Beyond Square-root Error: New Algorithms for Differentially Private All Pairs Shortest Distances

Chengyuan Deng *  Jie Gao †  Jalaj Upadhyay ‡  Chen Wang §

December 16, 2022

Abstract

We consider the problem of releasing all pairs shortest path distances (APSD) in a weighted undirected graph with differential privacy (DP) guarantee. We consider the weight-level privacy introduced by Sealfon [PODS’16], where the graph topology is public but the edge weight is considered sensitive and protected from inference via the released all pairs shortest distances. The privacy guarantee ensures that the probability of differentiating two sets of edge weights on the same graph differing by an $\ell_1$ norm of 1 is bounded. The goal is to minimize the additive error introduced to the released APSD while meeting the privacy guarantee. The best bounds known (Chen et al. [SODA’23]; Fan et al. [Arxiv’22]) is $\tilde{O}(n^{2/3})$ additive error for $\varepsilon$-DP and an $\tilde{O}(n^{1/2})$ additive error for $(\varepsilon,\delta)$-DP.

In this paper, we present new algorithms with improved additive error bounds: $\tilde{O}(n^{1/3})$ for $\varepsilon$-DP and $\tilde{O}(n^{1/4})$ for $(\varepsilon,\delta)$-DP, narrowing the gap with the current lower bound of $\Omega(n^{1/6})$ for $(\varepsilon,\delta)$-DP. The algorithms use new ideas to carefully inject noises to a selective subset of shortest path distances so as to control both ‘sensitivity’ (the maximum number of times an edge is involved) and the number of these perturbed values needed to produce each of the APSD output. In addition, we also obtain, for $(\varepsilon,\delta)$-DP shortest distances on lines, trees and cycles, a lower bound of $\Omega(\log n)$ for the additive error through a formulation by the matrix mechanism.
## Contents

1 Introduction  
   1.1 Our Contributions ......................................................... 1  
   1.2 Technique Overview ....................................................... 2  

2 Notations and Preliminaries .................................................. 3  

3 Tighter Analysis for Sublinear-Error DP Shortest Distances Algorithms  
   3.1 The Landmark Shortcut Algorithms ...................................... 6  
   3.2 Tighter Bounds on The Landmark Shortcut Algorithms ................... 8  

4 Improved Algorithms .......................................................... 9  
   4.1 Algorithm for $\epsilon$-DP ............................................... 10  
   4.2 Algorithm for $(\epsilon, \delta)$-DP ........................................ 13  

5 Bounds for Path, Tree and Cycle Graphs ................................... 15  
   5.1 Query Matrix in the Matrix Mechanism .................................... 16  
   5.2 The Lower bound .......................................................... 18  

A Technical Preliminaries for Matrix Mechanism  
   A.1 Review of Matrix Mechanism ............................................... 22  

B Proofs in Section 3  
   B.1 Proofs in Proposition 3.1 ................................................ 23  
   B.2 Proof of Lemma 3.4 ...................................................... 25  

C Proofs of Section 4 ............................................................ 26  

D Combinatorial Algorithms under Matrix Mechanism  
   D.1 The Landmark Shortcut-based Algorithms ................................ 27  
   D.2 Matrix Mechanism View for Canonical Segments .......................... 28
1 Introduction

Given a graph, $G := (V, E, w)$, whose topology, $E$, is public knowledge but edge weights, $w$, are considered to be private, sensitive information, the problem of private release of all pairs shortest distances asks whether one can publish all pairs shortest distances, approximately, such that the precise weight on any edge cannot be inferred up to a given privacy budget. This problem was initially formulated by Sealfon [Sea16] using the notion of differential privacy [DMNS16]. In particular, we consider a public graph $G$ with two sets of positive edge weights vectors $w$ and $w'$, which are called neighbors if their $\ell_1$ distance is at most one. A differentially private mechanism $A$ operates on the private (ground truth) edge weights and outputs an approximation of all pairs shortest path distances. Privacy of edge weights is guaranteed if the probability of $A$ generating similar output, using two neighboring sets of edge weights, is bounded. Intuitively, this means that an attacker with the published (approximate) all pairs shortest distances cannot infer too much of the real edge weights.

This problem is motivated by many networking applications where the network topology is public knowledge but edge weights can be sensitive. One example is the road map in a real-time navigation system. The weight of an edge maps to the latency of traversing the corresponding road segment, which is derived from traffic data that could be contributed by private users. In another example, a company with a private internal network would like to release end-to-end latency information but would not want to release detailed information on edge capacities, which could be used to derive network bottlenecks or vulnerabilities. In both cases, the network topology is public knowledge but the edge weights are sensitive information that should be protected.

Differentially private (DP in short) mechanisms address this problem by considering adding perturbations to all pairs shortest distances output. The magnitude of perturbation shall be enough to ensure specified privacy guarantee parameters and ideally as small as possible to get a good approximation of all pairs shortest distances. Sealfon in [Sea16] developed a mechanism with $\tilde{O}(n)$ additive error on a graph of $n$ vertices. It was also raised as an open problem [Sea20] whether there is a DP mechanism with sublinear (i.e., $o(n))$ additive error. This is achieved by two recent concurrent works [CGK+23, FLL22], which give essentially the same asymptotic bound of $\tilde{O}(n^{2/3})$ additive error for pure $\varepsilon$-DP, and $\tilde{O}(\sqrt{n})$ additive error for $(\varepsilon, \delta)$-DP. Moreover, Chen et. al. [CGK+23] proved a lower bound of $\Omega(n^{1/6})$ additive error for $(\varepsilon, \delta)$-DP. Note that all these results consider arbitrary graphs. There are $\varepsilon$-DP algorithms with polylog $n$ additive error if the graph is a tree [Sea16] or a planar graph [GDSG20].

1.1 Our Contributions

In this paper, we make significant progress towards the optimal bounds for differentially private release of all pairs shortest distances. The results by [CGK+23, FLL22] leave a significant gap between the upper bound of $\tilde{O}(\sqrt{n})$ and the lower bound $\tilde{O}(\Omega(n^{1/6}))$. Our main result gives quadratic improvements (on the exponent on $n$) for both $\varepsilon$-DP and $(\varepsilon, \delta)$-DP algorithms.

Main Result (Informal version of Theorem 2). Given a graph of $n$ vertices, for any $\varepsilon, \delta \in (0, 1)$, there exists:

1. (Pure-DP algorithm) An $\varepsilon$-differentially private algorithm that outputs all pairs shortest distances with additive error at most $\tilde{O}(n^{1/3}/\varepsilon)$ with high probability.

2. (Approximate-DP algorithm) An $(\varepsilon, \delta)$-differentially private algorithm that outputs all pairs shortest distances with additive error at most $\tilde{O}(\frac{1}{n^{1/4}} \cdot \log^{3/2}(\frac{1}{\varepsilon})$) with high probability.

Our algorithms leverage on new ideas to use the overlaps of the shortest path and the concentration of the summation of Laplace random variables (see Section 1.2 for more details). En route to Section 1.1, we show that by using the concentration of Laplace random variables, we can reduce the additive error of the algorithms in [CGK+23, FLL22] to $\tilde{O}(n^{2/3})$ and $\tilde{O}(n^{1/3})$, respectively for $\varepsilon$-DP and $(\varepsilon, \delta)$-DP. Our $(\varepsilon, \delta)$-DP algorithm combines the ideas in our $\varepsilon$-DP algorithms with $\varepsilon$-DP distance release on trees as in [Sea16, FL22].
We additionally show a $\Omega(\log^2 n)$ lower bound on the variance of $\ell_\infty$-error (also known as $\ell_\infty^2$-error [ENU20]) for lines, trees, and cycles for any approximate differentially private algorithm. There is an $(\varepsilon, \delta)$-differentially private algorithm that outputs all pair shortest distances with $\ell_\infty$-additive error $O\left(\log^{1.5} n\right)$ with high probability. This can be shown by using an approximate differentially private continual counting mechanism in the algorithm of Sealfon [Sea16]. The lower bound is obtained through a new perspective of answering all pair shortest distance using matrix mechanism, and it leverages known results for the completely bounded norm for certain matrices [Pau82]. As a byproduct, we derive the matrix mechanism equivalences of various combinatorial algorithms, including that in this paper and prior works [CGK23, FLL22]. We believe this viewpoint can be of independent interest.

1.2 Technique Overview

In general, differentially private mechanisms add perturbation to data samples. There are two standard primitives, namely output perturbation, where random noises are added to the final data output, and input perturbation, where random noises are added to each data element. Both mechanisms for the private release of APSD were developed in Sealfon’s paper and have the same asymptotically linear additive error [Sea16]. For simplicity in this section, we assume that $\varepsilon = O(1)$ and $\delta = \Theta(n^{-\log n})$ to focus on the improvement as a function of $n$. We also make the standard assumption that shortest paths are unique without loss of generality – if not, then we can perturb the weights slightly to get unique shortest paths.

We first explain the challenges in improving these two mechanisms. To guarantee privacy, the noise in the output perturbation should take a magnitude of the sensitivity of the function, the shortest distance in our case. In the shortest distance application, if the edge weight changes by 1 in the $\ell_1$ norm, there can be at most $O(n^2)$ $(u, v)$-pair shortest distances being impacted for $u, v \in V$. As such, if we apply a crude output perturbation, the noise for each pair of shortest distance should be $\tilde{O}(n^2)$ for $\varepsilon$-differential privacy and $\tilde{O}(n)$ for $(\varepsilon, \delta)$-differential privacy. On the other hand, with input perturbation, one can add a Laplace noise of magnitude proportional to $1/\varepsilon$ to each edge weight. This satisfies $\varepsilon$-privacy, but the shortest path may have up to order $n$ edges, and the noises on edges are accumulated with a total error of $\tilde{O}(n)$.

To circumvent the challenges, the recent breakthrough papers [CGK23, FLL22] use roughly the same idea of combining input and output perturbations. The general strategy is to sample a set $S$ of $s$ vertices and apply output perturbation on shortest distances between all pairs of vertices $(u, v) \in S \times S$. Since there are at most $O(s^2)$ vertex pairs in $S$, an $O(s^2)$-magnitude output perturbation suffices to privatize the distances among $S$. In addition, they use input perturbation of $O(1)$ on all edges in the original graph. To analyze the privacy and additive error, observe that, with high probability, any path $P(u, v)$ that is longer than $\tilde{O}(n/s)$ will have at least one vertex of $S$. As such, we can always decompose $P(u, v)$ as paths outside $S$ and inside $S$. For the paths inside $S$, the privacy is guaranteed by input perturbation, and the additive error is bounded by $\tilde{O}(n/s)$. On the other hand, for the paths outside $S$, the privacy is guaranteed by output perturbation, and the additive error is bounded by $\tilde{O}(s^2)$ for each pair. Therefore, by choosing the appropriate size of $S$ ($s \approx n^{1/3}$ for $\varepsilon$-DP and $s \approx n^{1/2}$ for $(\varepsilon, \delta)$-DP) and applying privacy composition, we obtain additive errors of $\tilde{O}(n^{2/3})$ for $\varepsilon$-DP and $\tilde{O}(\sqrt{n})$ for $(\varepsilon, \delta)$-DP.

Our improvement of the upper bound comes with three observations. First, we observe that the work of [CGK23, FLL22] uses summations of worst-case errors induced by each perturbation. However, note that the perturbation noises usually have light tails; if we add a number of Laplace noises, we can actually bound the additive error with the much stronger tail bound of summation. Basically, if we add up $k$ independent Laplace noises with scale parameter $b_1, b_2, \ldots, b_k$, the sum is no greater than $\sqrt{\sum b_i^2}$ with a constant probability. Applying this tighter tail bound on the above mechanism can immediately improve the error bound to $\tilde{O}(\sqrt{n/s + s^2})$ for $\varepsilon$-DP and $\tilde{O}(\sqrt{n/s + s^2})$ for $(\varepsilon, \delta)$-DP. By choosing the best values of $s$ we get $\tilde{O}(n^{2/3})$ for $\varepsilon$-DP and $\tilde{O}(n^{1/3})$ for $(\varepsilon, \delta)$-DP.

To further improve the upper bound, we actually need to tweak the ideas of input and output pertur-
bations. In general, the error due to output perturbation is defined by the sensitivity of the function – how many entries will be changed when we have neighboring graphs. The error for input perturbation depends on the graph hop diameter, i.e., the maximum number of edge weights that we need to sum up to output any shortest distance. The previous mechanisms pre-compute all pairs shortest distances in $S$ using recent advances in answering linear queries. This effectively adds shortcuts for $S \times S$ to the graph $G$ to reduce the graph hop diameter. The cost of reducing the hop diameter of the graph with extra shortcuts comes at the cost of increased sensitivity of these shortcuts. A natural question is, can we avoid increasing the sensitivity for shortcuts $S \times S$?

**Pure-DP algorithm.** The main idea in our first solution is to choose shortcuts with small sensitivity. By the assumption of unique shortest path, any two shortest paths would either be completely disjoint or intersect at exactly one common sub-path. For every intersecting shortest path between vertices $(u_1, u_2) \in V \times V$, we name $u_1, u_2$ as the cut vertices. Since there are $\binom{n}{2}$ shortest paths for all pairs in $S$, there are at most $O(s^2)$ cut vertices on any shortest path $P(u, v)$ with $(u, v) \in S \times S$. For every $(u, v) \in S \times S$, we cut the path $P(u, v)$ along these cut vertices into $O(s^2)$ canonical segments and pre-compute their length using output perturbation. The good thing is that the maximum sensitivity for the length of a canonical segment is one – since no two canonical segments can share any common edge. Reducing sensitivity by a multiplicative factor of $s^2$ at the cost of increasing the hop diameter by an additive value of $s^2$ turns out to be beneficial when we calculate the final additive error, which is $\tilde{O}(\sqrt{n/s+s^2})$, for our $\varepsilon$-DP algorithm. Plugging in $s = n^{1/3}$, we can get an error of $O(n^{1/3})$ and an $\varepsilon$-DP algorithm.

