Differentially Private Range Query on Shortest Paths

Chengyuan Deng, Jie Gao(✉), Jalaj Upadhyay(✉), and Chen Wang(✉)

Rutgers University, Piscataway, USA
{cd751,jg1555,upadhyay,wc497}@rutgers.edu

Abstract. We consider range queries on a graph under the constraints of differential privacy and query ranges are defined as the set of edges on the shortest path of the graph. Edges in the graph carry sensitive attributes and the goal is to report the sum of these attributes on the shortest path for *counting query* or the minimum of the attributes in a *bottleneck query*. We use differential privacy to ensure that answering these queries does not violate the privacy of the sensitive edge attributes. Our goal is to design mechanisms that minimize the additive error of the output with the given privacy budget.

For this, we develop the first set of non-trivial results for private range queries on shortest paths. For counting range queries we can achieve an additive error of $\widetilde{O}(n^{1/3})$ for ε-DP and $\widetilde{O}(n^{1/4})$ for (ε, δ)-DP. We present two algorithms where we control the final error by carefully balancing perturbation added to the edge attributes directly versus perturbation added to (a subset of) range query answers. Bottleneck range queries are easier and can be answered with polylogarithmic additive errors.

Keywords: Range query · Differential privacy · Shortest path

1 Introduction

Range counting has been extensively studied in the literature, particularly for geometric ranges. In the typical setting, there is a set of points X in $\mathbb{R}^d$. A range query is often formulated by a geometric shape, and range counting reports the number of points inside the range [34]. The points can be weighted, in which case the goal is to return the weighted sum inside the query range. Compared to the huge literature on geometric range queries [48], there has been much less work on the study of range queries with non-geometric ranges.

In this paper, we study private range counting when the ranges are defined as paths on a graph. This setting becomes interesting with the exploding amount of graph data. Graphs are used as a natural mathematical structure to model pairwise relations between objects. Often, the pairwise relations or attributes can represent private and confidential information. As such, performing statistics on such a graph without any robust privacy guarantee can be problematic. We consider the scenario where both the graph topology and

P. Morin and S. Suri (Eds.): WADS 2023, LNCS 14079, pp. 340–370, 2023.
https://doi.org/10.1007/978-3-031-38906-1_23

the query ranges (paths on the graph) are public information, but attributes on the edges of the graph, that may come from private sources, are sensitive and protected. Our goal is to return (approximate) range queries while protecting data privacy.

The above model is applicable in many real-world scenarios. In financial analysis, graph-based techniques have been adopted to combat fraud [39]. One can consider a graph where edges represent transactions between two financial entities with attributes such as the total amount being transferred. Forensic analysis researchers may want to issue queries along certain paths that involve multiple financial entities to detect anomalies. In supply chain networks, vertices represent participants such as producers, transporters or retailers, and edges represent their relationships. Resilience is a critical factor in supply chains and metrics on edges such as Time-to-Stockout (TTS) [29] have been used for estimating end-to-end resilience of certain paths. Response time or cost are also important edge attributes. In these settings, privacy and security issues of the attributes are natural and crucial (e.g., as trade secrets) [38]. In road networks, ranges can be naturally defined as paths that users take and queries are about collective statistics of traffic along the path. Privacy is also crucial in healthcare information systems [46].

1.1 Our Setting and Results

We consider the setting when query ranges are taken as shortest paths based on *public* edge weights, and the query answer is a function of *private* attributes on the edges involved in a query range/path. Using shortest paths between two vertices is natural in many of the application settings discussed above. Further, if the range query is applied on arbitrary paths in a graph, the additive query error needed to ensure privacy can be as large as $\Omega(n)$, where n is the number of vertices in the graph. We give a proof of this in Appendix A.

We consider two types of query function f on a path P:

- *Counting query*: return the sum of the attribute values on edges of P;
- *Bottleneck query*: return the minimum of the attribute values on edges of P.

Since the attribute values are private and sensitive, the reported range query answers are perturbed to ensure differential privacy guarantees. Specifically, we consider two neighboring attribute value sets w and w' on the same graph G, which differ by a ℓ_1 norm of 1. A mechanism $\mathcal{A}$ is called (ε, δ)-differentially private if the probability of obtaining query outputs on input attributes w or w' is relatively bounded by a multiplicative error of e^{ε} and an additive error of δ. When $\delta = 0$, we call $\mathcal{A}$ ε-DP or pure-DP. The objective is to achieve the specified privacy requirement with noise perturbation as small as possible.

