G,\mathcal{A}}\left[\sum_{e\in[m]} (\mathcal{A}(G))_e \cdot \frac{s}{2}\right].
\end{aligned}$$

Recall that the weight w_e of each $e \in [m]$ is the sum of s independent Bernoulli random variables with mean P_e, then by the standard Chernoff-Hoeffding's inequality and the union bound, we have that $\mathbb{E}\left[\max_{e\in[m]} |w_e - s\cdot P_e|\right] \leq \sqrt{3s\log n}$. Combing this and the fact that $\|\mathcal{A}(G)\|_1 = n-1$ since a spanning tree has exactly $n-1$ edges, we have that

$$\mathbb{E}_{G,\mathcal{A}}\left[\sum_{e\in[m]} (\mathcal{A}(G))_e \left(s\cdot P_e - \frac{s}{2}\right)\right] \geq \mathbb{E}_{G,\mathcal{A}}\left[\sum_{e\in[m]} (\mathcal{A}(G))_e w_e\right] - (n-1)(\sqrt{3s\log n} + s/2). \tag{2}$$

By the assumption and the definition of $\mathcal{A}$, we see that

$$\begin{aligned}
\mathbb{E}_{G,\mathcal{A}}\left[\sum_{e\in[m]} (\mathcal{A}(G))_e w_e\right] &= \mathbb{E}_{G\sim\mathcal{G},\mathcal{A}}\left[\sum_{e\in\mathcal{A}(G)} w_e\right] > \mathbb{E}_{G\sim\mathcal{G}}\left[\sum_{e\in T_G} w_e\right] - \frac{n^{1.5}}{1000} \\
&> 0.75\cdot s\cdot(n-1) - \frac{n^{1.5}\log n}{1000},
\end{aligned}$$

where the last inequality comes from Lemma 32. Combing this and eq (2), we have that

$$\begin{aligned}
\mathbb{E}_{G,\mathcal{A}}\left[\sum_{e\in[m]} (\mathcal{A}(G))_e \left(s\cdot P_e - \frac{s}{2}\right)\right] &> 0.75\cdot s\cdot(n-1) - \frac{n^{1.5}}{1000} - (n-1)(\sqrt{3s\log n} + s/2) \\
&= \frac{1}{4}\cdot s\cdot(n-1) - \frac{n^{1.5}}{1000} - (n-1)\sqrt{3s\log n} \\
&> s\gamma(n-1)
\end{aligned} \tag{3}$$

for $s = \frac{\sqrt{n}}{10}$, $\gamma = 0.02$ and large enough n. This is equivalently saying that

$$\mathbb{E}_{G,\mathcal{A}}\left[\sum_{e\in[m]} (\mathcal{A}(G))_e \left(P_e - \frac{1}{2}\right)\right] > \gamma\cdot k$$

where $k = \mathbb{E}_{P,X,\mathcal{A}}[\|\mathcal{A}(X)\|_2^2] \equiv n-1$. Then, by Lemma 34, we have that

$$s \geq \beta\gamma\sqrt{k} \geq 0.2\cdot\sqrt{n-1},$$

which leads to a contradiction for large enough n as $s = \sqrt{n}/10$. Therefore, we have that

$$\mathbb{E}_{G\sim\mathcal{G},\mathcal{A}}\left[\sum_{e\in\mathcal{A}(G)} w_e\right] \leq \mathbb{E}_{G\sim\mathcal{G}}\left[\sum_{e\in T_G} w_e\right] - \frac{n^{1.5}}{1000},$$

which concludes the proof of Theorem 27 by the probabilistic method. □

C Missing Proofs

C.1 Proof of Theorem 8.