**Approximate-DP algorithm.** Our solution for $(\varepsilon, \delta)$-DP exploits properties of strong composition [DRV10], which allows us to massage $k$ $(\varepsilon, \delta)$-DP mechanisms into a $(\varepsilon', \delta')$-DP mechanism, where $\varepsilon' \approx \varepsilon \sqrt{k}$ and $\delta' \approx k \delta$. Our strategy to leverage strong composition is to build a shortest path tree rooted at each vertex in the sampled set $S$. Tree graphs admit much better differentially private mechanisms – one can get poly-logarithmic additive error [Sea16, FL22]. Now for any two vertices $u, v$ in $G$, if the shortest path $P(u, v)$ has more than $O(n/s)$ vertices, $P(u, v)$ has at least one vertex $w$ in $S$ with high probability. Thus the length of $P(u, v)$ is taken as the sum of length $P(u, w)$ and $P(w, v)$, which, can be obtained by using pre-computed distances between $(u, w)$ and $(v, w)$ in the shortest path tree rooted at $w$. The sensitivity of an edge in this case goes up – an edge can appear in possibly all the $s$ trees. Thus, on the trees we take $(O(\varepsilon / \sqrt{s}), \delta/2s)$-differentially private mechanisms. The composition of $s$ of them gives $(\varepsilon, \delta)$-DP. The final error bound is $\tilde{O} \left(\sqrt{n/s + \sqrt{s}}\right)$. Optimizing the error by setting $s = \tilde{O}(\sqrt{n})$ gives an $(\varepsilon, \delta)$-DP mechanism with additive error of $\tilde{O}(n^{1/4})$.

**Remark 1.** Our scheme for approximate-DP algorithm can also be applied to pure-DP regime to obtain the same upper bound of $\tilde{O}(n^{1/3})$, using basic composition theorem (Proposition 2.5) and replacing Gaussian mechanism with Laplace mechanism. However, there will be an extra $\log^2 n$ on the additive error over the pure-DP algorithm described above.

2 Notations and Preliminaries

**Notation.** We use $G = (V, E, w)$ to denote a weighted graph on vertex set $V$ and edges $E$, where $w$ is a weight function $w : E \rightarrow \mathbb{R}^+$ that defines the edge weights on $G$. An edge $e \in E$ is also denoted by the tuple $(u, v)$ if $u$ and $v$ are its endpoints. We start with the definition of weight-level privacy and formalization of our problem as follows.

**Definition 1 (Neighboring Weights).** Two graphs $G = (V, E, w)$ and $G' = (V, E, w')$ with the same vertex set $V$ and edge set $E$, and with weight functions: $w, w' : E \rightarrow \mathbb{R}^+$ are neighboring, denoted as $G \sim G'$, if

$$\sum_{e \in E} |w(e) - w'(e)| \leq 1$$

When it is clear from context, we abuse the notation and denote the above by $\|w - w'\|_1 \leq 1$ – this should not be confused with the standard notation of $\ell_1$ norm of a functional.
Sealfon [Sea16] introduced a notion of neighboring graphs where we know the topology of the graph, but not the edge weight. In particular, Sealfon studied the following problem:

**Definition 2** (Weight-level differential privacy for graphs [Sea16]). For any graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, w)$, let $A$ be an algorithm that takes as input a weight function $w : \mathcal{E} \rightarrow \mathbb{R}^+$. Then $A$ is $(\epsilon, \delta)$-differentially private on $\mathcal{G}$ if for all pairs of neighboring weight functions $w, w'$ and all sets of possible outputs $S$, we have that

$$\Pr[A(w) \in S] \leq e^\epsilon \Pr[A(w')] + \delta$$

If $\delta = 0$, we say $A$ is $\epsilon$-differentially private on $\mathcal{G}$.

We move forward to the definition of the approximate all-pair shortest distance problem. For vertices $v_1, \ldots, v_k \in \mathcal{V}$, the weight of a path $P = \langle v_1, \cdots, v_k \rangle$ is a function, weight($\cdot$), that takes as input a path $P$ and is defined as below

$$\text{weight}(P) = \sum_{i=1}^{k-1} w(v_i, v_{i+1}).$$

The shortest path distance\(^2\) between $u, v$ is a function $d : \mathcal{V} \times \mathcal{V} \rightarrow \mathbb{R}^+$, defined as

$$d(u, v) = \min \{ \text{weight}(P) : P = \langle u, v_1, \cdots, v_k, v \rangle \text{ is a path from } u \text{ to } v \} ,\quad (1)$$

We focus on the problem of releasing approximate all-pair shortest path distances.

**Definition 3** (Approximate-APSD). A randomized algorithm $A$ is $(\alpha, \beta)$-accurate for all pair shortest distance (APSD), if, given a connected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, it outputs a function $\hat{d} : \mathcal{V} \times \mathcal{V} \rightarrow \mathbb{R}$ such that

$$\Pr \left[ \max_{(u,v) \in \mathcal{V} \times \mathcal{V}} |\hat{d}(u, v) - d(u, v)| \leq \alpha \right] \geq 1 - \beta,$$

where the randomness is over the internal private coin tosses of $A$.

In other words, the focus of this paper is to design an algorithm satisfying Definition 2 and has as small a value of $\alpha$ as possible. Further, note that we are interested in only outputting a $(\mathcal{V} \times \mathcal{V})$ vector representing the estimates of all pair shortest distance.

**Tools from Probability Theory** We first introduce some well-known results from probability theory. We refer interested readers to the standard textbooks on this subject for more details [Wai19].

**Definition 4** (Laplace distribution). We say a zero-mean random variable $X$ follows the Laplace distribution with parameter $b$ (denoted by $X \sim \text{Lap}(b)$) if the probability density function of $X$ follows

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

**Definition 5** (Gaussian distribution). We say a zero-mean random variable $X$ follows the Gaussian distribution with variance $\sigma^2$ (denoted by $X \sim \mathcal{N}(0, \sigma^2)$) if the probability density function of $X$ follows

$$p(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \cdot \exp \left( -\frac{x^2}{2\sigma^2} \right).$$

Both Laplace and Gaussian random variables have nice concentration properties. Furthermore, we can get stronger concentration results by the summation of both random variables [Wai19]. These techniques were used in privacy applications by previous work, e.g. [CSS11] for differentially private continual counting.

---

\(^2\) By convention we assume that the graph is connected; otherwise the shortest path distance is considered for vertex pairs in each connected component separately.
Lemma 2.1 (Sum of Laplace random variables, [CSS11, Wai19]). Let \( \{X_i\}_{i=1}^m \) be a collection of independent random variables such that \( X_i \sim \text{Lap}(b_i) \) for all \( 1 \leq i \leq m \). Then, for \( \nu \geq \sqrt{\sum b_i^2} \) and \( 0 < \lambda < \frac{2 \sqrt{2} \nu^2}{b} \) for \( b = \max_i \{b_i\} \),
\[
\Pr \left[ \left| \sum_i X_i \right| \geq \lambda \right] \leq 2 \cdot \exp \left( -\frac{\lambda^2}{8\nu^2} \right).
\]

Lemma 2.2 (Sum of Gaussian random variables, [Wai19]). Let \( \{X_i\}_{i=1}^m \) be a collection of independent random variables such that \( X_i \sim \mathcal{N}(\mu, \sigma^2) \) for all \( 1 \leq i \leq m \). Then,
\[
\Pr \left[ \left| \frac{\sum_i X_i}{m} - \mu \right| \geq \lambda \right] \leq 2 \cdot \exp \left( -\frac{\lambda^2}{2\sigma^2} \right).
\]

Tools in Differential Privacy

We proceed to existing tools used frequently in differential privacy:

Definition 6 (Sensitivity). Let \( p \geq 1 \). For any function \( f : \mathcal{X} \to \mathbb{R}^k \) defined over a domain space \( \mathcal{X} \), the \( \ell_p \)-sensitivity of the function \( f \) is defined as
\[
\Delta f, p = \max \left\{ \left\| f(w) - f(w') \right\|_p : w, w' \in \mathcal{X}, w \sim w' \right\}.
\]
Here, \( \left\| x \right\|_p := \left( \sum_{i=1}^d |x[i]|^p \right)^{1/p} \) is the \( \ell_p \)-norm of the vector \( x \in \mathbb{R}^d \) and \( x[i] \) denote the \( i \)-th coordinate.

Based on Laplace distribution, we can now define Laplace mechanism – a standard DP mechanism that adds noise sampled from Laplace distribution with scale dependent on the \( \ell_1 \)-sensitivity of the function. The formal definition is as follows.

Definition 7 (Laplace mechanism). For any function \( f : \mathcal{X} \to \mathbb{R}^k \), the Laplace mechanism on input \( w \in \mathcal{X} \) samples \( Y_1, \ldots, Y_k \) independently from \( \text{Lap}(\Delta f, 1/\epsilon) \) and outputs
\[
M_{\epsilon, f}(w) := f(w) + (Y_1, \ldots, Y_k).
\]

The following privacy property of Laplace mechanism is known.

Proposition 2.3 (Laplace mechanism [DMNS16]). The Laplace mechanism \( M_{\epsilon, f} \) is \( \epsilon \)-differentially private.

Similar to Laplace mechanism, we can define the Gaussian mechanism:

Definition 8 (Gaussian mechanism). For any function \( f : \mathcal{X} \to \mathbb{R}^k \), the Gaussian mechanism on input \( w \in \mathcal{X} \) samples \( Y_1, \ldots, Y_k \) independently from \( \mathcal{N}(0, 2\Delta f^2 \log(1.25/\delta) \epsilon^2) \) and outputs
\[
M_{\epsilon, \delta}(f) := f(w) + (Y_1, \ldots, Y_k).
\]

The following privacy property of Gaussian mechanism is known.

Proposition 2.4 (Gaussian mechanism [DKM+06]). The Gaussian mechanism \( M_{\epsilon, \delta} \) is \((\epsilon, \delta)\)-differentially private.

Finally, we introduce the celebrated private composition lemmas, which preserve privacy when combining multiple differentially private mechanisms even against adaptive adversary.

Proposition 2.5 (Composition theorem [DMNS16]). For any \( \epsilon > 0 \), the adaptive composition of \( k \) \( \epsilon \)-differentially private algorithms is \( k\epsilon \)-differentially private.
In this section, we show that, by a tighter analysis, previous algorithms \([CGK+3]\) obtain the optimal bound for \(\epsilon\) shortest distances from the first vertex of \(P\) distance, \(\hat{\delta}(P)\). As such, for each vertex pair \((u, v)\) between all pairs of vertices in \(S \subseteq V\) we start with an observation that the DP algorithms proposed in both the recent papers \([CGK+3]\) and give a self-contained analysis (details are in Appendix B). Here and throughout, we call these algorithms the landmark shortcut algorithms and use this as the base case result to be improved.

Concretely, the scheme can be describe as:

(i) add Laplace noise of \(\text{Lap}(O(\text{poly}(|S|)))\) to all shortest distances between vertices in \(S\), and (ii) add Laplace noise of \(\text{Lap}(O(1))\) on all edges. (We omit the dependency on \(\epsilon\) here for simplicity.) By a careful choice of the sample size \(s\), we can balance the parameters and obtain the optimal bound for \(\epsilon\)-DP and \((\epsilon, \delta)\)-DP, respectively. For simplicity, we follow the practice from Chen et al. \([CGK+23]\) and name step (i) as output perturbation and and step (ii) as input perturbation. We now formalize the above high-level strategy as the following meta-algorithm (which will be realized in \(\epsilon\)-DP and \((\epsilon, \delta)\)-DP under different parameters).

---

**Proposition 2.6 (Strong composition theorem [DRV10]).** For any \(\epsilon, \delta \geq 0\) and \(\delta' > 0\), the adaptive composition of \(k(\epsilon, \delta)\)-differentially private algorithms is \((\epsilon', k\delta + \delta')\)-differentially private for

\[ \epsilon' = \sqrt{2k \ln(1/\delta')} \cdot \epsilon + k\epsilon(\epsilon - 1). \]

Furthermore, if \(\epsilon' \in (0, 1)\) and \(\delta' > 0\), the composition of \(k\) \(\epsilon\)-differentially private mechanism is \((\epsilon', \delta')\)-differentially private for

\[ \epsilon' = \epsilon \cdot \sqrt{8k \log(\frac{1}{\delta'})}. \]

The following proposition follows from strong composition theorem.

**Proposition 2.7 (Corollary 3.21 in [DR+14]).** Let \(A_1, \ldots, A_k\) be \(k(\epsilon', \delta')\)-differentially private algorithm for \(\epsilon' = \epsilon/\sqrt{8k \log(1/\delta')}\). Then an algorithm \(A\) formed by adaptive composition of \(A_1, \ldots, A_k\) is \((\epsilon, k\delta' + \delta)\)-differentially private.

3 **Tighter Analysis for Sublinear-Error DP Shortest Distances Algorithms**

In this section, we show that, by a tighter analysis, previous algorithms \([CGK+23, FLL22]\) output private distances with additive error up to \(O(n^{2/3})\) for \(\epsilon\)-DP and \(O(n^{1/3})\) for \((\epsilon, \delta)\)-DP. We can achieve this by applying a lemma from [Wai19] to the exact same framework as in \([CGK+23, FLL22]\). On a high level, such an improvement is made possible by applying concentration bounds on the sum of Laplace random variables. Concretely, the analysis of all previous work simply takes \(t\) times the Laplace variance on the shortest path. However, we observe that the additive errors on the shortest paths follow the tail bound of the sum of sub-exponential random variables. As such, without any modification of the algorithm, we are able to obtain improved bounds for both \(\epsilon\)-DP and \((\epsilon, \delta)\)-DP.

For completeness, we first present a version of the \(\epsilon\)-DP and \((\epsilon, \delta)\)-DP algorithms in \([CGK+23, FLL22]\) and give a self-contained analysis (details are in Appendix B). Here and throughout, we call these algorithms the landmark shortcut algorithms and use this as the base case result to be improved.

3.1 **The Landmark Shortcut Algorithms**

We start with an observation that the DP algorithms proposed in both the recent papers \([CGK+23, FLL22]\) follow the same high-level idea: they use a random subset of vertices (call it \(S \subseteq V\)) to construct private shortest distances between all pairs of vertices in \(S\), and then add edge-level noise to all edges in the graph. As such, for each vertex pair \((u, v)\) with the shortest path \(P(u, v)\), the algorithms release the shortest distance, \(\hat{d}(u, v)\), as the summation of the following three quantities:

1. **noisy edge weights**, along \(P(u, v)\), to the first vertex of \(P(u, v)\) in \(S\),
2. **shortest distances** from the first vertex of \(P(u, v)\) in \(S\) to the last vertex of \(P(u, v)\) in \(S\), and
3. **noisy edge weights** from the last vertex of \(P(u, v)\) in \(S\) to \(v\).