In this paper, we study the private range query (both counting and bottleneck) on the shortest paths. As standard in the literature of differential privacy, our aim is to understand the trade-off between privacy and additive error in the final query answer, i.e., for a given privacy budget, minimize the additive

error. One can additionally consider the query time and space required for the data structure. We leave designing a differentially private data structure with a better query time-space trade-off as a direction of future research.

For counting queries, we present two algorithms with privacy guarantees of pure-DP and approximate-DP respectively (in Sect. 3 and Sect. 4), returning the counts with relatively small worst-case additive errors. Our main results are captured by the following theorem:

Result 1 (ε-DP algorithm for counting query, informal version of Theorem 1). *There exists an ε-differentially private algorithm that outputs counting queries along all pairs shortest paths with additive error at most $\widetilde{O}(\frac{n^{1/3}}{\varepsilon})$ with high probability.*

Result 2 ((ε, δ)-DP algorithm for counting query, informal version of Theroem 2). *There exists an (ε, δ)-differentially private algorithm that outputs counting queries along all pairs shortest paths with additive error at most $\widetilde{O}(\frac{n^{1/4}}{\varepsilon} \log^{1/2} \frac{1}{\delta})$ with high probability.*

The above results are the first known upper bounds for this specific problem. Meanwhile, we establish a lower bound of $\Omega(n^{1/6})$ adapted from the construction of the lower bound for private all pairs shortest distances [10] (with details in Appendix D). The gap between the best-known upper and lower bounds provokes an interesting perspective of private range queries: we do not yet have optimal bounds for specific ranges, despite the results by [36] presenting optimal bounds for generic range query problems. Closing the gap for counting queries would also be an interesting open question. Our next result, however, shows that the bottleneck query yields simple algorithms using existing techniques to achieve logarithm additive error:

Result 3 (DP algorithms for bottleneck query, informal version of Theroem 3). *There exists an ε-differentially private algorithm and an (ε, δ)-differentially private algorithm, such that with high probability, outputs bottleneck queries along all pairs shortest paths with additive error at most $\widetilde{O}(\frac{\log n}{\varepsilon})$ and $\widetilde{O}(\frac{\sqrt{\log n \log \frac{1}{\delta}}}{\varepsilon})$ respectively.*

Collectively, our results give the first set of non-trivial bounds for privately releasing queries for shortest paths on range query systems. We further show that it is possible to use the VC-dimension of shortest paths queries to obtain a bound similar to Result 2, albeit with a much more complicated algorithm for generic range query applications from [36].

1.2 Main Techniques

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.

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 range query function. If the edge attribute changes by 1 in the ℓ_1 norm, there can be up to $\Theta(n^2)$ query pairs being impacted – e.g., when $\Theta(n^2)$ shortest paths share one edge. As such, if we apply a crude output perturbation, the noise for each query should be $\widetilde{O}(n^2)$ for ε-differential privacy and $\widetilde{O}(n)$ for (ε, δ)-differential privacy. On the other hand, with input perturbation, one can add a Laplace noise of magnitude proportional to $1/\varepsilon$ to each edge attribute. This satisfies ε-privacy, but the shortest path may have up to order n edges, and the noises on edges are accumulated with a total error of $\widetilde{O}(n)$.

To improve the error bound, we actually need to combine input and output perturbations. 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 attributes. The error for input perturbation depends on the *graph hop diameter*, i.e., the maximum number of edge attributes that we need to sum up as the output of counting queries. Therefore, one natural idea is to introduce 'shortcuts' (to replace a selective set of shortest paths) to the graph such that the network diameter is reduced. We then apply output perturbation on the shortcuts and use input perturbation on the graph with shortcuts. Of course, when the shortcuts are introduced, we need to be mindful of their sensitivity. The natural question is, can we reduce the network diameter with no or limited increment to the edge sensitivity with the introduction of the shortcuts?

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), we name u_1, u_2 as the *cut vertices*. Since there are $\binom{s}{2}$ shortest paths for all pairs in $\mathcal{S}$, there are at most $O(s^2)$ cut vertices on any shortest path $P(u, v)$ with $(u, v) \in \mathcal{S} \times \mathcal{S}$. For every $(u, v) \in \mathcal{S} \times \mathcal{S}$, 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 $\widetilde{O}(\sqrt{n/s + s^2})$, for our ε-DP algorithm. Plugging in $s = n^{1/3}$, we can get an error of $\widetilde{O}(n^{1/3})$ and an ε-DP algorithm.