Theorem 35 (Restatement of Theorem 8)**.** *Algorithm 1 preserves (ε, δ)-DP under ρ-node privacy. Moreover, with high probability, the spanning tree T it outputs has a total weight that exceeds that of the true minimum spanning tree of G by at most $\widetilde{O}\left(\frac{n\rho\sqrt{\Delta}}{\varepsilon}\right)$.*

Proof. (Of Theorem 8.) First we show the privacy guarantee. Suppose $G = ([n], E, w)$ and $G' = ([n], E, w')$ are a pair of neighboring graphs under the ρ-node privacy notion, then clearly $\|w - w'\|_2 \le \rho\sqrt{\Delta}$. Thus, by basic Gaussian mechanism, step 2 in Algorithm 1 is enough to produce a private synthetic graph and the privacy guarantee comes from that DP algorithms are robust to post-processing. Next we consider the utility. For any $T \subseteq E$, we define $\mathrm{Cost}_G(T)$ be the sum of edge weights of edges in T on the graph G. Let $\hat{G}$ be the noisy graph obtained by adding Gaussian noise to edge weights in G as Algorithm 1. Then with high probability, for each $T \subseteq E$, we have

$$|\mathrm{Cost}_G(T) - \mathrm{Cost}_{\hat{G}}(T)| \le \frac{\sigma|T| \cdot \mathrm{polylog}(n, 1/\delta)}{\varepsilon} = \frac{|T|\rho\sqrt{\Delta} \cdot \mathrm{polylog}(n, 1/\delta)}{\varepsilon}.$$

Let T^* and $\hat{T}^*$ be the MST in G and $\hat{G}$ respectively, then

$$\begin{aligned}\mathrm{Cost}_G(\hat{T}^*) - \mathrm{Cost}_G(T^*) &\le \mathrm{Cost}_G(\hat{T}^*) + \mathrm{Cost}_{\hat{G}}(T^*) - \mathrm{Cost}_{\hat{G}}(\hat{T}^*) - \mathrm{Cost}_G(T^*) \\ &\le \frac{n\rho\sqrt{\Delta} \cdot \mathrm{polylog}(n, 1/\delta)}{\varepsilon}.\end{aligned}$$

This completes the proof. □

C.2 Proof of Theorem 11

Theorem 36 (Restatement of Theorem 11)**.** *Given a (d, r, η, β)-norm preserving projection P, Algorithm 2 preserves (ε, β)-DP under ρ-dist privacy.*

Proof. (Of Theorem 11.) We first show that if X and X' are a pair of neighboring dataset, then $P(X)$ and $P(X')$ are also "close" with high probability for the same Π. Recall that in ρ-dist privacy, there exists at most one $i \in [n]$ such that $\|x_i - x'_i\|_2 \le \rho$. Then, $P(X)$ and $P(X)$ can only differ in the i-th entry (i.e., $P(x_i)$ and $P(x'_i)$). Further, by the fact that P is a (d, r, η, β)-norm preserving projection, we have that with probability at least $1 - \beta$,

$$\begin{aligned}\frac{1}{\sqrt{r}} \cdot \|P(x_i) - P(x'_i)\|_1 &\le \|P(x_i) - P(x'_i)\|_2 \\ &\le \sqrt{(1+\eta)} \cdot \|x_i - x'_i\|_2 \\ &\le \rho\sqrt{(1+\eta)}.\end{aligned}$$

Let $\mathcal{M}$ be the post-processing step that projects $\hat{X}$ to $\widetilde{X}$ (from Step 4 to Step 7 of Algorithm 2). Clearly under ρ-dist privacy, $\mathcal{M}$ preserves ε'-differential privacy where $\varepsilon' = \varepsilon/(\rho\sqrt{r(1+\eta)})$. Then, the proof of Theorem 11 completes by directly applying Lemma 12 where we interpret $P(X)$ in Algorithm 2 as $\mathcal{S}(X)$ in Lemma 12. The coupling of $\mathcal{S}(X)$ and $\mathcal{S}(X')$ is constructed by letting both share the same randomness of the projection P. Finally, the proof is completed due to the post-processing property of differentially private algorithms (Lemma 18). □

C.3 Proof of Theorem 13

Theorem 37 (Restatement of Theorem 13)**.** *Fix $\varepsilon > 0$, $0 < \delta < n^{-4}$ and small η. There is a random norm preserving P such that Algorithm 2 is (ε, δ)-ρ-dist DP. Further, given any $X \in \mathbb{R}^{d \times n}$, and a fixed edge set $E \subseteq \binom{[n]}{2}$, with probability at least $1 - 1/\mathrm{poly}(n)$ (over the randomness of P and Laplace noise), Algorithm 2*

finds a spanning tree $\widetilde{T}$ of the Euclidean similarity graph $G = ([n], E, w)$ of the input dataset X such that

$$Cost_G(\widetilde{T}) \leq (1 + O(\eta))Cost_G(T) + \tilde{O}\left(\frac{n\rho\log(1/\delta)}{\varepsilon \cdot \eta^2}\right).$$

Here, T is one of the minimum spanning trees of G.