Concretely, the scheme can be describe as\(^3\): (i) add Laplace noise of \(\text{Lap}(O(\text{poly}(|S|)))\) to all shortest distances between vertices in \(S\), and (ii) add Laplace noise of \(\text{Lap}(O(1))\) on all edges. (We omit the dependency on \(\epsilon\) here for simplicity.) By a careful choice of the sample size \(s\), we can balance the parameters and obtain the optimal bound for \(\epsilon\)-DP and \((\epsilon, \delta)\)-DP, respectively. For simplicity, we follow the practice from Chen et al. \([CGK+23]\) and name step (i) as output perturbation and and step (ii) as input perturbation.

We now formalize the above high-level strategy as the following meta-algorithm (which will be realized in \(\epsilon\)-DP and \((\epsilon, \delta)\)-DP under different parameters).

---

\(^3\)This is a simplification of the algorithms in \([CGK+23, FLL22]\) as they use slightly more complicated private mechanisms. The mechanisms allow the algorithm to shave off errors terms logarithm in \(n\); however, it is unclear how to improve the exponent by using the black-box mechanism.
Parameter Set 1:

- Laplace parameters for \( \varepsilon \text{-DP} \):
  \[ \zeta = n^{1/3}, \quad s = 100 \cdot \zeta \cdot \log n, \quad \epsilon^0 = \frac{\zeta^2}{\varepsilon}, \quad \epsilon^1 = \frac{2}{\varepsilon}. \]

- Laplace parameters for \((\varepsilon, \delta)\text{-DP}\):
  \[ \zeta = (n/\log \frac{1}{\delta})^{1/2}, \quad s = 100 \cdot \zeta \cdot \log n, \quad \epsilon^0 = \frac{4s}{\varepsilon} \cdot \log \frac{2}{\delta}, \quad \epsilon^1 = \frac{2}{\varepsilon}. \]

We now use the parameters of Parameter Set 1 to give the landmark shortcut scheme, which can be realized as both \( \varepsilon \text{-DP} \) and \((\varepsilon, \delta)\text{-DP}\) algorithms.

The Landmark Shortcut Scheme – A meta-algorithm

**Input:** Weighted graph \( \mathcal{G} = (\mathcal{V}, \mathcal{E}, w) \) and privacy budget \( \varepsilon, \delta \in (0, 1) \).

1. Sample a set \( S \subseteq \mathcal{V} \) of \( s = 100 \cdot \zeta \cdot \log n \) vertices uniformly at random.
2. Compute shortest paths for every vertex pair \((x, z) \in S\).
3. **Output perturbation:** For \( x, z \in S \), let the shortest path length between \( x, z \) be \( d(x, z) \). We add independent Laplace noise \( \text{Lap}(\epsilon^0_x) \) (resp. \( \text{Lap}(\epsilon^0_z) \)) to \( d(x, z) \), and let the resulting distances be \( d_S(x, z) \).
4. **Input perturbation:** Add independent Laplace noise \( \text{Lap}(\epsilon^1) \) (resp. \( \text{Lap}(\epsilon^1_z) \)) to the weight of edge \( e \) in graph \( \mathcal{G} \). For any vertices \( u, v \in \mathcal{V} \), let \( P(u, v) \) be the shortest path in \( \mathcal{G} \) and \( d'(u, v) \) be the sum of the noisy weights of the edges along \( P(u, v) \).
5. For each pair of vertices \((u, v) \in \mathcal{V} \times \mathcal{V}\),
   - If there are at least two vertices along \( P(u, v) \cap S \), take \( x \) as the first vertex and \( v \) as the last one in \( P(u, v) \cap S \), compute the private distance \( \hat{d}(u, v) = d'(u, x) + d_S(x, z) + d'(z, v) \).
   - Otherwise, compute the private distance \( \hat{d}(u, v) = d'(u, v) \).
   - **Output** \( \hat{d}(u, v) \)

The formal guarantees of the meta-algorithm under both \( \varepsilon \text{-DP} \) and \((\varepsilon, \delta)\text{-DP}\) regimes (as captured by [CGK+23, FLL22]) are as follows.

**Proposition 3.1** ([CGK+23, FLL22]). The Landmark Shortcut Scheme satisfies the following guarantees.

- With parameters \( \epsilon^0 \) and \( \epsilon^1 \), the algorithm is \( \varepsilon \)-differentially private, and the additive error is \( \tilde{O} \left( \frac{n^{2/3}}{\varepsilon} \right) \) with high probability, i.e.,
  \[ \Pr \left( \max_{u,v \in \mathcal{V}} |\hat{d}(u, v) - d(u, v)| = \tilde{O} \left( \frac{n^{2/3}}{\varepsilon} \right) \right) \geq 1 - \frac{1}{n}. \]

- With parameters \( \epsilon^0_{\varepsilon, \delta} \) and \( \epsilon^1_{\varepsilon, \delta} \), the algorithm is \((\varepsilon, \delta)\)-approximate differentially private, and is \( \tilde{O} \left( \frac{n^{1/2}}{\varepsilon} \cdot \log^{1/2} \left( \frac{1}{\delta} \right) \right) \) with high probability, i.e.,
  \[ \Pr \left( \max_{u,v \in \mathcal{V}} |\hat{d}(u, v) - d(u, v)| = \tilde{O} \left( \frac{n^{1/2}}{\varepsilon} \cdot \log^{1/2} \left( \frac{1}{\delta} \right) \right) \right) \geq 1 - \frac{1}{n}. \]

We provide a self-contained proof of Proposition 3.1 to Appendix B.1. For now, we present a structural lemma of the Landmark Shortcut Scheme which we will use extensively in the rest of the paper.
Theorem 1. With high probability, the Landmark Shortcut Scheme satisfies the following guarantees.

1. Suppose that the number of edges on the shortest path $P(u,v)$, denoted by $|P(u,v)|$, is at most $\frac{n}{\xi}$, then, with high probability, there exist at least two vertices $(x, z) \in P(u,v)$ such that $x \in S$ and $z \in S$.

2. Suppose without any loss of generality, $|P(u,x)| \leq |P(u,z)|$, then the numbers of edges from $u$ to $x$ and from $z$ to $v$ are at most $\frac{n}{\xi}$, i.e. $|P(u,x)| \leq \frac{n}{\xi}$ and $|P(z,v)| \leq \frac{n}{\xi}$.

As a result of Lemma 3.2, we can now decompose the shortest distances in $G$ and bound the Laplace noise as at most a single output perturbation variance and at most $\tilde{O}(n/\xi)$ input perturbation variances. The formal statement of such a decomposition is as follows.

Lemma 3.3. With high probability, for any vertex pair $(u, v) \in V$, the difference between $\hat{d}(u, v)$ and $d(u, v)$ is at most $\left(\frac{2n}{\xi} \cdot e^\epsilon_i + e^\epsilon_o\right) \cdot \log n$ with the $\varepsilon$-DP parameters and $(\frac{2n}{\xi} \cdot e^\epsilon_i + e^\epsilon_o) \cdot \log n$ with the $(\varepsilon, \delta)$-DP parameters.

The proofs of Lemma 3.2 and Lemma 3.3 can be found in Appendix B. Lemma 3.3 provides a way to bound the total additive error as the summation of the Laplace noises in the worst case. The error bounds obtained as such match the work of [CGK+23, FLL22]. A self-contained analysis of this error (up to logarithm factor) is presented in Appendix B.

In the following, we present a tighter analysis using tail bounds [Wai19] on the sum of Laplace random variables, showing that by balancing the parameters, we can achieve the error of $\tilde{O}(n^{2/5})$ for $\varepsilon$-DP and $\tilde{O}(n^{1/3})$ for $(\varepsilon, \delta)$-DP.

3.2 Tighter Bounds on The Landmark Shortcut Algorithms

We present a tighter analysis for the landmark shortcut algorithms as in Section 3.1.

Theorem 1. With high probability, the Landmark Shortcut Scheme satisfies the following guarantees.

- With parameters $e^\epsilon_i$ and $e^\epsilon_o$, the algorithm is $\varepsilon$-differentially private, and the additive error is at most $\tilde{O}(\frac{n^{2/5}}{\varepsilon})$ with high probability, i.e.
  \[
  \Pr\left(\max_{u,v \in V} |\hat{d}(u,v) - d(u,v)| = \tilde{O}\left(\frac{n^{2/5}}{\varepsilon}\right)\right) \geq 1 - \frac{1}{n}.
  \]

- With parameters $e^\epsilon_i, e^\epsilon_o, e^\epsilon_i, e^\epsilon_o$ and $e^\epsilon_o$, the algorithm is $(\varepsilon, \delta)$-approximate differentially private, and the additive error is at most $\tilde{O}\left(\frac{n^{1/3}}{\varepsilon} \log^{1/3} \left(\frac{1}{\delta}\right)\right)$ with high probability, i.e.
  \[
  \Pr\left(\max_{u,v \in V} |\hat{d}(u,v) - d(u,v)| = \tilde{O}\left(\frac{n^{1/3}}{\varepsilon} \cdot \log^{1/3} \left(\frac{1}{\delta}\right)\right)\right) \geq 1 - \frac{1}{n}.
  \]

Theorem 1 crucially relies on the sum-of-Laplace tail bound in Lemma 2.1 – this allows us to apply concentration bound on the total of the Laplace noises. As such, the privacy analysis still holds as before, and the additive error becomes asymptotically smaller. This intuition is formalized as the following lemma.

Lemma 3.4 (Tighter version of Lemma 3.3). Let the noise parameters $e^\epsilon_i, e^\epsilon_o, e^\epsilon_i, e^\epsilon_o$ be with the formula as prescribed in Parameter Set 1. For sufficiently large absolute constant $C$, with high probability, for any vertex pair $(u, v) \in V$, the difference between $\hat{d}(u,v)$ and $d(u,v)$ is

1. at most $O\left(\sqrt{\frac{n}{\xi} \cdot (e^\epsilon_i)^2 + (e^\epsilon_o)^2} \cdot \log n\right)$ with the $\varepsilon$-DP parameters when $n \geq C \cdot \xi^5 \log n$;
2. at most $O\left(\sqrt{\frac{n}{\xi} \cdot (e_{\epsilon,\delta})^2 + (e_{\delta})^2} \cdot \log n\right)$ with the $(\epsilon, \delta)$-DP parameters when $n \geq C \cdot \xi^3 \log^2 n \log \frac{1}{\delta}$.

The proof of Lemma 3.4 is our first application of the concentration of Laplace random variables. Since it is very similar to the proofs in Section 4, we defer the details to Appendix B.2. For now, let us first prove Theorem 1 with Lemma 3.4.

**Proof of Theorem 1.** The privacy guarantee follows from the same analysis as in Proposition 3.1 (Lemma B.1 and Lemma B.3) since the algorithms are unchanged. For the additive error, we condition on the high probability event of Lemma 3.4 and analyze the two algorithms as follows.

1. For the $\epsilon$-DP algorithm, we can pick $\xi = O\left(\frac{n}{\log(n)}\right)^{1/5}$ to satisfy the condition prescribed in Lemma 3.4. As such, the additive error is

$$O\left(\sqrt{\frac{n}{\xi} \cdot (e_{\epsilon})^2} \cdot \log n\right) = \tilde{O}\left(\frac{n^{2/5}}{\epsilon}\right).$$

2. For the $(\epsilon, \delta)$-DP algorithm, we pick $\xi = O\left(\frac{n}{\log^2(n) \log \frac{1}{\delta}}\right)^{1/3}$ to satisfy the condition prescribed in Lemma 3.4. As such, the additive error is

$$O\left(\sqrt{\frac{n}{\xi} \cdot (e_{\epsilon,\delta})^2} \cdot \log n\right) = \tilde{O}\left(\frac{n^{1/3}}{\epsilon} \cdot \log^{1/3} \frac{1}{\delta}\right).$$

The error bounds are as claimed, concluding the proof. 

---

### 4 Improved Algorithms

To break the barrier of the algorithms in Section 3, we design two new algorithms in this section, which give additive errors of $\tilde{O}(n^{1/3})$ for $\epsilon$-DP and $\tilde{O}(n^{1/4})$ for $(\epsilon, \delta)$-DP, respectively. Formally, we state our results for pure-DP and approximate-DP as follows.

**Theorem 2.** Given a graph $G = (V, E, w)$ with a known edge topology. Then

1. For any $\epsilon \geq 0$, there exists an efficient algorithm that is $\epsilon$-differentially private, and with high probability, outputs all-pair shortest distances with additive error $O\left(\frac{n^{1/3} \log^{5/6} n}{\epsilon}\right)$. That is, the algorithm outputs an estimate $\hat{d}(u, v)$ such that

$$\Pr\left(\max_{u,v \in V} |\hat{d}(u, v) - d(u, v)| = O\left(\frac{n^{1/3} \log^{5/6} n}{\epsilon}\right)\right) \geq 1 - \frac{1}{n}. \quad \text{(Pure DP)}$$

2. For any $\epsilon, \delta \geq 0$, there exists an efficient algorithm that is $(\epsilon, \delta)$-differentially private, and with high probability, outputs all-pair shortest distances with additive error $O\left(\frac{n^{1/4} \log^{2/3} n \log^{1/4} 1/\delta}{\epsilon}\right)$. That is, the algorithm outputs an estimate $\hat{d}(u, v)$ such that

$$\Pr\left(\max_{u,v \in V} |\hat{d}(u, v) - d(u, v)| = O\left(\frac{n^{1/4} \log^{5/4} n \log^{1/2} 1/\delta}{\epsilon}\right)\right) \geq 1 - \frac{1}{n}. \quad \text{(Approximate DP)}$$

We prove Theorem 2 by the explicit construction and analysis of the algorithms in Section 4.1 and Section 4.2. As clarified in Remark 1, the algorithms using single-source shortest-path tree scheme can achieve pure and approximate-DP regime using only different parameters, just as the Landmark Shortcut algorithms. However, we propose a different idea for pure-DP algorithm, which shaves off a $\log^2 n$ factor.
Figure 1: (i) Two shortest paths $P(u, v)$ and $P(x, z)$, $u, v, x, z \in S$, intersect at a common subpath as the shortest path between two cut vertices $w, w'$. (ii) The shortest path $P(u, v)$ is partitioned into canonical segments $P(u, w_1), P(w_1, w_2), \ldots, P(w_{\ell}, w_{\ell+1}), P(w_{\ell+1}, v)$, where $w_1, w_2, \ldots, w_{\ell+1}$ are (ordered) cut vertices along path $P(u, v)$.