Approximate-DP Algorithm. Our solution for (ε, δ)-DP exploits properties of strong composition [17], which allows us to massage k (ε, δ)-DP mechanisms into an (ε', δ')-DP mechanism, where $\varepsilon' \approx \varepsilon\sqrt{k}$ and $\delta' \approx k\delta$. Our strategy to leverage strong decomposition is to build a shortest path tree rooted at each vertex in the sampled set $\mathcal{S}$. Tree graphs admit much better differentially private

mechanisms – one can get polylogarithmic additive error for running queries on a tree graph [18,45]. Now for any two vertices u, v in $\mathcal{G}$, if the shortest path $P(u,v)$ has more than $\widetilde{O}(n/s)$ vertices, $P(u,v)$ has at least one vertex w in $\mathcal{S}$ 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 query values 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 (ε, δ)-DP. The final error bound is $\widetilde{O}\left(\sqrt{n/s} + \sqrt{s}\right)$. Optimizing the error by setting $s = \widetilde{O}(\sqrt{n})$ gives an (ε, δ)-DP mechanism with an additive error of $\widetilde{O}(n^{1/4})$.

Remark 1. Our scheme for the approximate-DP algorithm can also be applied to the pure-DP regime to obtain the same upper bound of $\widetilde{O}(n^{1/3})$, using the basic composition theorem (Proposition 4) 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.

Remark 2. The algorithm using canonical segments works only for undirected graphs, while the algorithm using shortest path trees can be extended for directed graphs. In particular, we can build two shortest path trees at each sampled vertex w, one $T_{in}(w)$ with edges pointing towards w and one tree $T_{out}(w)$ with edges pointing away from w. Any shortest path $P(u,v)$ that visits a vertex $w \in S$ is composed of the shortest path from u to w (captured in the tree $T_{in}(w)$) and then a path from w to v (captured in tree $T_{out}(w)$). With this in mind, throughout the paper we assume an undirected graph.

1.3 Related Work

Geometric Range Queries. Geometric range queries typically consider halfplane ranges, axis-parallel rectangles (orthogonal range query), or simplices (simplex range query). The majority of work on range counting considers upper and lower bounds on the running time for answering a query, with different data storage requirements [48]. Designing geometric data structures while preserving differential privacy has also gained attention in the recent past. For example, Biemel et al. [3,30] looked at the problem of the center point of a convex hull. They instantiated exponential mechanism with *Tukey depth* [49] as the score function. Since then, several works have looked at various geometric problems, like learning axis-aligned rectangles [4,44], where one can achieve optimal error bound under pure differential privacy using exponential mechanism; however, the case for approximate differential privacy is still open. There has been some recent work that studied differentially private geometric range queries (e.g., orthogonal range queries) under both the *central model* and *local model* of privacy [11,12,21,36,40,51,53].

Differentially Private Linear Queries. A fundamental class of queries studied in the literature of differential privacy are linear queries on a dataset [2,5–8,15,23–28,31,32,37,41,42,52]. Here, given a dataset from a data universe $\mathcal{U}$ of size d (usually represented in a form of a histogram $D \in \mathbb{R}^d$) and a query $q \in \mathbb{R}^d$, the goal is to estimate $q^\top D$. One can replace the query vector with a predicate $\phi : \mathcal{U}^n \to \{0,1\}$, where n is the size of the database, $D = \{d_1, \cdots, d_n\} \in \mathcal{U}^n$. The counting query is then simply $\sum_{i=1}^n \phi(d_i)$. Range queries can be seen as a special case of linear queries with a properly defined set of predicates.

The most relevant work to this paper is the work by Muthukrishnan and Nikolov [36], who proposed a differentially private mechanism for answering (generic) range queries when the ranges have bounded VC-dimension [36]. We can apply their techniques to get results for our setting of using shortest paths as ranges. Our algorithm can be easily extended to guarantee ε-differentially private with a slight change of parameters, while this substitution is non-trivial for the algorithm of Muthukrishnan and Nikolov [36], and to the best of our understanding, yields sub-optimal error bound. More discussion of this is in Sect. 6.

Private Release of Graph Data. Private release of graph data has been studied in recent years on many graph properties; see the survey [33]. There has been recent work on differentially private release of all pairs shortest path length [10,18,19,45]. Here, the edge weights w is considered sensitive, and the goal is to produce an approximate distance matrix for all pairs shortest paths length with differential privacy guarantees. In other words, the edge weights w *are* the sensitive attributes a. This is a harder problem than the problem considered in this paper. Specifically, the topology of the shortest paths are public information in our setting, but the knowledge of which edges are on the shortest path may reveal knowledge of the sensitive edge length w. It has been shown in [45] that when one releases the set of edges on an approximate shortest path in a differentially private manner, the additive error in the distance report has to be as large as $\Omega(n)$. The best known results for private release of all pairs shortest distance have an additive error of $\widetilde{O}(n^{2/3})$ for pure-DP and $\widetilde{O}(\sqrt{n})$ for approximate-DP [10,18,19] for general graphs. There is a lower bound of $\Omega(n^{1/6})$ for approximate-DP [10]. For trees the two problems are the same since for any two nodes the shortest path is unique regardless of edge length.