Proof. (Of Theorem 13.) We set $r = \frac{8\log(2/\delta)}{\eta^2}$. Then, we have that for any $\eta \in (0, \frac{1}{2})$,

$$\rho\sqrt{r(1+\eta)} \leq 4 \cdot \frac{\rho\sqrt{\log(2/\delta)}}{\eta}$$

and $\beta = 2e^{-\eta^2 r/8} = \delta$, which implies that Algorithm 2 is (ε, δ)-DP. For any pair of entries $u, v \in [n]$, we use w_{uv}, $\hat{w}_{uv}$ and $\tilde{w}_{uv}$ to denote the edge weights (i.e., the ℓ_2 distance) in the Euclidean embeddings of X, $\widehat{X}$ and $\widetilde{X}$, respectively. According to Lemma 25 and a union bound over all $O(n^2)$ pair of entries, we have that with probability at least $1 - O(1/n)$, for any $u, v \in [n]$ we have

$$\begin{aligned}
\tilde{w}_{uv} = \|\tilde{x}_u - \tilde{x}_v\|_2 &= \|\hat{x}_u + z_u - (\hat{x}_v + z_v)\|_2 \\
&\leq \|\hat{x}_u - \hat{x}_v\|_2 + \|z_u - z_v\|_2 \\
&= \frac{1}{\sqrt{r}}\|\Pi(x_u - x_v)\|_2 + \|z_u - z_v\|_2 \\
&\leq (1+\eta)\|x_u - x_v\|_2 + \|z_u - z_v\|_2 \\
&\leq (1+\eta) \cdot w_{uv} + \tilde{O}\left(\frac{\rho}{\varepsilon} \cdot \frac{\log(1/\delta)}{\eta^2}\right).
\end{aligned}$$

Here, the last inequality comes from the tail bound of Laplace distribution. For the same reasoning, we also have

$$\tilde{w}_{uv} \geq (1-\eta)w_{uv} - \tilde{O}(\rho\log(1/\delta)/(\varepsilon\eta^2)).$$

Let $\widetilde{T}$ be the minimum spanning tree found on $\widetilde{G}$ in Algorithm 2. Note according to Algorithm 2, $\widetilde{G}$ has the same edge set as G. Therefore, following the proof of Theorem 8, we have that

$$\begin{aligned}
\mathrm{Cost}_G(\widetilde{T}) &\leq \frac{1+\eta}{1-\eta} \cdot \mathrm{Cost}_G(T) + \tilde{O}\left(\frac{n\rho\log(1/\delta)}{\varepsilon \cdot \eta^2}\right) \\
&\leq (1 + O(\eta))\mathrm{Cost}_G(T) + \tilde{O}\left(\frac{n\rho\log(1/\delta)}{\varepsilon \cdot \eta^2}\right)
\end{aligned}$$

for small $0 < \eta$, which completes the proof.

□

C.4 Proof of Lemma 12

Lemma 38 (Restatement of Lemma 12)**.** *Let $\mathcal{S} : \mathcal{R} \to \mathcal{R}$ be a random mapping and $\mathcal{M} : \mathcal{R} \to \mathcal{Y}$ be an $(\varepsilon, 0)$-differentially private algorithm. Given $k > 0$, suppose that for any $x, y \in \mathcal{R}$ that $x \sim y$, there exists a coupling $\mathcal{C}$ of $\mathcal{S}(x)$ and $\mathcal{S}(y)$ such that with probability at least $1 - \delta'$ over the randomness of $\mathcal{C}$, $\mathcal{S}(x) \sim_k \mathcal{S}(y)$. Then, the mechanism $\mathcal{M} \circ \mathcal{S}$ preserves $(k\varepsilon, \delta')$-differential privacy.*