4.1 Algorithm for $\varepsilon$-DP

In the landmark shortcut algorithm for $\varepsilon$-DP, the weight change of a single edge may trigger the change of potentially every pair of shortest distances of vertices in $S$. As such, by the composition theorem, we need to boost the privacy parameter by a factor of $|S|^2$ since each shortest distance can change by 1. On the other hand, shortest paths have special structures. Since we assume without loss of generality that shortest paths are unique, two shortest paths only overlap by one common shortest path segment. Therefore instead of using output perturbation on the shortest path distances among vertices in $S$, we will be better off by decomposing the shortest paths by how they overlap and privatize the decomposed segments. As will become evident, the size of decomposed segments is less than $|S|^2$, hence the cumulative error is reduced.

To formalize the above intuition, we introduce the notion of cut vertices and canonical segments. Both notions are defined w.r.t a subset of vertices $S \subseteq V$. Informally, a vertex $w$ becomes a cut vertex if it is a vertex of $S$, or if it witnesses the branching – either ‘merging’ or ‘splitting’ – of two shortest paths between different pairs of vertices in $S$. The formal definition is as follows.

**Definition 9** (Cut Vertices). Let $S \subseteq V$ be an arbitrary subset of vertices. For any pair of vertices $(u, v) \in S$ and their shortest path $P(u, v)$, we say $w \in P(u, v)$ is a cut vertex for $(u, v)$ if it satisfies one of the following two conditions:

1. $w \in \{u, v\}$;
2. $w \not\in \{u, v\}$ and

   (a) $w \in P(x, z)$ for some $(x, z) \in S$ and $(x, z) \neq (u, v)$;

   (b) Without any loss of generality, suppose the path is from $x$. Let $\text{pred}(w)$ be the vertex before $w$ on $P(x, z)$ and $\text{succ}(w)$ be the vertex after $w$ on $P(x, z)$. Then either $\text{pred}(w) \not\in P(u, v)$ or $\text{succ}(w) \not\in P(u, v)$.

See Figure 1 (i) for an illustration of cut vertices. Based on Definition 9, we can now define the canonical segments as the path between two adjacent cut vertices along shortest paths of vertices in $S$.

**Definition 10** (Canonical Segments). Let $S \subseteq V$ be an arbitrary subset of vertices. For any pair of vertices $(u, v) \in S$ and their shortest path $P(u, v)$, a subpath $P(w, w')$ of $P(u, v)$ is a canonical segment if

1. $w$ is a cut vertex for some $(x, z) \in S$;
2. $w'$ is a cut vertex for some $(x', z') \in S$;
3. None of the vertices between $w$ and $w'$ on $P(u, v)$ is a cut vertex for any $(x'', z'') \in S$.

Note that $(u, v)$, $(x, z)$, and $(x', z')$ may or may not be the same in the above definition. One can think of the cut vertices as all vertices that witnesses the shortest path branching between all pairs of vertices in $S$, and the canonical segments are exactly the collection of segments between adjacent cut vertices along
shortest paths of vertices in $S$. See Figure 1 (ii) for an example: $\{u, v, w_1, w_2, w_3, w_4\}$ are all cut vertices, which define 5 canonical segments.

For a fixed vertex pair $(u, v) \in S$, we define $\text{Canon}(S, u, v)$ as the set of canonical segments on the shortest path of $(u, v)$. Note that the canonical segments need not to be among the edges between the vertices in $S$: the shortest path between $(u, v) \in S$ may well be outside of $S$. We provide some observations about the basic properties of canonical segments.

**Observation 4.1.** Canonical segments defined as in Definition 10 satisfy the following properties:

1. Any two canonical segments are disjoint.
2. The segments in $\text{Canon}(S, u, v)$ covers all edges in $P(u, v)$, i.e. $P(u, v) = \cup_{z} \text{Canon}(S, u, v) P(x, z)$.  
3. For any pair of vertices $(u, v) \in S$, there are at most $|S|^2$ canonical segments in $\text{Canon}(S, u, v)$ for $|S| \geq 2$.

**Proof.** Observation 1 is by definition. Concretely, if two canonical segments overlap, there must be one cut vertex inside another canonical segment, which is not possible by definition. Observation 2 follows from the fact that $u$ and $v$ themselves are cut vertices, and any other cut vertices on $P(u, v)$ only further divides the path. Finally, observation 3 holds since every pair of vertices in $S$ contributes to at most two cut vertices on $P(u, v)$. Thus there are at most $2 \cdot (|S|)^2 \leq |S|^2$ canonical segments.

With the definition and properties of canonical segments, we are now ready to present our $\varepsilon$-DP algorithm as follows.

| **Canon-APSD:**| An $\varepsilon$-DP algorithm to release all pair shortest distances |
|---------------|----------------------------------------------------------|
| **Input:** | An $n$ vertices graph, $G = (V, E, w)$ and privacy parameter $\varepsilon > 0$. |
| 1. | Sample a set $S$ of $s = 100 \zeta \cdot \log n$ vertices uniformly at random, where $\zeta = O(n^{1/3} \log^{-2/3} n)$ |
| 2. | Compute all-pair shortest path for every vertex pair $(x, z) \in G$, and let $P_S$ be the set of the paths. |
| 3. | Compute $\text{Canon}(S)$ based on the sampled vertices $S$ and their shortest paths $P_S$. |
| 4. | **$S$ Perturbation:** For each canonical segment $P \in \text{Canon}(S, u, v)$, add an independent Laplace noise $\text{Lap}(2/\varepsilon)$ to its shortest path length. Compute a function $d_S(\cdot, \cdot)$ for shortest distances between any vertices $(u, v) \in S$, by summing up the noisy shortest path distances of the canonical segments in $\text{Canon}(S, u, v)$. |
| 5. | **Non-$S$ Perturbation:** For each edge in $G$, add independent Laplace noise $\text{Lap}(2/\varepsilon)$ to the edge weight. For any vertices $u, v \in V$, let $P(u, v)$ be the shortest path in $G$ and $d'(u, v)$ be the sum of the noisy weights of the edges along $P(u, v)$. |
| 6. | For each pair of vertices $(u, v)$, |
| 6.1. | If there are at least two vertices in $P(u, v)$ that are in $S$, let vertex $x$ be the first one along $P(u, v)$ and $z$ be the last one such that $x, z \in S$, release $\tilde{d}(u, v) = d'(u, x) + d_S(x, z) + d'(z, v)$. |
| 6.2. | Otherwise, release $\tilde{d}(u, v) = d'(u, v)$. |

We now give the formal analysis of the privacy guarantee and bounds for the additive error.

**Proof of statement 1 in Theorem 2**

We start with an observation of the sensitivity of canonical segments. Since canonical segments do not overlap, the weight change of a single edge can only trigger changes of the shortest path distances of at most one canonical segment.
Claim 4.2. Fix any $S \subseteq V$, and let $g : (2^V, 2^E) \rightarrow \mathbb{R}^{\text{Canon}(S)}$ be the function that computes the distances for canonical segments. Then, the $\ell_1$ sensitivities for $g$ is at most 1.

Proof. The claim follows from the fact that the canonical segments are disjoint (statement 1 of Observation 4.1). Concretely, recall that for two neighboring graphs $G \sim G' \in \mathcal{X}$, we have

$$\sum_{e \in E} |w(e) - w'(e)| \leq 1.$$ 

As such,

$$\Delta_{g,1} = \max_{w,w' \in \mathcal{X}, w \sim w'} \|g(w) - g(w')\|_1 \leq \max_{w,w' \in \mathcal{X}, w \sim w'} \|w - w'\|_1 \leq 1,$$

where the first inequality follows from the disjointness of canonical segments and the second inequality is by the neighboring graphs. \hfill \Box

Notably, Claim 4.2 is already sufficient for us to prove the privacy of the algorithm.

Lemma 4.3. The CANON-APSD algorithm is $\epsilon$-differentially private.

Proof. By Claim 4.2, the functions in steps 4 is of $\ell_1$ sensitivity at most 1. As such, by Proposition 2.3, its output is $\frac{\epsilon}{2}$-DP. Furthermore, we note that the privacy analysis for Line 5 is exactly the same as in Lemma B.1, i.e. the weight function has sensitivity at most 1. Combining the above results and the basic composition theorem Proposition 2.5 gives the desired result. \hfill \Box

We now proceed to bounding the additive error, which follows the same idea as in Theorem 1: we decompose the noise into different parts, and use the concentration of Laplace distribution to get the tight bound.

Lemma 4.4. With high probability, for any vertex pair $(u,v) \in V$, the difference between $d(u,v)$ and $\hat{d}(u,v)$ released by CANON-APSD is at most $O\left(\frac{1}{\epsilon} \sqrt{n/\zeta + \epsilon^2 \log^2 n} \cdot \log n\right)$. More precisely,

$$|d(u,v) - \hat{d}(u,v)| \leq \frac{900}{\epsilon} \cdot \sqrt{n/\zeta + \epsilon^2 \log^2 n} \cdot \log n$$

for any $n \geq C \cdot \zeta \log n$ where $C$ is a sufficiently large absolute constant.

Proof. Like the proof of Lemma 3.4, we once again condition on the high probability event of Lemma 3.2, and decompose the error into different parts to apply the concentration inequality of Laplace noise. Fix a pair of vertices $(u,v) \in V$ and their shortest path $P(u,v)$, the additive noises are:

1. At most $\frac{2n}{\zeta}$ independent noises sampled from $\text{Lap} \left( \frac{\zeta}{2} \right)$.

2. At most $s^2 = 100^2 \cdot \zeta^2 \cdot \log^2 n$ independent noises sampled from $\text{Lap} \left( \frac{\zeta}{2} \right)$ for the canonical segments.

The second line is obtained from statements 2 and 3 of Observation 4.1: to compute the all-pairs shortest distances between pair in $S$, it suffices to estimate the canonical segments, and there are at most $s^2 \geq 30 \sqrt{\log n}$ of them. As such, in the CANON-APSD algorithm, we let each Laplace noise be with variance $b_i = 2/\epsilon$ for all $i$, we again pick $v = \sqrt{\sum b_i^2}$ and $\lambda = 30v \sqrt{\log n} = \frac{60}{\epsilon} \cdot \sqrt{n \log n}$. Recall that $s = 100 \log n \cdot n^{1/3}$ (since $\zeta = n^{1/3}$), which implies $\frac{2v\zeta}{\max_i b_i} \geq 30 \sqrt{\log n}$ (this only needs $n \geq C \cdot \zeta \log n$ for some constant $C$). Therefore, we can apply the concentration of Laplace tail in Lemma 2.1, which gives us

$$\Pr \left[ |d(u,v) - \hat{d}(u,v)| \geq 30 \sqrt{\log n} \right] \leq 2 \exp \left( -\frac{900 \log n}{8} \right) \leq \frac{1}{n^3}.$$
Therefore, with high probability, the additive error between \( d(u, v) \) and \( \hat{d}(u, v) \) is at most
\[
30 \sqrt{\log n} \cdot \nu \leq \frac{90}{\epsilon} \sqrt{\left( \frac{n}{\zeta} + 100^2 \cdot \nu^2 \cdot \log^2 n \right) \cdot \log n}.
\]
A union bound over the above event and the high probability event in Lemma 3.2 gives us the desired statement. \( \square \)

In fact, Lemma 4.4 holds for any \( \zeta = n^{1-O(1)} \) for sufficiently large \( n \) (as long as \( n^{O(1)} > 900 \log(n) \)). We can now finalize the analysis of the additive error of the CANON-APSD algorithm.

**Lemma 4.5.** With high probability, the CANON-APSD algorithm has an additive error of at most \( O \left( \frac{n^{1/3}}{\epsilon} \cdot \log^{5/6} n \right) \).

**Proof.** We use Lemma 4.4 by setting the parameter \( \zeta = \frac{1}{\epsilon} \cdot n^{1/3} \log^{-2/3} n \) with the \( C \) in Lemma 4.4. As such, the total additive error becomes
\[
O \left( \frac{1}{\epsilon} \cdot \sqrt{\left( \frac{n}{n^{1/3} \log^{-2/3} n} + (n^{1/3} \log^{-2/3} n)^2 \cdot \log^2 n \right) \cdot \log n} \right) = O \left( \frac{n^{1/3}}{\epsilon} \cdot \log^{5/6} n \right),
\]
as claimed. \( \square \)

This completes the proof of item 1 in Theorem 2.

### 4.2 Algorithm for \((\epsilon, \delta)-\text{DP}\)

We proceed to the algorithm for \((\epsilon, \delta)-\text{DP}\) as stated in Theorem 2. Our algorithm builds single-source shortest path trees (see formal definition in Definition 11) for each sampled vertex, then employs an \((\epsilon, \delta)-\text{DP}\) algorithm for distances release in tree graph. Notice that the construction of single-source shortest-path trees follows from folklore algorithms based on Dijkstra’s algorithm, which takes \( O(m \log n + m) \) time with the classical Fibonacci heap implementation.

**Definition 11** (Single-source shortest-path tree). Given a graph \( G = (V, E) \) and a vertex \( s \in V \), the single-source shortest-path tree rooted at \( s \) is a spanning tree \( G’ \) such that the unique path from \( s \) to \( v \) in \( G’ \) is the shortest path from \( s \) to \( v \) in \( G \).

We will use the following result of the \((\epsilon, \delta)-\text{DP}\) algorithm for tree graphs shown rigorously in Appendix C.