Differentially private range query on shortest paths has been done on a planar graph in [22], where they provide mechanisms with polylogarithmic additive error. But this problem has not been studied for the general graph setting.

2 Preliminaries

Notation. We use $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ to denote a graph on vertex set $\mathcal{V}$ and edges $\mathcal{E}$. An edge $e \in \mathcal{E}$ is also denoted by the tuple (u, v) if u and v are its endpoints. For a pair of vertices (u, v), we denote $P(u, v)$ as their shortest path, and $d(u, v)$

as the shortest distance. We can define the attribute function $w : \mathcal{E} \to \mathbb{R}^m$ over all the edges *independent* of the shortest paths. On a path $P(u,v)$, we let $\gamma(u,v) := \min_e\{w(e)|e \in P(u,v)\}$ as the minimum attribute value along the shortest path $P(u,v)$. We use $\mathcal{R} = (X,\mathcal{S})$ to denote a set system, where $\mathcal{S}$ is a collection of sets with elements from X.

2.1 The Models for Range Query and Privacy

Shortest Paths as Ranges. Let $\mathcal{R} = (X,\mathcal{S})$ be a set system, where X is a set of elements and $\mathcal{S}$ is a collection of subsets $S_i \subseteq X$ called *ranges*. In a graph $\mathcal{G}$ when shortest paths are unique[1], we can define shortest paths as ranges. We take X to be the set of m edges in $\mathcal{G}$, and each set of $\mathcal{S}$ corresponds to a set of edges on a (u,v) shortest path. In particular, for an undirected graph $\mathcal{G}$, its corresponding $\mathcal{S}$ has $\binom{n}{2}$ order sets; and for a directed graph $\mathcal{G}$, $\mathcal{S}$ may have up to n^2 ordered sets.

Based on the set system $\mathcal{R} = (X,\mathcal{S})$, we can define *range queries* on $\mathcal{R}$ as $(\mathcal{R}, f)$ with a *query function* $f : \mathcal{S} \to \mathbb{R}$ as $\{f(S)\}_{S\in\mathcal{S}}$ for every set in $\mathcal{S}$. We can further extend this notion of range queries on shortest distances with *attribute* functions $w : X \to \mathbb{R}^{\geq 0}$, and the queries on each set S become $f(w(S))$, where $w(S)$ means to apply attribute function to each element in S. Note that the attribute function should *not* be considered as edge weights as it does not affect the shortest paths. Our goal is to release the statistics of *all* sets with small additive errors and privacy guarantees following the definitions in Definition 2.

We now formally define the privacy model for range queries on shortest paths.

Definition 1 (Range Queries with Neighboring Attributes). *Let $(\mathcal{R} = (X,\mathcal{S}), f)$ be a system of range queries, and let $w, w' : X \to \mathbb{R}^{\geq 0}$ be attribute functions that map each element in X to a non-negative real number. We say the attributes are* neighboring

$$\sum_{x\in X} |w(x) - w'(x)| \leq 1.$$

We emphasize that the attributes do not *change the shortest paths, i.e., the graphs operate on the same set system $\mathcal{R} = (X,\mathcal{S})$. When it is clear from context, we abuse the notation and denote the above by $\|w - w'\|_1 \leq 1$.*

We shall define the pure- and approximate DP with the notions of the neighboring attributes on range queries as follows.

Definition 2 (Differentially Private Range Queries). *Let $(\mathcal{R} = (X,\mathcal{S}), f)$ be a system of range queries and $w, w' : X \to \mathbb{R}^{\geq 0}$ be attribute functions as prescribed in Definition 1. Furthermore, let $\mathcal{A}$ be an algorithm that takes $(\mathcal{R}, f, w)$ as input. Then*

[1] One can use symbolic perturbation of edge distances to produce unique shortest paths.

$\mathcal{A}$ is (ε, δ)-differentially private on $\mathcal{G}$ if, for all pairs of neighboring attribute functions w, w' and all sets of possible outputs $\mathcal{C}$, we have that

$$\Pr[\mathcal{A}(\mathcal{R}, f, w) \in \mathcal{C}] \leq e^{\varepsilon} \cdot \Pr[\mathcal{A}(\mathcal{R}, f, w') \in \mathcal{C}] + \delta.$$

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

We now define the notion that characterizes the *utility* of the algorithm. In the range query model, we say an algorithm $\mathcal{A}$ provides (α, β)-approximation to *all sets range queries* (ASRQ) if, given a range query system $(\mathcal{R} = (X, \mathcal{S}), f)$ and a attribute function w, with probability at least $1 - \beta$, algorithm $\mathcal{A}$ outputs an answer within an α additive error for the original query value on every set.