Proof. (Of Lemma 12.) We prove it in the discrete case, the continuous case follows identically. Let $\mathcal{R}_k \subseteq \mathcal{R}^2$ be the collection of all $(w, z) \in \mathcal{R}^2$ satisfying that $w \sim_k z$. For any neighboring datasets $x \sim y$ and any possible

output $a \in \mathcal{Y}$, we have

$$\begin{aligned}\frac{\mathbf{Pr}_{\mathcal{M},\mathcal{S}}[\mathcal{M}(\mathcal{S}(x))=a]}{\mathbf{Pr}_{\mathcal{M},\mathcal{S}}[\mathcal{M}(\mathcal{S}(y))=a]} &= \frac{\sum_{(x',y')\in\mathcal{R}^2}\mathbf{Pr}_{\mathcal{C}}[\mathcal{S}(x)=x',\mathcal{S}(y)=y']\cdot\mathbf{Pr}_{\mathcal{M}}[\mathcal{M}(x')=a]}{\sum_{(x',y')\in\mathcal{R}^2}\mathbf{Pr}_{\mathcal{C}}[\mathcal{S}(x)=x',\mathcal{S}(y)=y']\cdot\mathbf{Pr}_{\mathcal{M}}[\mathcal{M}(y')=a]}\\ &\le \frac{\sum_{(x',y')\in\mathcal{R}_k}\mathbf{Pr}_{\mathcal{C}}[\mathcal{S}(x)=x',\mathcal{S}(y)=y']\cdot\mathbf{Pr}_{\mathcal{M}}[\mathcal{M}(x')=a]+\delta'}{\sum_{(x',y')\in\mathcal{R}^2}\mathbf{Pr}_{\mathcal{C}}[\mathcal{S}(x)=x',\mathcal{S}(y)=y']\cdot\mathbf{Pr}_{\mathcal{M}}[\mathcal{M}(y')=a]}\\ &\le \frac{\delta'}{\mathbf{Pr}_{\mathcal{M},\mathcal{S}}[\mathcal{M}(\mathcal{S}(y)=a)]}+\max_{(x',y')\in\mathcal{R}_k}\frac{\mathbf{Pr}_{\mathcal{M}}[\mathcal{M}(x')=a]}{\mathbf{Pr}_{\mathcal{M}}[\mathcal{M}(y')=a]}\\ &\le e^{k\varepsilon}+\frac{\delta'}{\mathbf{Pr}_{\mathcal{M},\mathcal{S}}[\mathcal{M}(\mathcal{S}(y))=a]},\end{aligned}$$

which implies that

$$\mathbf{Pr}_{\mathcal{M},\mathcal{S}}[\mathcal{M}(\mathcal{S}(x))=a]\le e^{k\varepsilon}\cdot\mathbf{Pr}_{\mathcal{M},\mathcal{S}}[\mathcal{M}(\mathcal{S}(y))=a]+\delta',$$

and thus completes the proof. □

C.5 Proof of Theorem 16

Theorem 39 (Restatement of Theorem 16)**.** *Fix any $\varepsilon>0$ and $0\le\delta\le\frac{1-e^{-\varepsilon}}{2(1-e^{-2\varepsilon})}$. If the mechanism M for finding minimum spanning trees preserves (ε,δ)-differential privacy under ρ-dist privacy, then there exists an Euclidean graph G such that*

$$\mathbb{E}_{M(G)\to T}[w(T)]\ge(1+\eta)w(T^*)+\Omega\left(\frac{\rho n}{e^{\varepsilon}}\right).$$

for any $0\le\eta\le\frac{e^{-\varepsilon}}{32(1+e^{-\varepsilon})}$. Further, G is an Euclidean embedding of $d=O(\log n)$ dimensional data.