**Lemma 4.6** ((\(\epsilon, \delta\))-DP for tree graph). Given a tree graph \( G = (V, E, w) \) and privacy parameter \( \epsilon, \delta \in (0, 1) \), there exists an \((\epsilon, \delta)-\text{DP}\) algorithm releasing shortest distances from the root vertex to the rest such that, with high probability, induce additive error at most \( O \left( \frac{1}{\epsilon} \log^{1.5} n \sqrt{\log(\frac{1}{\delta})} \right) \).

We have three remarks for Lemma 4.6. First, prior work ([Sea16, FL22]) focused on \( \epsilon\)-DP. Between [Sea16, FL22], Fan and Li’s algorithm [FL22] uses heavy-light decomposition of the tree, with a better error bound only when the tree is shallow. Thus we present the version of \((\epsilon, \delta)-\text{DP}\) based on Sealfon’s algorithm [Sea16]. Second, Sealfon’s algorithm exploits Laplace mechanism, which is replaced by Gaussian mechanism with \( \sigma^2 := 1/\epsilon^2 \cdot \ln(1.25/\delta) \log n \) in Lemma 4.6. Third, the additive error bound for \( \epsilon\)-DP on tree graph is \( O \left( \frac{1}{\epsilon} \log^{2.5} n \right) \) with high probability for single-source distance. **Lemma 4.6** implies that the \((\epsilon, \delta)-\text{DP}\) algorithm can shave off a \( \log n \) factor, end up with a quadratic improvement on the logarithm term in the final algorithm for private all pairs shortest distances.

For simplicity, call this algorithm PrivateTree \( (G) \) with an input tree graph \( G \). Also we use SSSP \((v)\) for the single-source shortest path tree algorithm, which takes any \( v \in V \) as input and outputs a shortest path tree with \( v \) as the root. The \((\epsilon, \delta)-\text{DP}\) algorithm is presented as follows.
SSSP-APSD: An $\epsilon, \delta$-DP algorithm to release all pair shortest distances

**Input:** An $n$ vertices graph, $G = (V, E, w)$ and privacy parameter $\epsilon, \delta > 0$.

1. Sample a set $S$ of $s = \zeta \cdot \log n$ vertices uniformly at random, where $\zeta = O(\sqrt{n} \log^{2.5} n)$.

2. For each vertex $v \in S$, compute $T(v) = \text{SSSP}(v)$. Call the set of all trees $T$.

3. **$S$ Perturbation:** For each tree $T \in T$, privatize it by running $\text{PrivateTree}(T)$ with the Gaussian noise $\mathcal{N}(\mu = 0, \sigma^2 := \frac{1}{\xi_0} \ln(1.25/\delta_0) \log n)$, $\epsilon_0, \delta_0$ will be specified later, let the output distance be $d_T(u, v)$.

4. **Non-$S$ Perturbation:** For each edge in $G$ add independent Gaussian noise $\mathcal{N}(\mu = 0, \sigma^2 := \frac{4}{\xi_0^2} \ln(2.5/\delta) \log n)$. For any vertices $u, v \in V$, let $P(u, v)$ be the shortest path in $G$ and $d'(u, v)$ be the sum of the noisy weights of the edges along $P(u, v)$.

5. For each pair of vertices $(u, v)$
   - If at least one of $u, v$ is in $S$, release $\hat{d}(u, v) = d_T(u, v)$.
   - If $u, v \notin S$ and the path $P(u, v)$ has one vertex $x \in S$, release $\hat{d}(u, v) = d_T(u, x) + d_T(x, v)$.
   - Otherwise, release $\hat{d}(u, v) = d'(u, v)$.

**Proof of approximate-DP result (statement 2) in Theorem 2**

Our analysis mainly hinges on the concentration of Laplace random variables (Lemma 2.1), a corollary (Proposition 2.7) of strong composition theorem (Proposition 2.6) and the observation that any shortest path with length larger than $\frac{\delta}{\xi}$ goes through at least one vertex in the sampled set $S$ with high probability (Lemma 3.2).

**Lemma 4.7.** The SSSP-APSD algorithm is $(\epsilon, \delta)$-differentially private.

**Proof.** First observe that any edge in $G$ can only appear in at most $s$ trees ($s = |S|$), since we only build one single-source shortest path tree for each vertex in $S$. Therefore, the PrivateTree algorithm (Lemma 4.6) is applied at most $s$ times to any edge. In $S$ perturbation, the Gaussian mechanism achieves $(\epsilon_0, \delta_0)$-DP for each tree. Pick $\epsilon_0, \delta_0$ such that $\epsilon_0 = \frac{\epsilon}{4\sqrt{2s \ln(4/\delta)}}$ and $\delta_0 = \frac{\delta}{4s}$, using a corollary of strong composition theorem (Proposition 2.7) on $s$ number of PrivateTree algorithms, we have that the $S$ perturbation is $(\epsilon/2, \delta/2)$-differentially private. Combining with the Non-$S$ perturbation, which is also $(\epsilon/2, \delta/2)$-differentially private, it is straightforward to see that the SSSP-APSD algorithm is $(\epsilon, \delta)$-differentially private.

The analysis of the additive error is again, similar as in Theorem 1 and Lemma 4.4. The only difference is that $s$ takes various values to balance the contribution from output perturbation and the input perturbation, leading to different additive errors.

**Lemma 4.8.** With high probability, the SSSP-APSD algorithm has additive error at most $O(\frac{n^{3/4}}{\epsilon} \cdot \log^{1.25} n \sqrt{\log \frac{1}{\delta}})$

**Proof.** We first show that with high probability, for any vertex pair $(u, v) \in V$, the difference between $d(u, v)$ and $\hat{d}(u, v)$ released by SSSP-APSD is at most $\max \left\{ O(\sqrt{n/\zeta} \log \frac{1}{\epsilon}), O\left(\sqrt{s \log \frac{2s}{\delta} \cdot \log^{1.5} n \log \frac{1}{\epsilon}}\right)\right\}$, only constants are omitted. Notice that the additive error is once again decomposed into noises from ‘output perturbation’ ($S$ perturbation) and ‘input perturbation’ (Non-$S$ perturbation). By Lemma 3.2 and Lemma 4.7, and fix a pair of vertices $(u, v) \in V$, denote their shortest path as $P(u, v)$, the additive noises must be either of the following two cases:

\[14\]
1. At most \( \frac{2n}{\xi} \) independent noises sampled from \( \mathcal{N}(\mu = 0, \sigma^2 := \frac{4}{\epsilon^2} \ln(2.5/\delta) \log n) \)

2. At most two independent noises induced by the PrivateTree algorithm, which is upper bounded by \( O\left( \frac{2}{n^2} \log^{1.5} n \sqrt{\log \frac{1}{\delta}} \right) \).

The first case considers the third bullet point in Step 5 of the SSSP-APSD algorithm, from Lemma 3.2 we know that the additive error is the summation of at most \( \frac{2n}{\xi} \) independent Gaussian noises. The second case considers the first and second points in Step 5 of the SSSP-APSD algorithm, where \( \hat{d}(u, v) \) is decomposed into two distances output by the PrivateTree algorithm. Notice that only one of the two cases can happen, hence the additive error bound is the maximum of the two. This is different from the analysis in Lemma 4.4, where the two cases are combined together to construct the shortest paths. In the following, we give detailed upper bounds of the additive error of two terms.

We now apply the concentration of Gaussian tail (Lemma 2.2) for the first case,

\[
\Pr \left[ \left| d(u, v) - \hat{d}(u, v) \right| \geq t \right] \leq 2 \exp \left( -\frac{t^2}{2n/\xi} \right),
\]

Let \( t = (n/\xi)^{1/4} \log^{0.5} n \cdot \delta \), the above probability is smaller than \( \frac{1}{n^4} \). Apply union bound on all vertex pairs, then with high probability, the additive error between \( d(u, v) \) and \( \hat{d}(u, v) \) for the first case is at most

\[
\ell = (n/\xi)^{1/2} \log^{0.5} n \cdot \delta = O\left( \frac{1}{\xi} (n/\xi)^{1/2} \sqrt{\log \frac{1}{\delta}} \right)
\]

Next, we show the additive error in the second case. In the \( S \) perturbation that we pick the privacy parameter \((\varepsilon_0, \delta_0)\) for the Gaussian mechanism where \( \varepsilon_0 = \frac{\varepsilon}{4\sqrt{2\ln(2\varepsilon)}} \) and \( \delta_0 = \frac{\delta}{25} \).

Recall Lemma 4.6, the additive error is at most

\[
\frac{1}{\varepsilon_0} \log^{1.5} n \sqrt{\log \frac{1}{\delta_0}} = O\left( \frac{1}{\xi} s \log\frac{1}{\delta} \cdot \log^{1.5} n \sqrt{\log \frac{2\varepsilon}{\delta}} \right)
\]

It only remains to balance the two terms to obtain the maximum additive error. Recall that \( s = O(\xi \cdot \log n) \), we pick \( \xi = C \sqrt{n} \log^{-2.5} n \), where \( C \) is a fixed constant, leading to the following additive error:

\[
O\left( \frac{1}{\xi} \sqrt{\frac{2n}{\xi} \cdot \log \frac{1}{\delta}} \right) = O(n^{1/4} \log^{1.25} n \cdot \sqrt{\log \frac{1}{\delta}})
\]

With Lemma 4.8 and Lemma 4.5, we complete the proof of Theorem 2.

## 5 Bounds for Path, Tree and Cycle Graphs

We present the upper and lower bounds for special graphs in this section. In particular, for paths, trees, and cycles, we obtained an upper bound for \( O(\log^{3/2} n) \) additive error and a lower bound of \( \Omega(\log n) \), both for \((\varepsilon, \delta)\)-DP. More formally, the result can be presented as the following theorem.

**Theorem 3.** Let \( G \) be either a line graph, cycle, or a tree on \( n \) vertices. Then there is an \((\varepsilon, \delta)\)-differentially private algorithm that, with probability \( 1 - \beta \), outputs all pair shortest path with \( \ell_\infty \) error

\[
O\left( \sigma_{\varepsilon, \delta} \ln n \sqrt{\frac{\ln n}{\beta}} \right).
\]
In contrast, for all sufficiently small constant $\epsilon$ and $\delta$, there is no $(\epsilon, \delta)$-differentially private algorithm that takes as input a line graph $G$ and outputs a vector $v \in \mathbb{R}^{\binom{n}{2}}$ such that
\[
\mathbb{E} \left[ \|v - d\|_\infty^2 \right] \leq o \left( \frac{1}{\epsilon^2} \ln^2 n \right),
\]
where $d \in \mathbb{R}^{\binom{n}{2}}$ is the vector representing the actual all pair shortest distance.

Previously, only pure-DP results are known for these special families of graphs ([Sea16, FL22]). In Lemma 4.6 we presented the approximate DP version for Sealfon’s algorithm [Sea16]. Unlike all these results, which rely on combinatorial algorithms, our bounds for special graphs are obtained by matrix mechanism. On the high level, matrix mechanism provides a way to analyze the privacy from the linear algebra perspective. In particular, for a query matrix $Q$ that encodes the all-pair distance queries, we can bound the additive error as a function of the completely bounded norm (see Definition 12 for details) of $Q$. While computing the completely bounded norm for general query matrices is hard, we can however obtain upper and lower bounds for special families of graphs, which is the main idea for our results.

In addition to the bounds for special graphs, we also note that many existing combinatorial algorithms can be interpreted as matrix mechanism with carefully-designed query matrices. This perspective can be of independent interests, and we derive the matrix mechanism form of two existing algorithm in Appendix D.

5.1 Query Matrix in the Matrix Mechanism

In this section, we prove the upper bound in Theorem 3. In particular, we show the following result:

**Lemma 5.1.** Let $G$ be either a line graph, cycle, or a tree on $n$ vertices. Then there is an $(\epsilon, \delta)$-differentially private algorithm that, with probability $1 - \beta$, outputs a vector $v \in \mathbb{R}^{\binom{n}{2}}$ such that, with probability $1 - \beta$ over the random coin tosses of the algorithm,
\[
\|v - d\|_\infty \in O \left( \sigma_\epsilon \delta \ln n \sqrt{\frac{\ln n}{\beta}} \right),
\]
where $d$ is the vector denoting the shortest distance between all pair of vertices in $G$.

This lemma can be shown by using the approximate differentially private counting algorithm as a subroutine in the algorithm of Sealfon [Sea16]. However, we use a linear algebraic approach in this section to facilitate our lower bound proof in the next section.

**Query Matrices.** We introduce the query matrix that is also essential in our proof of the lower bound. In particular, we can think of answering all pair shortest path in a linear algebraic manner as follows. Let $x \in \mathbb{R}^m$ denote the $m$-dimensional vector representing the edge weights of an $n$ nodes $m$ edges graph. Let $Q \in \{0, 1\}^{\binom{n}{2}} \times m$ be a binary matrix defined as follows:
\[
Q_{(u, v), e} = \begin{cases} 
1 & e \text{ is an edge on the shortest path } P(u, v) \\
0 & \text{ otherwise}
\end{cases}
\]
Note that the rows of the matrix $Q$ are indexed by a pair of vertices and columns corresponds to the edges in the same order as the vector $x$ represents those edges. With this linear algebraic view, we can compute all pair shortest distances by computing the vector $Qx$.

**Proof of Lemma 5.1.** We now use this approach to privately compute distances on line graph. A line graph with $n$ vertices indexed $\{1, 2, \cdots, n\}$ and edges $(i, i+1)$ for $1 \leq i \leq n-1$. The weights of the edges are considered sensitive information. To define the matrix $Q$ corresponding to the distance matrix for the line graph, we start by defining the following set of matrices
\[
Q_{\text{line}}(i)_{j,k} = \begin{cases} 
1 & j \geq k \\
0 & \text{ otherwise}
\end{cases} \in \{0, 1\}^{\binom{n-i}{2} \times (n-i)},
\]
(2)
where \( Q_{\text{line}}(i)_{j,k} \) is the \((j, k)\)-th entry of the matrix \( Q_{\text{line}}(i) \).