Definition 3 (Approximate-ASRQ). *We say a randomized algorithm $\mathcal{A}$ is an (α, β)-approximation for* all sets range queries *(ASRQ) on a range query system $(\mathcal{R} = (X, \mathcal{S}), f)$ with attribute function w if for any $S \in \mathcal{S}$,*

$$\Pr\left[|f(w(S)) - \mathcal{A}(w(S))| \leq \alpha\right] \geq 1 - \beta.$$

Since S contains the ranges of all-pairs shortest paths, the approximation in Definition 3 naturally corresponds to the additive approximation of shortest distances when f is the *counting query*. Trivially, if we output the range queries simply based on the elements and the attribute function w, we have $\alpha = \beta = 0$. However, such an output will *not* be private – and to guarantee both privacy and approximation is the main focus of this paper.

Remark 3. Our model of Definition 1 is closely related to the all-pair shortest distances release studied in [10,18,19,45]. In particular, in the model of private all-pair shortest distances, the neighboring graphs are also defined as the norm of attributes differing by at most 1. However, there is a subtle difference: in the shortest distances model, the shortest paths are private and subject to protection; while in the range query model, the shortest paths are known, and we do *not* have to protect their privacy. This allows us to bypass the $\Omega(n)$ additive error lower bound in [45] for any algorithm that privately reveal the shortest paths, and obtain much stronger results.

2.2 Standard Technical Tools

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 [50].

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

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

Definition 5 (Gaussian distribution) . *We say a zero-mean random variable X follows the Gaussian distribution with variance σ^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 [50].

Lemma 1 (Sum of Laplace random variables, [9,50]). *Let $\{X_i\}_{i=1}^m$ be a collection of independent random variables such that $X_i \sim \mathsf{Lap}(b_i)$ for all $1 \leq i \leq m$. Then, for $\nu \geq \sqrt{\sum_i b_i^2}$ 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 (Sum of Gaussian random variables, [50]). *Let $\{X_i\}_{i=1}^m$ be a collection of independent random variables such that $X_i \sim \mathcal{N}(\mu, \delta^2)$ for all $1 \leq i \leq m$. Then,*

$$\Pr\left[\left|\frac{\sum_i X_i}{m} - \mu\right| \geq \lambda\right] \leq 2 \cdot \exp\left(-\frac{m\lambda^2}{2\delta^2}\right).$$

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 ℓ_p-sensitivity of the function f is defined as*

$$\Delta_{f,p} = \max_{\substack{w,w' \in \mathcal{X} \\ w \sim w'}} \|f(w) - f(w')\|_p,$$

Here, $\|\mathbf{x}\|_p := \left(\sum_{i=1}^d |\mathbf{x}[i]|^p\right)^{1/p}$ is the ℓ_p-norm of the vector $\mathbf{x} \in \mathbb{R}^d$ and $\mathbf{x}[i]$ 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 ℓ_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 $\mathsf{Lap}(\frac{\Delta_{f,1}}{\varepsilon})$ and outputs*

$$M_\varepsilon(f) = f(w) + (Y_1, \ldots, Y_k).$$

The following privacy property of Laplace mechanism is known.

Proposition 1 (Laplace mechanism [15]). *The Laplace mechanism $M_\varepsilon(f)$ is ε-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 the Gaussian distribution $\mathcal{N}\left(0, \frac{2\Delta_{f,2}^2 \log(1.25/\delta)}{\varepsilon^2}\right)$ and outputs*

$$M_\varepsilon(f) = f(w) + (Y_1, \ldots, Y_k).$$

The following privacy property of Gaussian mechanism is known.

Proposition 2 (Gaussian mechanism [14]). *For $\varepsilon \in (0,1)$, the Gaussian mechanism $M_{\varepsilon;\delta}(f)$ is (ε, δ)-differentially private.*

It is well-known that if a mechanism M provides (ε, δ)-DP output, any function g that takes the output of M as input is also (ε, δ)-DP. This is known as the *post-processing theorem*, formalized as follows.