Proof. (Proof of Theorem 16.) Given a pair (u_a,u_b), we write $e_a=\{v,u_a\}$, $e_b=\{v,u_b\}$ and $e_{ab}=\{u_a,u_b\}$. Let T be the minimum spanning tree of the graph G described above. By the construction above, for any two different u_a and u_a' that are α-far from the central point, the angle between them is at least $\pi/3$. Therefore the distance between them satisfies that $w(\{u_a,u_a'\})>\alpha$. Then, clearly T includes e_a,e_{ab} for every pair (u_a,u_b), and the weight of T is $2k\alpha$. Further, every spanning tree of G must include either e_a or e_b to connect both u_a and u_b to the central vertex. We we are ready to present the proof of Theorem 16.

Let M be a mechanism on finding minimum spanning tree. For the sake of contradiction, we assume that M is (ε,δ)-differentially private. Given any pair (u_a,u_b), we define the *projection* of M onto the sub-graph induced $\{v,u_a,u_b\}$ as M_u. Specifically, M_u selects which two edges among $\{e_a,e_b,e_{ab}\}$ are included to connect this subgraph. With a slight abuse of notation, we write $e_a\in M_u$ to indicate that the mechanism includes e_a as an edge in the spanning tree. Fix a pair (u_a,u_b). Since at least one of $\{e_a,e_b\}$ must be selected into any feasible spanning tree, then

$$\mathbf{Pr}_{M(G)}(e_a\in M_u(G))+\mathbf{Pr}_{M(G)}(e_b\in M_u(G))\ge\mathbf{Pr}_{M(G)}(e_a\in M(G)\vee e_b\in M_u(G))=1 \tag{4}$$

Let G' be a new graph obtained from G by exchanging the position of a pair (u_a,u_b). By the post-processing property, M_u preserves (ε,δ)-differential privacy, and thus

$$\begin{aligned}\mathbf{Pr}_{M(G)}(e_b\in M_u(G)) &\ge \mathbf{Pr}_{M(G')}(e_b\in M_u(G'))\cdot e^{-2\alpha\varepsilon}-\frac{1-e^{-2\alpha\varepsilon}}{e^{\varepsilon}-1}\cdot\delta\\ &= \mathbf{Pr}_{M(G)}(e_a\in M_u(G))\cdot e^{-2\alpha\varepsilon}-\frac{1-e^{-2\alpha\varepsilon}}{e^{\varepsilon}-1}\cdot\delta.\end{aligned} \tag{5}$$

Here, the inequality comes from the group privacy. Plug eq. (4) into eq. (5), we have that

$$\mathbf{Pr}_{M(G)}(e_b\in M_u(G))\ge\frac{e^{-2\alpha\varepsilon}}{1+e^{-2\alpha\varepsilon}}-\frac{1-e^{-2\alpha\varepsilon}}{e^{\varepsilon}-1}\cdot\delta\ge\frac{e^{-\varepsilon}}{2(1+e^{-\varepsilon})}$$

if we choose $\alpha = 1/2$ and $0 \le \delta \le \frac{1-e^{-\varepsilon}}{2(1-e^{-2\varepsilon})}$. Suppose M chooses a spanning tree T. Note that M_u is not optimal if and only if it selects e_b. Therefore, the expected error can be reformulated as

$$\begin{aligned}\mathbb{E}_{M(G)}[w(T) - w(T^*)] &= \sum_{(u_a,u_b)} \alpha \cdot \mathbb{E}_{M(G)}[\mathbb{1}\{e_b \in M_u(G)\}] \\ &= k\alpha \cdot \mathbf{Pr}_{M(G)}(e_b \in M_u(G)) \ge k\alpha \cdot \frac{e^{-\varepsilon}}{2(1+e^{-\varepsilon})} \\ &\ge \frac{n}{16} \cdot \frac{e^{-\varepsilon}}{(1+e^{-\varepsilon})}.\end{aligned}$$

Combining the fact that $w(T^*) = 2k\alpha = n - 1$, we have that

$$\mathbb{E}_{M(G)}[w(T)] \ge (1+\eta)w(T^*) + \frac{n}{32} \cdot \frac{e^{-\varepsilon}}{(1+e^{-\varepsilon})}.$$

for any $0 \le \eta \le \frac{e^{-\varepsilon}}{32(1+e^{-\varepsilon})}$. This completes the proof of Theorem 16. □