Now the query matrix is defined as follows:

\[
Q'_{\text{line}} = \begin{pmatrix}
Q_{\text{line}}(1) & 0 & \cdots & 0 \\
0 & Q_{\text{line}}(2) & \cdots & 0 \\
\vdots & \vdots & \ddots & \vdots \\
0 & 0 & \cdots & Q_{\text{line}}(n-1)
\end{pmatrix} \in \{0, 1\}^{\binom{n}{2} \times (n-1)}
\]

and the input being the histogram of edge weights. We can use this matrix to private output the all pair shortest distance; however, it is more convenient to work with a slight variation of the same problem. That is, we change the histogram to be the following:

\[
W = \begin{pmatrix}
E_{[1,n]} \\
E_{[2,n]} \\
\vdots \\
E_{[n-1,n]}
\end{pmatrix},
\]

where \( E_{[i,j]} \) denotes a vector of dimension \( j - i \) containing the weights on the edges in the set \([i, \cdots, j]\).

Then we can redefine the query matrix as follows:

\[
Q_{\text{line}} = \begin{pmatrix}
Q_{\text{line}}(1) & 0 & \cdots & 0 \\
0 & Q_{\text{line}}(2) & \cdots & 0 \\
\vdots & \vdots & \ddots & \vdots \\
0 & 0 & \cdots & Q_{\text{line}}(n-1)
\end{pmatrix} \in \{0, 1\}^{\binom{n}{2} \times (n-1)}.
\]

It is easy to verify that \( Q_{\text{line}} W \) is the vector corresponding to all pair shortest distance. We can compute a factorization for the above as follows\(^4\): Let’s define a function \( f : \mathbb{Z} \rightarrow \mathbb{R} \) as

\[
f(k) = \begin{cases}
1 & k = 0 \\
\frac{2^{k-1}}{2k} f(k-1) & k \geq 1
\end{cases}
\]

and define \( L(i) \) and \( R(i) \) be defined as follows:

\[
R(i)_{j,k} = L(i)_{j,k} = f(j - k),
\]

where \( L(i) \) and \( R(i) \) have the same dimension as \( Q_{\text{line}}(i) \).

Then we define the factorization as follows:

\[
L = \begin{pmatrix}
L(1) & 0 & \cdots & 0 \\
0 & L(2) & \cdots & 0 \\
\vdots & \vdots & \ddots & \vdots \\
0 & 0 & \cdots & L(n)
\end{pmatrix}
\]

and \( R = L \). The factorization mechanism would be then

\[
\mathcal{M}_{L,R}(Q_{\text{line}}, w) = L(Rw + z),
\]

where \( z \sim N(0, \|R\|_1^2 \sigma^2 \varepsilon_{\alpha, \delta}^{(n)}) \).

\(^4\)This function has been used to compute the factorization in Fichtenberger et al. [FHU22] and Henzinger et al. [HUU23] for continual counting.
Now we can compute $\|L\|_{2\rightarrow\infty}$ and $\|R\|_{1\rightarrow2}$ as in the case of Fichtenberger et al. [FHU22]:

$$\|L\|_{2\rightarrow\infty} = \|L(1)\|_{2\rightarrow\infty} = \sqrt{1 + \sum_{i=1}^{n-2} \left( \frac{1}{2} \frac{3}{4} \cdots \left( \frac{2i-1}{2i} \right) \right)^2}$$

$$\in \Theta \left( \sqrt{n} \right) \in \Theta \left( \sqrt{\int_{1}^{n} \frac{1}{x} \, dx} \right) \in \Theta \left( \sqrt{\log n} \right),$$

where the first inequality follows from Theorem 1 in Chen and Qi [CQ05] and second inequality comes from the trapezoid rule. Using the fact that the $\ell_\infty$ error of matrix mechanism is

$$O \left( \frac{1}{\epsilon} \|L\|_{2\rightarrow\infty} \|R\|_{1\rightarrow2} \sqrt{\ln(|Q|/\beta) \ln(1/\delta)} \right)$$

for the queries defined by the matrix $Q$ such that $Q = LR$, we get that the error in computing shortest distance in a line graph is

$$O \left( \frac{1}{\epsilon} \ln n \sqrt{\ln(n/\beta) \ln(1/\delta)} \right)$$

with probability $1 - \beta$.

As a result, we get the bound on computing the shortest distance on a line graph. We can do the same manipulations combined with Sealfon [Sea16] in the case of trees and get the same bound as above. For the cycle, we can do the same manipulation where $Q_{\text{line}}(i)$ is replaced by $Q_{\text{cycle}}(i)$ defined as below:

$$Q_{\text{cycle}}(i) = \begin{cases} 1 & j \geq k/2 \\ 0 & \text{otherwise} \end{cases} \in \{0,1\}^{n-i \times n-i}$$

and

$$Q_{\text{cycle}}(i) = \begin{pmatrix} Q_{\text{cycle}}(i) & 0 \\ 0 & Q_{\text{cycle}}(i)^\top \end{pmatrix}$$

Using the same calculation as above, we can compute the shortest distance with accuracy

$$O \left( \frac{1}{\epsilon} \ln n \sqrt{\ln(n/\beta) \ln(1/\delta)} \right)$$

with probability $1 - \beta$.

### 5.2 The Lower bound

We now proceed to the proof of our lower bound for special graphs.

**Lemma 5.2.** Let $\mathcal{G}$ be the set of line graphs and $(\epsilon, \delta)$ be a sufficiently small constant. Then, there for every $(\epsilon, \delta)$-differentially private algorithm, $\mathcal{M}$, if $y \leftarrow \mathcal{M}(\mathcal{G})$, there is

$$\max_{\mathcal{G} \in \mathcal{G}} \mathbb{E} \left[ \|y - Q_{\text{line}}x\|_\infty^2 \right] \in \Omega \left( \frac{\log^2 n}{\epsilon^2} \right),$$

where $x$ is the weight vector corresponding to the line graph $G \in \mathcal{G}$.

**Proof.** The following computation is done in [FHU22]. We include it here for the sake of completeness and because the updated manuscript is not publicly available.
Let us first consider the case when we are using an additive, data-independent mechanism (also known as data-oblivious noise mechanism [BDKT12]). That is, let
\[ \mathcal{M} = \{ M : M(x) = Q_{\text{line}} x + z \}, \]
where \( z \) is a random variable over \( \mathbb{R}^{|E|} \) whose distribution does not depend on the weight vector \( x \in \mathbb{R}^{|E|} \).

The proof follows similarly as in the mean-squared case in Edmonds et al. [ENU20]. Note that,
\[ \max_{x \in \{0,1\}^{|E|}} \mathbb{E} \left[ \| M(x) - Q_{\text{line}} x \|_2^2 \right] = \max_{x \in \{0,1\}^{|E|}} \mathbb{E} \left[ \| Q_{\text{line}} x + z - Q_{\text{line}} x \|_2^2 \right] = \mathbb{E} [\| z \|_2^2], \tag{3} \]
where the expectation is over the coin tosses of \( M \).

Let \( \Sigma = \mathbb{E}[zz^\top] \) be the covariance matrix of \( z \). Then
\[ \mathbb{E}[\| z \|_2^2] \geq \max_{1 \leq i \leq T} \Sigma_{i,i}. \tag{4} \]

Now let us define \( K = Q_{\text{line}} B_1^{|E|} \) to be the so called sensitivity polytope [NTZ13] and \( B_1 \) to be the \( |E| \)-dimensional \( \ell_1 \) unit ball. As \( Q_{\text{line}} \) has full rank, it follows that \( K \) is full dimensional. Now using Lemma 27 in Edmonds et al. [ENU20], we have that there exists an absolute constant \( C \) such that, for all sufficiently small constants \( \epsilon, \delta \), we have
\[ \max_{y \in K} \left\| \Sigma^{-1/2} y \right\|_2 \leq Ce. \]

Define \( L = \Sigma^{1/2} \) and \( R = \Sigma^{-1/2} Q_{\text{line}} \). Then
\[ \| R \|_{1\to 2} = \max_{1 \leq i \leq T} \left\| \Sigma^{-1/2} Q_{\text{line}} : i \right\|_2 \leq \max_{y \in K} \left\| \Sigma^{-1/2} y \right\|_2. \]

That is, \( \| R \|_{1\to 2} \leq Ce \). Further,
\[ \| L \|_{2\to \infty} = \max_{1 \leq i \leq T} (L^\top L)_{i,i} = \max_{1 \leq i \leq T} \Sigma_{i,i} \leq \mathbb{E}[\| z \|_2^2]. \]

By the definition of \( \| Q_{\text{line}} \|_{cb} \), we thus have
\[ \| Q_{\text{line}} \|_{cb} \leq \| L \|_{2\to \infty} \| R \|_{1\to 2} \leq C^2 \epsilon^2 \mathbb{E}[\| z \|_2^2]. \]

Using Lemma 6 in [FHU22] and adapting the proof in Mathias [Mat93], we have
\[ \| Q_{\text{line}} \|_{cb} \geq \Omega(\log n). \]

Rearranging the last two inequalities and plugging them into eq (3) gives us the lower bound
\[ \mathbb{E}[\| z \|_2^2] \in \Omega \left( \frac{\log^2 n}{\epsilon^2} \right). \tag{5} \]
as required when the mechanism is data-independent and additive.

To prove the result for general mechanisms, we use a result by Bhaskar et al. [BDKT12] which gives a black-box reduction for the data-independent noises to guarantee general privacy. In particular, the following result can be derived from Theorem 4.3 in Bhaskar et al. [BDKT12]:

**Proposition 5.3 (Bhaskar et al. [BDKT12]).** Let \( F \) be a linear query matrix. Consider an \((\epsilon, \delta)\)-differentially private mechanism \( M \) which has an (worst-case) expected error of \( \text{err}(M, F) \). Then there is a data-independent additive noise mechanism \( M' \) which is \((2\epsilon, \delta\epsilon)\)-differentially private mechanism and \( \text{err}(M', F) \leq \text{err}(M, F) \).

Now using Proposition 5.3 in conjunction with eq (5) and eq (3), we have the bounds for any \((\epsilon, \delta)\)-differentially private mechanism for sufficiently small \( \epsilon \) and \( \delta \). This completes the proof. \( \square \)
References

[BDKT12] Aditya Bhaskara, Daniel Dadush, Ravishankar Krishnaswamy, and Kunal Talwar. Unconditional differentially private mechanisms for linear queries. In Proceedings of the forty-fourth annual ACM Symposium on Theory of computing, pages 1269–1284, 2012. 19

[CGK+23] Justin Y. Chen, Badih Ghazi, Ravi Kumar, Pasin Manurangsi, Shyam Narayanan, Jelani Nelson, and Yinzhan Xu. Differentially private all-pairs shortest path distances: Improved algorithms and lower bounds. In 2023 Symposium on Discrete Algorithm (SODA 2023), 2023. 1, 2, 6, 7, 8, 23, 27

[CQ05] Chao-Ping Chen and Feng Qi. The best bounds in wallis’ inequality. Proceedings of the American Mathematical Society, 133(2):397–401, 2005. 18

[CSS11] T-H Hubert Chan, Elaine Shi, and Dawn Song. Private and continual release of statistics. ACM Transactions on Information and System Security (TISSEC), 14(3):1–24, 2011. 4, 5

[DKM+06] Cynthia Dwork, Krishnaram Kenthapadi, Frank McSherry, Ilya Mironov, and Moni Naor. Our data, ourselves: Privacy via distributed noise generation. In Annual international conference on the theory and applications of cryptographic techniques, pages 486–503. Springer, 2006. 5

[DMNS16] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. Journal of Privacy and Confidentiality, 7(3):17–51, 2016. 1, 5, 25

[DR+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. 6

[DRV10] Cynthia Dwork, Guy N Rothblum, and Salil Vadhan. Boosting and differential privacy. In 2010 IEEE 51st Annual Symposium on Foundations of Computer Science, pages 51–60. IEEE, 2010. 3, 6

[ENU20] Alexander Edmonds, Aleksandar Nikolov, and Jonathan Ullman. The power of factorization mechanisms in local and central differential privacy. In Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing, pages 425–438, 2020. 2, 19

[FHU22] Hendrik Fichtenberger, Monika Henzinger, and Jalaj Upadhyay. Constant matters: Fine-grained complexity of differentially private continual observation using completely bounded norms. arXiv preprint arXiv:2202.11205, 2022. 17, 18, 19

[FL22] Chenglin Fan and Ping Li. Distances release with differential privacy in tree and grid graph. arXiv preprint arXiv:2204.12488, 2022. 1, 3, 13, 16

[FLL22] Chenglin Fan, Ping Li, and Xiaoyun Li. Breaking the linear error barrier in differentially private graph distance release. arXiv preprint arXiv:2204.14247, 2022. 1, 2, 6, 7, 8, 23, 27

[GDSG20] Abhirup Ghosh, Jiaxin Ding, Rik Sarkar, and Jie Gao. Differentially private range counting in planar graphs for spatial sensing. In Proceedings of the 39th Annual IEEE International Conference on Computer Communications (INFOCOM’20), pages 2233–2242, April 2020. 1

[HP93] Uffe Haagerup and Gilles Pisier. Bounded linear operators between c*-algebras. Duke Mathematical Journal, 71(3):889–925, 1993. 22

[HUU23] Monika Henzinger, Jalaj Upadhyay, and Sarvagya Upadhyay. Almost tight error bounds on differentially private continual counting. In 2023 Symposium on Discrete Algorithm (SODA 2023), 2023. 17
[LMH+15] Chao Li, Gerome Miklau, Michael Hay, Andrew McGregor, and Vibhor Rastogi. The matrix mechanism: optimizing linear counting queries under differential privacy. The VLDB journal, 24(6):757–781, 2015. 22

[Mat93] Roy Mathias. Matrix completions, norms and hadamard products. Proceedings of the American Mathematical Society, 117(4):905–918, 1993. 19, 22

[NTZ13] Aleksandar Nikolov, Kunal Talwar, and Li Zhang. The geometry of differential privacy: the sparse and approximate cases. In Proceedings of the forty-fifth annual ACM symposium on Theory of computing, pages 351–360, 2013. 19

[Pau82] Vern I Paulsen. Completely bounded maps on $c^*$-algebras and invariant operator ranges. Proceedings of the American Mathematical Society, 86(1):91–96, 1982. 2, 22

[Sea16] Adam Sealfon. 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, 2016. 1, 2, 3, 4, 13, 16, 18, 26

[Sea20] Adam Sealfon. Open problem - private all-pairs distances. DifferentialPrivacy.org, 08 2020. https://differentialprivacy.org/open-problem-all-pairs/. 1

[Wai19] Martin J Wainwright. High-dimensional statistics: A non-asymptotic viewpoint, volume 48. Cambridge University Press, 2019. 4, 5, 6, 8
A Technical Preliminaries for Matrix Mechanism

Linear Algebra We fix the following notation for vectors and matrices. We use the lowercase letters for vectors and uppercase for matrices. For a vector \( v \in \mathbb{R}^d \) and a positive integer \( p \), we use \( \|v\|_p \) to denote the \( \ell_p \) norm, i.e. \( \|v\|_p = \left( \sum_{i=1}^d |v_i|^p \right)^{1/p} \). For a matrix \( A \), we use the notation \( A_{ij} \) to denote the \((i, j)\)-th entry of \( A \), and we use \( A_i \) to denote the \( i\)-th row of \( A \), and \( A_j \) to denote the \( j\)-th column of \( A \). For two matrices \( A \) and \( B \) with the same dimension, we use the notation \( A \cdot B \) to denote a matrix \( C \) such that \( C_{ij} = A_{ij}B_{ij} \). We use various norms. For a matrix \( A \in \mathbb{R}^{n \times d} \),

1. \( \|A\|_F \) denotes the Frobenius norm. That is,

\[
\|A\|_F = \left( \sum_{i=1}^n \sum_{j=1}^d |A_{ij}|^2 \right)^{1/2}.
\]

2. For integers \( a \) and \( b \), we denote by \( \|A\|_{a \to b} \) the norm

\[
\max_{\|x\|_a = 1} \|Ax\|_b.
\]

Specifically, when \( a = 2, b = 2 \), this is the spectral norm \( \|A\|_{op} = \max_{\|x\|_2 = 1} \|Ax\| \).