Proposition 3 (Post-processing theorem [16]). *Let $M : \mathbb{R}^{d_1} \to \mathbb{R}^{d_2}$ be an (ε, δ)-differentially private mechanism and let $g : \mathbb{R}^{d_2} \to \mathbb{R}^{d_3}$ be an arbitrary function. Then, the function $g \circ M : \mathbb{R}^{d_1} \to \mathbb{R}^{d_3}$ is also (ε, δ)-differentially private.*

Finally, we introduce another useful property of differential privacy: privacy is preserved when combining multiple differentially private mechanisms even against adaptive adversary.

Proposition 4 (Composition theorem [15]). *For any $\varepsilon > 0$, the adaptive composition of k ε-differentially private algorithms is $k\varepsilon$-differentially private.*

Proposition 5 (Strong composition theorem [17]). *For any $\varepsilon, \delta \geq 0$ and $\delta' > 0$, the adaptive composition of k (ε, δ)-differentially private algorithms is $(\varepsilon', k\delta + \delta')$-differentially private for*

$$\varepsilon' = \sqrt{2k \ln(1/\delta')} \cdot \varepsilon + k\varepsilon(e^\varepsilon - 1).$$

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

$$\varepsilon' = \varepsilon \cdot \sqrt{8k \log(\frac{1}{\delta'})}.$$

The following proposition follows from strong composition theorem.

Proposition 6 (Corollary 3.21 in [16]). *Let $\mathcal{A}_1, \cdots, \mathcal{A}_k$ be k (ε', δ')-differentially private algorithm for*

$$\varepsilon' = \frac{\varepsilon}{\sqrt{8k \log(1/\delta)}}.$$

Then an algorithm $\mathcal{A}$ formed by adaptive composition of $\mathcal{A}_1, \cdots, \mathcal{A}_k$ is $(\varepsilon, k\delta' + \delta)$-differentially private.

3 An ε-DP Algorithm for Counting Queries

In the current and following section, we focus on private algorithms for the counting query function. As clarified in Remark 1, the algorithms using single-source shortest-path tree scheme can achieve ε and (ε, δ)-DP regime using only different parameters. However, we propose a different algorithmic idea for pure-DP algorithm, which shaves off a $\log^2 n$ factor. We formally state the results on ε-DP as follows.

Theorem 1. *For any $\varepsilon \geq 0$, there exists an ε-differentially private efficient algorithm that given a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, w)$ as a range query system $(\mathcal{R} = (X, \mathcal{S}), f, w)$ such that $\mathcal{S}$ is the set of the shortest paths and f is the counting query, with high probability, outputs all pairs counting queries with additive error $O(\frac{n^{1/3}\log^{5/6} n}{\varepsilon})$. That is, the algorithm outputs an estimate $\widehat{f}(\cdot,\cdot)$ such that*

$$\Pr\left(\max_{u,v\in\mathcal{V}} |\widehat{f}(u,v) - f(u,v)| = O\left(\frac{n^{1/3}\log^{5/6} n}{\varepsilon}\right)\right) \geq 1 - \frac{1}{n}.$$

We start with some high-level intuitions. Our algorithm leverages both input-perturbation and output-perturbation, as mentioned in Sect. 1.2. A naive solution would be applying output-perturbation to the pair-wise counting queries for vertices in $\mathcal{S}$. However, the change of a single edge attribute may trigger the change of potentially all counting queries for vertices in $\mathcal{S}$. As such, by the composition theorem, we need to boost the privacy parameter by a factor of $|\mathcal{S}|^2$ since *each* counting query can change by 1. On the other hand, note that the ranges are shortest paths, which have special structures. With the standard assumption that all shortest paths are unique, two shortest paths only overlap by one common shortest path segment. Therefore instead of using output perturbation directly among vertices in $\mathcal{S}$, 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 $|\mathcal{S}|^2$, 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 $\mathcal{S} \subseteq \mathcal{V}$. Informally, a vertex w becomes a cut vertex if it is a vertex of $\mathcal{S}$, or if it witnesses the branching – either 'merging' or 'splitting' – of two shortest paths between different pairs of vertices in $\mathcal{S}$. The formal definition is as follows.

Definition 9 (Cut Vertices). *Let $\mathcal{S} \subseteq \mathcal{V}$ be an arbitrary subset of vertices. For any pair of vertices $(u,v) \in \mathcal{S}$ 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. $w \notin \{u, v\}$ *and*
   (a) $w \in P(x,z)$ *for some* $(x,z) \in \mathcal{S}$ *and* $(x,z) \neq (u,v)$;

(b) Without any loss of generality, suppose the path is from x. Let $\mathrm{pred}(w)$ *be the vertex before w on $P(x,z)$ and* $\mathrm{succ}(w)$ *be the vertex after w on $P(x,z)$. Then either* $\mathrm{pred}(w) \notin P(u,v)$ *or* $\mathrm{succ}(w) \notin P(u,v)$.