C.6 Proof of Theorem 17

Theorem 40 (Restatement of Theorem 17)**.** *For any input dataset X whose Euclidean embedding is G, let $\mathcal{C}^*$ be the optimal non-singleton clustering of G. The first round of Algorithm 2 finds a clustering $\mathcal{C}$ of G such that*

$$\mathit{cost}(\mathcal{C}^*(G)) \ge \frac{\mathit{cost}(\mathcal{C}(G))}{2(1+O(\eta))} - \tilde{O}\left(\frac{(n - \#\mathcal{C}(G)) \cdot \log(1/\delta)}{\varepsilon\eta^2}\right)$$

Here, $\mathit{cost}(\mathcal{C}(G))$ is the cost of clustering $\mathcal{C}$ on graph G, and $\#\mathcal{C}(G)$ is the number of clusters of clustering $\mathcal{C}$.

Proof. (Of Theorem 17.) Let $G = ([n], E, \mathbf{w})$ represent the Euclidean embedding of the given dataset X. For each $v \in [n]$, denote by v_{near} as the nearest node in G to v, and define the ℓ_2 distance between v and v_{near} as $w(v, v_{near})$. Let $\mathcal{C}$ be an arbitrary non-singleton clustering on $[n]$. For each vertex v, let T_v be the minimum spanning tree of the cluster in $\mathcal{C}$ that v belongs to. Since $\mathcal{C}$ is non-singleton, then there is an edge in T_v that incident on v, and thus the weight of this edge is at least $w(v, v_{near})$. Since each edge can join at most two nodes into a cluster, then by a double counting principle, the sum of the weights of minimum spanning trees over all clusters is at least $\mathrm{cost}(\mathcal{C}(G)) \ge \sum_{v\in[n]} \frac{w(v,v_{near})}{2}$.

On the other hand, let $\mathcal{C}_{\mathcal{A}}$ be the clustering of the first round of Algorithm 2. For each vertex $v \in [n]$, one round of Algorithm 2 adds at most one edge. Suppose this edge is e_v. Also let e_v^* be the shortest edge incident on v. By the utility guarantee of edge weights, we have that with high probability, the following holds:

$$w(e_v) \le \frac{1}{1-\eta} \cdot \tilde{w}(e_v) + O(\log(1/\delta)/(\varepsilon\eta^2)), \quad \tilde{w}(e_v^*) \le (1+\eta) \cdot w(e_v^*) + O(\log(1/\delta)/(\varepsilon\eta^2)).$$

Since v_e is chosen by the algorithm, then we have $\tilde{w}(e_v) \le \tilde{w}(e_v^*)$. Combining these two facts we have that

$$w(e_v) \le \frac{1+\eta}{1-\eta} \cdot w(e_v^*) + O(\log(1/\delta)/(\varepsilon\eta^2)) = \frac{1+\eta}{1-\eta} \cdot w(v, v_{near}) + O(\log(1/\delta)/(\varepsilon\eta^2)).$$

It is easy to verify that the number of edges selected by Algorithm 2 is exactly $n - \#\mathcal{C}(G)$. Therefore, for small $\eta > 0$, we have

$$\mathrm{cost}(\mathcal{C}_{\mathcal{A}}(G)) \le \sum_{v\in[n]} (1+O(\eta)) \cdot w(v, v_{near}) + (n - \#\mathcal{C}(G)) \cdot O(\log(1/\delta)/(\varepsilon\eta^2)),$$

which completes the proof of Theorem 17. □

D Connections between ρ-node privacy and geometric dataset

Here, we illustrate why ρ-node privacy is a natural notion if we consider geometric dataset. In particular, we discuss how this privacy notion implies some natural privacy notions in geometric data.

D.1 ρ-dist privacy in Euclidean embeddings.