For the next part, we would be using a concept of operator algebra, known as completely bounded norms [Pau82]. It has been also studied in functional analysis and communication complexity by the name of factorization norm ever since Pisier used the factorization characterization of the completely bounded norm Paulsen [Pau82] in his 1986 book. It is also known as hadamard operator norm in computational linear algebra [Mat93].

Definition 12. For a matrix \( A \), its completely bounded spectral norm is defined as

\[
\|A\|_{cb} := \min \left\{ \|L\|_{2 \to \infty} \|R\|_{1 \to 2} : LR = A \right\}.
\]

Haagerup characterized it dual form in his seminal manuscript, a part of which was used in Haagerup and Pisier [HP93]:

\[
\|A\|_{cb} = \max_Q \|Q \cdot A\|_{op} \|Q\|_{op}^{-1},
\]

where \( Q \cdot A \) denotes the Hadamard product and \( \|\|_{op} \) denote the operator norm of the matrix. For a finite-dimensional matrix, we can arrive at this dual characterization using a proper semidefinite program.

A.1 Review of Matrix Mechanism

A fixed set of \( q \) linear queries can be represented in the form of matrix \( M \in \mathbb{R}^{r \times n} \) such that, for any \( n \)-dimensional input vector \( x \in \mathbb{R}^n \) (given in a continual or non-continual manner), the answer for query \( i \) is \((Mx)[i] \) (the \( i\)-th coordinate of the vector \( Mx \in \mathbb{R}^r \)).

Matrix mechanism [LMH+15] is a mechanism where, given \( M \), we first construct an alternate set of matrices known as strategy matrix \( R \in \mathbb{R}^{r \times n} \) and reconstruction matrix \( L \in \mathbb{R}^{d \times r} \) such that \( M = LR \). The strategy matrix is used to generate a private vector, \( v \), by adding a Gaussian noise vector to \( Rx \in \mathbb{R}^r \).

The answer to the original queries are then evaluated from \( v \) by computing \( Lv \), which can be seen as a post-processing step. In particular, the canonical matrix mechanism is

\[
\mathcal{M}_{LR}(M, x) = L(Rx + z),
\]

where \( z \sim N(0, |R|_1^2 \sigma_q^2 \Delta_{M2} L) \) for \( \Delta_{M2} \) be the \( \ell_2 \)-sensitivity of the linear queries defined by \( M \).

The following lemma characterizes the guarantees of the canonical matrix mechanism.
Lemma A.1. Let $Q \in \mathbb{R}^{q \times d}$ be a query matrix consisting of $q$ queries on a histogram of $d$ entries. Then the matrix mechanism, $M_{LR}(Q, x) = L(Rx + z)$, where $z$ is an appropriately distributed Gaussian random vector and $Q = LR$, satisfies the following properties:

1. If $\Sigma = \mathbb{E}[zz^\top]$ has high enough variance, then $y \leftarrow M_{LR}$ is $(\varepsilon, \delta)$-differentially private.
2. With probability $1 - \beta$, $y \leftarrow M_{LR}(Q, x)$ we have
   \[ \|y - Qx\|_\infty \leq O\left(\|L\|_2 \|R\|_{1 \rightarrow 2} \|\Sigma\|_\infty \sqrt{\log(q/\beta)}\right), \]

3. Further for $y \leftarrow M_{LR}(Q, x)$, we have
   \[ \mathbb{E}\left[\|y - Qx\|_2^2\right] \in \Theta\left(\|L\|_F^2 \|R\|_{2 \rightarrow 2}^2 \|\Sigma\|_\infty^2 \sqrt{\log(1/\beta)}\right). \]

B Proofs in Section 3

We present the proofs we skipped in Section 3 in this section.

B.1 Proofs in Proposition 3.1

The algorithms in [CGK+23, FLL22] are slightly different and they rely on different machinery to analyze. Here, we provide a self-contained proof that does not rely on any external machinery other than basic Laplace noise properties. To begin with, we first prove Lemma 3.2 and Lemma 3.3, whose statements can be found in Section 3.1.

Proofs of Lemma 3.2 and Lemma 3.3

Proof of Lemma 3.2. The lemma is proved by a simple application of the Chernoff bound. For each path $P(u, v)$ with more than $\frac{2}{5}n$ edges, let $v'$ be the $\left(\frac{2}{5} + 1\right)$-th vertices on the path $P(u, v)$ from $u$. Similarly, let $u'$ be the $\left(\frac{2}{5} + 1\right)$-th vertices on the path $P(u, v)$ from $v$ (traversing backward). We show that there must be two vertices sampled in $S$ on both $P(u, v')$ and $P(u', v)$, which is sufficient to prove the lemma statement.

Define $X_{u,v'}$ as the random variable for the number of vertices on $P(u, v')$ that are sampled in $S$, and define $X_z$ for each $z \in P(u, v')$ as the indicator random variable for $z$ to be sampled in $S$. It is straightforward to see that $X_{u,v'} = \sum_{z \in P(u, v')} X_z$. Since $P(u, v')$ has at least $\frac{2}{5}n$ vertices, and we are sampling $s = 100 \log n \cdot \frac{n}{\varepsilon}$ vertices uniformly at random as $S$, the expected number of vertices on $P(u, v')$ that are sampled is at least $100 \log n$. Formally, we have

\[ \mathbb{E}[X_{u,v'}] \geq 100 \log n \cdot \frac{\varepsilon}{n} \cdot \frac{n}{\varepsilon} = 100 \log n. \]

As such, by applying the multiplicative Chernoff bound, we have

\[ \Pr[X_{u,v'} \leq 2] \leq \exp\left(-\frac{0.8^2 \cdot 100 \log n}{3}\right) \leq \frac{1}{n^{10}}. \]

The same argument can be applied to $P(u', v)$ by defining $X_{u',v}$ as the total number of vertices that are sampled in $S$. We omit the repetitive details for simplicity. Finally, although the random variables for different $(u, v)$ pairs are dependent, we can still apply a union bound regardless the dependence, and get the desired statement. \qed
Proof of Lemma 3.3. We prove the \( \varepsilon \)-DP case and the \((\varepsilon, \delta)\)-DP case follows with the same argument. Let us condition on the high probability event in Lemma 3.2, and analyze the the paths with \( |P(u, v)| < \frac{n}{\varepsilon} \) and \( |P(u, v)| \geq \frac{n}{\varepsilon} \) separately.

For the vertex pairs where \( |P(u, v)| < \frac{n}{\varepsilon} \), there are at most \( \frac{n}{\varepsilon} \) Laplace noise terms induced by Line 4. By the tail bound of Laplace distribution, with high probability, each of them induces an error of at most \( \varepsilon \cdot \frac{e}{\varepsilon} \log n \). As such, the total difference is at most \( \frac{n}{\varepsilon} \cdot \varepsilon \cdot \frac{e}{\varepsilon} \log n \).

We now analyze the vertex pairs where \( |P(u, v)| \geq \frac{n}{\varepsilon} \). By conditioning on the high probability event in Lemma 3.2, we know that there exists \( x, z \in S \) such that \( |P(u, x)| \leq \frac{n}{\varepsilon} \) and \( |P(z, v)| \leq \frac{n}{\varepsilon} \). As such, for the part of \( d_S(u, x) \) and \( d_S(z, v) \), the noise is at most \( \frac{2n}{\varepsilon} \cdot \varepsilon \cdot \frac{e}{\varepsilon} \cdot \log n \) with high probability (same analysis as the \( |P(u, v)| < \frac{n}{\varepsilon} \) case). For the noise-induced error in \( d_S(x, z) \), we can apply the tail bound of Laplace distribution again to show that with high probability, the error is at most \( \varepsilon \cdot \frac{e}{\varepsilon} \cdot \log n \). Therefore, the total difference between \( d(u, v) \) and \( \tilde{d}(u, v) \) is at most \( \left( \frac{2n}{\varepsilon} \cdot \varepsilon \cdot \frac{e}{\varepsilon} \log n \right) \), as claimed. \( \square \)

Lemma 3.3 is a formalization of the following simple idea: for any vertex pair \((u, v)\), if the shortest path (denoted \( u \leadsto v \)) does not include any edge in the induced graph of vertices in \( S \), the length of the path is bounded by \( \tilde{O}(n \log |S|) \), the algorithm just outputs the summation of noisy weights on all edges along the path from the input-perturbation. Otherwise, suppose the shortest path between \((u, v)\) is \( u \leadsto a \leadsto b \leadsto v \), where \( a, b \) are the first and last vertices along the path through \( S \). The algorithm outputs the private distance between \((a, b)\) obtained by the output-perturbation, and concatenate with the private distances between \((u, a)\) and \((b, v)\).

Proof of Proposition 3.1

We are now ready to complete the proof of the landmark shortcut algorithms.

Analysis of the \( \varepsilon \)-DP algorithm

Lemma B.1. The Landmark Shortcut Scheme with the \( \varepsilon \)-DP parameter in Parameter Set 1 is \( \varepsilon \)-differentially private.

Proof. We can simply use the (basic) composition theorem (Proposition 2.5) to obtained the desired privacy guarantee. Note that one can view Lines 3 (output perturbation, noise on \( f_{out} \)) and 4 (input perturbation, noise on \( f_{in} \)) as two Laplace mechanisms as defined in Definition 7. As such, we only need to prove that both perturbation mechanisms are \( O(\varepsilon) \)-DP.

For the output perturbation, note that by the assumption of \( |w - w'| \leq 1 \), the shortest distance in \( d_S(x, z) \) for each vertex pair \((x, z)\) differs by at most 1. As such, the sensitivity of \( \Delta_{f_{in},1} \) is at most \( \frac{1}{\varepsilon} \).

Therefore, by using the noise from distribution \( \text{Lap} \left( \frac{\varepsilon}{\varepsilon} \right) \) and invoking Proposition 2.3, we can show that the output perturbation mechanism is \( \varepsilon/2 \)-DP.

For the input perturbation, we are directly operating on the edge weights (\( f_{in} \) as the edge weight function). As such, we have \( |f_{in} - f'_{in}| = |w - w'| \leq 1 \). Therefore, by Proposition 2.3, the \( \text{Lap} \left( \frac{\varepsilon}{\varepsilon} \right) \) noise gives an \( \varepsilon/2 \)-DP algorithm.

Finally, we can apply the basic composition theorem (Proposition 2.5) and conclude that the resulting mechanism is \( \varepsilon \)-DP. \( \square \)

Lemma B.2. With high probability, the Landmark Shortcut Scheme with the \( \varepsilon \)-DP parameter in Parameter Set 1 has additive error at most \( O \left( \frac{\varepsilon^2}{\varepsilon} \cdot \log^2 n \right) \).

Proof. By Lemma 3.3, with high probability, the total differences between \( d(u, v) \) and \( \tilde{d}(u, v) \) is at most \( \left( \frac{2n}{\varepsilon} \cdot \varepsilon \cdot \frac{e}{\varepsilon} \log n \right) \). As such, by picking \( e^2 = \frac{\varepsilon^2}{\varepsilon} \) and \( e^2 = \frac{\varepsilon^2}{\varepsilon} \), the total additive error is at most

\[
O \left( \frac{1}{\varepsilon} \cdot \left( \frac{n}{\varepsilon} \cdot \log n + \varepsilon^2 \log^2 n \right) \right).
\]
By letting $\zeta = n^{1/3}$, the resulting error is at most $O\left(\frac{2^{2/3}}{\epsilon} \cdot \log^2 n\right)$.

### Analysis of the $(\epsilon, \delta)$-DP algorithm

**Lemma B.3.** The Landmark Shortcut Scheme with the $(\epsilon, \delta)$-DP parameter in Parameter Set 1 is $(\epsilon, \delta)$-differentially private.

**Proof.** In the analysis of the $\epsilon$-DP mechanism, we use simple composition theorem [DMNS16]. In contrast, in the case of approximate-DP, we can use strong composition (Proposition 2.6) in the proof of the $(\epsilon, \delta)$ mechanism. In particular, for the analysis of the output perturbation (Line 3), we analyze the noise added on the shortest distance for each pair of vertices $x, z \in S$ separately (i.e., $f_{out}^{(x,z)} : S \times S \to \mathbb{R}$). The sensitivity for each $f_{out}^{(x,z)}$ function is at most 1; as such, by Proposition 2.3, we know the mechanism on $f_{out}^{(x,z)}$ is $\frac{1}{\epsilon \cdot \delta}$-DP.

We can then apply the strong composition theorem of Proposition 2.6 to obtain the $\epsilon, \delta$ for each $(\epsilon, \delta)$-DP mechanism. As such, we can again apply the basic composition in Proposition 2.5 to obtain the desired $(\epsilon, \delta)$-DP mechanism.