See Fig. 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 $\mathcal{S}$.

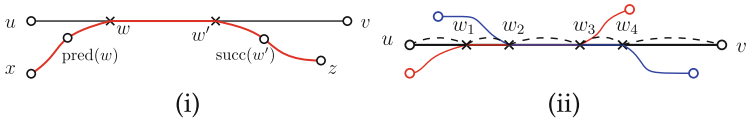

Fig. 1. (i) Two shortest paths $P(u,v)$ and $P(x,z)$, $u,v,x,z \in \mathcal{S}$, intersect at a common subpath as the shortest path between two cut vertices w, w'. (ii) The shortest path $P(u,v)$ is partitioned into canonical segments $P(u,w_1), P(w_1,w_2), \cdots, P(w_\ell, w_{\ell+1}), P(w_{\ell+1},v)$, where $w_1, w_2, \cdots, w_{\ell+1}$ are (ordered) cut vertices along path $P(u,v)$.

Definition 10 (Canonical Segments). *Let $\mathcal{S} \subseteq \mathcal{V}$ be an arbitrary subset of vertices. For any pair of vertices $(u,v) \in \mathcal{S}$ 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 $(x,z) \in \mathcal{S}$;*
2. *w' is a cut vertex for some $(x',z') \in \mathcal{S}$;*
3. *None of the vertices between w and w' on $P(u,v)$ is a cut vertex for any $(x'',z'') \in \mathcal{S}$.*

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 $\mathcal{S}$, and the canonical segments are exactly the collection of segments between adjacent cut vertices along shortest paths. See Fig. 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 $(u,v) \in \mathcal{S}$, we define $\mathsf{Canon}(\mathcal{S},u,v)$ 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 $\mathcal{S}$: the shortest path between $(u,v) \in \mathcal{S}$ may well be outside of $\mathcal{S}$. We provide some observations about the basic properties of canonical segments.

Observation 3. *Canonical segments defined as in Definition 10 satisfy the following properties:*

1. *Any two canonical segments are* disjoint.
2. *The segments in $\mathsf{Canon}(\mathcal{S},u,v)$ covers all edges in $P(u,v)$, i.e.*

$$P(u,v) = \cup_{P(x,z) \in \mathsf{Canon}(\mathcal{S},u,v)} P(x,z).$$

3. *For any pair of vertices* $(u, v) \in \mathcal{S}$, *there are at most* $|\mathcal{S}|^2$ *canonical segments in* $\mathsf{Canon}(\mathcal{S}, u, v)$ *for* $|\mathcal{S}| \geq 2$.

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 $\mathcal{S}$ contributes to at most two cut vertices on $P(u, v)$. Thus there are at most $2 \cdot \binom{|\mathcal{S}|}{2} \leq |\mathcal{S}|^2$ canonical segments.

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

CANON-APSD: **An ε-DP algorithm to release all pairs counting queries**
Input: An n vertices graph, $\mathcal{G} = (\mathcal{V}, \mathcal{E}, w)$ and privacy parameter $\varepsilon > 0$.

1. Sample a set $\mathcal{S}$ 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 $(x, z) \in \mathcal{S}$ in $\mathcal{G}$, and let $P_{\mathcal{S}}$ be the set of the paths.
3. Compute $\mathsf{Canon}(\mathcal{S})$ based on the sampled vertices $\mathcal{S}$ and their shortest paths $P_{\mathcal{S}}$.
4. $\mathcal{S}$ **Perturbation:** For each canonical segment $P \in \mathsf{Canon}(\mathcal{S}, u, v)$, add an independent Laplace noise $\mathsf{Lap}(2/\varepsilon)$ to its shortest path length. Compute a function $f_{\mathcal{S}}(\cdot, \cdot)$ for counting queries between any vertices $(u, v) \in \mathcal{S}$, by summing up the noisy attributes of the canonical segments in $\mathsf{Canon}(\mathcal{S}, u, v)$.
5. **Non-$\mathcal{S}$ Perturbation:** For each edge in $\mathcal{G}$, add independent Laplace noise $\mathsf{Lap}(2/\varepsilon)$ to the edge attribute. For any vertices $u, v \in \mathcal{V}$, let $P(u, v)$ be the shortest path in $\mathcal{G}$ and $f'(u, v)$ be the sum of the noisy attributes of the edges along $P(u, v)$.
6. For each pair of vertices (u, v),
   - If there are at least two vertices in $P(u, v)$ that are in $\mathcal{S}$, let vertex x be the first one along $P(u, v)$ and z be the last one such that $x, z \in \mathcal{S}$, release $\widehat{f}(u, v) = f'(u, x) + f_{\mathcal{S}}(x, z) + f'(z, v)$.
   - Otherwise, release $\widehat{f}(u, v) = f'(u, v)$.