Consider a d-dimensional real-valued dataset $X = \{x_1, x_2, \cdots, x_n\} \in \mathbb{R}^{d\times n}$ and its natural Euclidean embedding where each data is a point in d-dimensional Euclidean space, and the weight between two points is just the ℓ_p distance between them, i.e., $w_{xy} = \|x - y\|_p$ for any $x, y \in X$. A natural privacy notion in such metric space is ρ-dist privacy Epasto et al. (2023). An algorithm is ρ-dist-DP if the algorithm protects privacy of a single data point if it is moved in the metric space by at most ρ. Clearly, ρ-node privacy encompasses ρ-dist privacy since once we move a point x by ρ, then

$$\begin{aligned}|w_{xy} - w'_{xy}| &= |\|x - y\|_p - \|x' - y\|_p| = |\|(x - x') + (x' - y)\|_p - \|x' - y\|_p| \\ &\le \|x - x'\|_p \le \rho.\end{aligned}$$

That is, if an algorithm preserves ρ-node privacy, it also preserves ρ-dist privacy.

D.2 ρ-rotate privacy in Euclidean or cosine similarity embeddings.

In high-dimensional unsupervised learning, cosine similarity is also commonly employed to quantify the distance between data points within a graph embedding. Again consider a d-dimensional dataset $X = \{x_1, x_2, \cdots, x_n\} \in \mathbb{R}^{d\times n}$, where each data point x is normalized to a unit vector. Then, the cosine similarity

$$S_C(x, y) = x^\top y/(\|x\|_2\|y\|_2) = x^\top y$$

between two unit-vectors $x, y \in X$ is determined by the angle between them, which we refer to as the angular distance. Similarly, we define ρ-rotate privacy as a variant of ρ-dist privacy, where two graphs are considered neighboring if one data point is rotated by an angle of at most ρ. In such graphs we could define the edge weight between x, y as

$$w_{xy} := \sqrt{2(1 - S_C(x, y))},$$

which is the ℓ_2 distance between x and y (so it is also an Euclidean embedding). Then, ρ-node privacy also encompasses the ρ-rotate privacy as

$$|w_{xy} - w'_{xy}| = \left|\sqrt{2(1 - S_C(x, y))} - \sqrt{2(1 - S_C(x', y))}\right| \le |\theta - \theta'| \le \rho$$

according to Proposition 41, where θ and θ' are the angles between x, y and x', y respectively. That is, under Euclidean embeddings, if an algorithm preserves ρ-node privacy, it also preserves ρ-rotate privacy.

Similarly, if we consider the cosine similarity embedding, we could define the edge weight between x, y in such embedding as

$$\overline{w}_{xy} := 1 - S_C(x, y) = 1 - x^\top y.$$

Then, under the ρ-rotate privacy, if we let θ and θ' be the angles between x, y and x', y respectively, it is easy to verify that

$$|\overline{w}_{xy} - \overline{w}'_{xy}| = |\cos(\theta) - \cos(\theta')| \le |\theta - \theta'| \le \rho$$

as $|\frac{d\cos(\theta)}{d\theta}| \le 1$. Therefore, under cosine similarity embeddings, if an algorithm preserves ρ-node privacy, it also preserves ρ-rotate privacy.

Proposition 41. *For any $\theta_1, \theta_2 \in [0, \pi/2]$, we have*

$$\left|\sqrt{2 - 2\cos(\theta_1)} - \sqrt{2 - 2\cos(\theta_2)}\right| \le |\theta_1 - \theta_2|.$$

Proof. Let $f(\theta) = \sqrt{2 - 2\cos(\theta)}$ for all $\theta \in [0, \pi/2]$, it is enough to show that $|\frac{df(\theta)}{d\theta}| \le 1$. Indeed, we have that

$$\begin{aligned}\frac{d}{d\theta}\sqrt{2 - 2\cos(\theta)} &= \frac{1}{2}(2 - 2\cos(\theta))^{-\frac{1}{2}} \cdot \frac{d}{d\theta}[2 - 2\cos(\theta)] \\ &= \frac{\sin(\theta)}{\sqrt{2 - 2\cos(\theta)}} = \sqrt{\frac{1 - \cos^2(\theta)}{2 - 2\cos(\theta)}} \\ &= \sqrt{\frac{1 + \cos(\theta)}{2}} \le 1,\end{aligned}$$

which completes the proof. □