**Lemma B.4.** With high probability, the Landmark Shortcut Scheme with the $(\epsilon, \delta)$-DP parameter in Parameter Set 1 has additive error at most $O\left(\sqrt{\frac{n}{\epsilon}} \cdot \log n \cdot \sqrt{\log 1/\delta}\right)$.

**Proof.** Again, by Lemma 3.3, with high probability, the total differences between $d(u, v)$ and $\tilde{d}(u, v)$ is at most $\left(\frac{2n}{\epsilon} \cdot e_{\epsilon, \delta}^0 + e_{\epsilon, \delta}^0\right) \cdot \log n$. That is, the summation is in $O\left(\frac{1}{\epsilon} \cdot \left(\frac{\delta}{\zeta} + \frac{\zeta}{\epsilon} \cdot \log 1/\delta \cdot \log n\right)\right)$. By picking $\zeta = n^{1/2} / \sqrt{\log 1/\delta}$, the total error is at most $O\left(\frac{\sqrt{n}}{\epsilon} \cdot \log n \cdot \sqrt{\log 1/\delta}\right)$.

### B.2 Proof of Lemma 3.4

We apply the tail bound for the concentration of Laplace noise as in Lemma 2.1. In particular, conditioning on the high probability event of Lemma 3.2, for any released shortest distance $\tilde{d}(u, v)$, the independent Laplace noises can be listed as follows.

1. At most $\frac{2n}{\epsilon}$ independent noises sampled from $\text{Lap}(e_{\epsilon}^i)$ (resp. $\text{Lap}(e_{\epsilon, \delta}^i)$).
2. At most one noise sampled from $\text{Lap}(e_{\epsilon}^0)$ (resp. $\text{Lap}(e_{\epsilon, \delta}^0)$).

As such, for the parameters in Lemma 2.1, we can let $\nu = \sqrt{\sum_{i} b_i^2}$ and $\lambda = 30 \sqrt{\log n} \cdot \nu$, where $b_i$’s are the variance of independent Laplace noise. As such, to satisfy the condition of $\lambda < \frac{2\sqrt{2\nu^2}}{\max_{i} b_i}$, we need to have $\frac{2\sqrt{2\nu}}{\max_i b_i} \geq 30 \sqrt{\log n}$. In particular, this condition is satisfied by the choice of parameters as follows.

- For the $\epsilon$-DP case, we have $e_{\epsilon}^i = \frac{2}{\epsilon}$ and $e_{\epsilon}^0 = \frac{2}{\epsilon}$. As such, the requirement becomes
  \[2\sqrt{2} \cdot \sqrt{\frac{2n}{\epsilon} + s^2} \geq 30 \sqrt{\log n} \cdot s^2,\]
  which is satisfied when $n \geq C \cdot \zeta^5 \log(n)$ for sufficiently large $C = \Theta(1)$. 

\[25\]
• For the \((\epsilon, \delta)\)-DP case, we have \(e_{\epsilon, \delta}^i = \frac{2}{\epsilon} \) and \(e_{\epsilon, \delta}^o = \frac{4\epsilon}{\delta} \cdot \log \frac{2}{\delta} \). As such, the requirement becomes

\[
2\sqrt{2} \cdot \sqrt{\frac{2n}{\epsilon} + 16s^2 \log^2 1/\delta} \geq 30\sqrt{\log n} \cdot 4s \cdot \log 1/\delta,
\]

which is satisfied when \(n \geq C \cdot \beta^3 \log^2 n \log^2 \frac{1}{\delta} \) for sufficiently large \(C = \Theta(1)\).

Therefore, in both cases, we can apply the concentration results in Lemma 2.1, which gives us that

\[
\Pr \left[ |d(u, v) - \hat{d}(u, v)| \geq 30\sqrt{\log n} \right] \leq 2 \exp \left( - \frac{900 \log n}{8} \right) \leq \frac{1}{n^3}.
\]

A union bound on the above event and the high probability event in Lemma 3.2 completes the proof.

### C Proofs of Section 4

**Proof of Lemma 4.6.** We first claim that we can answer all pair shortest distance on a tree with \((\alpha, \beta)\)-accuracy for

\[
\alpha = O \left( \frac{1}{\epsilon} \log n \sqrt{\frac{\log n}{\beta}} \log \left( \frac{1}{\delta} \right) \right),
\]

showing the utility guarantee of Lemma 4.6. Specifically, if we wish to have high probability bounds for the shortest path distance errors, i.e., \(\beta = O(1/n)\), the error is upper bounded by \(O \left( \frac{1}{2} \log^{1.5} n \sqrt{\log \left( \frac{1}{\delta} \right)} \right)\).

In Sealfon’s algorithm [Sea16], a tree rooted at \(v_0\) is partitioned into subtrees each of at most \(n/2\) vertices. Specifically, define \(v^*\) to be the vertex with at least \(n/2\) descendants but none of \(v^*\)’s children has more than \(n/2\) descendants. The tree is partitioned into the subtrees rooted at the children of \(v^*\), and a subtree of the remaining vertices rooted at \(v_0\). In Sealfon’s algorithm a Laplace noise of \(\text{Lap}(\log n/\epsilon)\) is added to the shortest path distance from \(v_0\) to \(v^*\) and the edges from \(v^*\) to each of its children. The algorithm then repeatedly privatize each of the subtrees recursively. Using Sealfon’s algorithm, we know that for a given root node \(v_0\), computing the single source (with root being the source) shortest path distance requires adding at most \(O(\log n)\) privatized edges. Further, their algorithm ensures that any edge can be in at most \(\log n\) levels of recursion and hence can be used to compute \(O(\log n)\) noisy answers. In other words, the number of adaptive composition we need is \(O(\log n)\).

We use Gaussian mechanism to privatize the edges. Since we are concerned with approximate-DP guarantee, the variance of the noise required to preserve \((\epsilon, \delta)\)-differential privacy is \(\sigma^2 := O \left( \frac{1}{\epsilon} \log (1/\delta) \log n \right)\).

Fix a node \(u\). Let \(\hat{d}(u, v_0)\) be the distance estimated by using Sealfon’s algorithm instantiated with Gaussian mechanism instead of Laplace mechanism. Now the noise added are zero mean. Therefore, \(E[\hat{d}(u, v_0)] = d(u, v_0)\).

Using the standard concentration of Gaussian distribution (Lemma 2.2) implies that

\[
\Pr \left( |\hat{d}(u, v_0) - E[\hat{d}(u, v_0)]| > a \right) \leq 2e^{-a^2/(2\sigma^2 \log n)}.
\]

Setting \(a = \frac{C}{\epsilon} \log n \sqrt{\log \left( \frac{2\epsilon}{\beta} \right)} \log \left( \frac{1}{\delta} \right) \) for some constant \(C > 0\), we have

\[
\Pr \left( |\hat{d}(u, v_0) - E[\hat{d}(u, v_0)]| > \frac{C}{\epsilon} \log n \sqrt{\log \left( \frac{2\epsilon}{\beta} \right)} \log \left( \frac{1}{\delta} \right) \right) \leq 2e^{-C \log (2n/\beta)} \leq \frac{\beta}{n}.
\]
Now union bound gives that
\[
\Pr \left( \max_{u \in V} |\hat{d}(u, v_0) - \hat{d}(u, v'_0)| \leq \frac{C}{\varepsilon} \log n \sqrt{\log \left( \frac{n}{\beta} \right) \log \left( \frac{1}{\delta} \right)} \right) \geq 1 - \beta.
\]

We can now use the above result to answer all pair shortest path by fixing a node \(v^*\) to be the root node and compute single source shortest distance with root node being the source node. Once we have all these estimates, to compute all pair shortest distance, for any two vertices, \((u, v) \in V \times V\), we first compute the least common ancestor \(z\) of \(u\) and \(v\). We then compute the distance as follows:
\[
\hat{d}(u, v) = \hat{d}(u, v^*) + \hat{d}(v^*, v) - 2\hat{d}(z, v^*).
\]

Since each of these estimates can be computed with an absolute error \(O \left( \frac{1}{\varepsilon} \log n \sqrt{\log \left( \frac{n}{\beta} \right) \log \left( \frac{1}{\delta} \right)} \right)\), we get the final additive error bound. That is,
\[
\Pr \left( \max_{u, v \in V} |\hat{d}(u, v) - d(u, v)| \right) = O \left( \frac{1}{\varepsilon} \log n \sqrt{\log \left( \frac{n}{\beta} \right) \log \left( \frac{1}{\delta} \right)} \right) \geq 1 - \beta
\]
completing the proof of the claim.  

\[\square\]

D Combinatorial Algorithms under Matrix Mechanism

En route to proving the bounds on special graphs, we observe that the power of matrix mechanism goes beyond being a proof tool: it offers an independent perspective for differential privacy, and many known combinatorial DP algorithms can be transformed into a matrix mechanism. In particular, in this section, we derive the matrix mechanism form of the landmark shortcut-based algorithms (Section 3, see also [FLL22, CGK+23]) and our improved \(\varepsilon\)-DP algorithm (Section 4.1). These results can be of independent interest.

D.1 The Landmark Shortcut-based Algorithms

Let \(S\) be a random subset of nodes for which we compute all pair shortest paths in a privacy-preserving manner. One can construct the following factorization:
\[
R = \begin{pmatrix}
I_m & 0 \\
0 & I_s
\end{pmatrix},
\]
where \(I_n\) denotes an \(n \times n\) identity matrix. Now the matrix \(L\) is an \(\binom{n}{2} \times (m + s^2)\) such that the \(i\)-th row corresponds to the shortest path query for nodes \(u, v \in V\). That is, the \(i\)-th row \((L_{i,1}, L_{i,2}) \in \{0, 1\}^{m+s^2}\) has a specific form: \(L_{i,1}\) has 1 corresponding to the part of the shortest paths from \((u_1, v_1)\) and \((v_1, v)\) where \(v_1, u_1 \in S\) and \(L_{i,2}\) has 1 only in the position corresponding to the distance \((u_1, v_1)\). Using this factorization view, we can see the shortest path computation as the following linear algebraic manipulation:
\[
L(Rx + z), \quad \text{where} \quad x = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}, \quad z = \begin{pmatrix} z_1 \\ z_2 \end{pmatrix}.
\]

Here \(x_1 \in \mathbb{R}^m\) corresponds to the weights on the edges and \(x_2 \in \mathbb{R}^s\) corresponds to the real shortest path between the nodes in \(S\). Also, for privacy \(z_1 \sim N(0, \sigma_{\varepsilon, \delta}^2 m)\) and \(z_2 \sim N(0, s^2 \sigma_{\varepsilon, \delta}^2)\). Here \(\sigma_{\varepsilon, \delta}\) is the variance required for \((\varepsilon, \delta)\)-differential privacy. Using this linear algebraic form, we get that the error is
\[
O \left( \sqrt{\frac{2n}{s} \sigma_{\varepsilon, \delta}^2 + s^2 \sigma_{\varepsilon, \delta}^2} \log(n) \right),
\]
which is minimized when \(s = n^{1/3}\). This leads to the total error of \(O(n^{1/3} \sigma_{\varepsilon, \delta} \sqrt{\log(n)})\).
D.2 Matrix Mechanism View for Canonical Segments

Let $Q$ be the shortest distance matrix for a given graph. Let $S$ be the set of random vertices picked and $\text{Canon}(S)$ be the canonical segment corresponding to $S$. Let $P_{\text{Canon}}(S)$ be the set of pairs of vertices $(u, v) \in V \times V$ such that $(u \leadsto v) \cap \text{Canon}(S)$ is non-empty, i.e., the shortest path between $u$ and $v$ contains an entry from $\text{Canon}(S)$. Let $E_{\text{Canon}}(S)$ be the edges in $\text{Canon}(S)$. We form a block matrix $R_2 \in \{0, 1\}^{p \times e}$ whose rows are $P_{\text{Canon}}(S)$ and columns are $E_{\text{Canon}}(S)$ and the corresponding entries from the matrix $Q$. In other words, $p = |P_{\text{Canon}}(S)|$ and $e = |E_{\text{Canon}}(S)|$, and, the $(i, j)$-th entry of $R_2$ is 1 if the shortest path for vertices corresponding to row $i$ contains the edge corresponding to column $j$, and 0 otherwise. Finally, let $R_1 = I_m$.

Our right matrix is, therefore,

$$R = \begin{pmatrix} R_1 & 0 \\ 0 & R_2 \end{pmatrix}.$$

Our left matrix $L \in \{0, 1\}^{\binom{n}{2} \times (m + p)}$ is such that the $i$-th row corresponds to the shortest path query for nodes $(u, v) \in V \times V$. That is, the $i$-th row $(L_{[i], 1} L_{[i], 2}) \in \{0, 1\}^{m + p}$ has a specific form: $L_{[i], 1}$ has 1 corresponding to the part of the shortest paths from $(u, u_1)$ and $(v_1, v)$ where $v_1, u_1 \in S$ and $L_{[i], 2}$ has 1 only in the position corresponding to the set in $\text{Canon}(S)$ and $L_{[i], 2}$ has 1 only in canonical segments. Using this factorization view, the shortest path computation is the following linear algebraic manipulation:

$$L(Rx + z), \quad \text{where} \quad x = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}, \quad z = \begin{pmatrix} z_1 \\ z_2 \end{pmatrix}.$$

Here $x_1 \in \mathbb{R}^m$ corresponds to the weights on the edges and $x_2 \in \mathbb{R}^p$ corresponds to the weights in the edges corresponding to $\text{Canon}(S)$, and $x_2 \in \mathbb{R}^q$ corresponds to the in canonical segments. Also, for privacy entries of $z_1$ are sampled from $\text{Lap}(\varepsilon/2)$, and entries of $z_2$ are sampled from $\text{Lap}(\varepsilon/s^2)$. Using this linear algebraic form, we get that the error is

$$O \left( \frac{1}{\varepsilon} \sqrt{\left( \frac{2n}{\xi} + \xi^2 \cdot \log^2 n \right) \log(n)} \right),$$

which is minimized when $\xi = \tilde{O}(n^{1/3})$. This recovers the result with our canonical segments technique.