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

3.1 Proof of Thorem 1

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. Fix any $\mathcal{S} \subseteq \mathcal{V}$, and let $g : (2^{\mathcal{V}}, 2^{\mathcal{E}}) \to \mathbb{R}^{|\mathsf{Canon}(\mathcal{S})|}$ be the function that computes the distances for canonical segments. Then, the ℓ_1 sensitivities for g is at most 1.

Proof. The claim follows from the fact that the canonical segments are disjoint (statement 1 of Lemma 3). Concretely, recall that for two neighboring graphs $\mathcal{G} \sim \mathcal{G}' \in \mathcal{X}$, we have

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

As such,

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

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

Notably, Sect. 3.1 is already sufficient for us to prove the *privacy* of the algorithm.

Lemma 4. *The* CANON-APSD *algorithm is ε-differentially private.*

Proof. We can simply use the (basic) composition theorem (Proposition 4) to obtained the desired privacy guarantee. Note that one can view $\mathcal{S}$ Perturbation and Non-$\mathcal{S}$ perturbation as two Laplace mechanisms as defined in Definition 7. As such, we only need to prove that both perturbation mechanisms are $O(\varepsilon)$-DP.

By Sect. 3.1, the functions in steps 4. is of ℓ_1 sensitivity at most 1. As such, by Propostion 1, its output is $\frac{\varepsilon}{2}$-DP. For the input perturbation, we are directly operating on the edge attributes. As such, we have $\|w - w'\|_1 \leq 1$. Therefore, by Proposition 1, the $\mathsf{Lap}(\frac{2}{\varepsilon})$ noise gives an $\varepsilon/2$-DP algorithm.

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

Lemma 5. *With high probability, for any vertex pair $(u, v) \in V$, the difference between $f(u, v)$ and $\widehat{f}(u, v)$ by* CANON-APSD *is $O\left(\frac{1}{\varepsilon}\sqrt{\left(\frac{n}{\zeta} + \zeta^2 \log^2 n\right) \log n}\right)$. More precisely,*

$$\left| f(u, v) - \widehat{f}(u, v) \right| \leq \frac{900}{\varepsilon} \cdot \sqrt{\left(\frac{n}{\zeta} + \zeta^2 \log^2 n\right) \cdot \log n}$$

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

Proof. We start with proving a structural lemma, which powers the algorithm to decompose the error into different parts to apply the concentration inequality of Laplace noise. The following lemma will be extensively used in the paper:

Lemma 6. *For any pair of vertices (u, v), if the number of edges on the shortest path $P(u, v)$, denoted by $|P(u, v)|$, is at least $\frac{n}{\zeta}$, then, with high probability, there exist at least two vertices $(x, z) \in P(u, v)$ such that*

1. *$x \in S$ 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 $\frac{n}{\zeta}$, i.e. $|P(u, x)| \leq \frac{n}{\zeta}$ and $|P(z, v)| \leq \frac{n}{\zeta}$.*

We defer the proof Lemma 6 to Appendix C. Now, coming back to the analysis on separate parts of additive error, fix a pair of vertices $(u, v) \in \mathcal{V}$ and their shortest path $P(u, v)$, the additive noises are:

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

The second line is obtained from statements 2 and 3 of Lemma 3: to compute the all-pairs shortest distances between pair in $\mathcal{S}$, it suffices to estimate the canonical segments, and there are at most s^2 many of them. As such, in the CANON-APSD algorithm, we let each Laplace noise be with variance $b_i = 2/\varepsilon$ for all i, we again pick $\nu = \sqrt{\sum_i b_i^2}$ and $\lambda = 30\nu\sqrt{\log n} = \frac{60}{\varepsilon} \cdot \sqrt{n \log n}$. Recall that $s = 100 \log n \cdot n^{1/3}$ (since $\zeta = n^{1/3}$), which implies $\frac{2\sqrt{2}\nu}{\max_i b_i} \geq 30\sqrt{\log n}$ (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 1, which gives us

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

Therefore, with probability $1 - \frac{1}{n^3}$,

$$\left|f(u,v) - \widehat{f}(u,v)\right| \leq 30\sqrt{\log n} \cdot \nu \leq \frac{90}{\varepsilon}\sqrt{\left(\frac{n}{\zeta} + 100^2 \cdot \zeta^2 \cdot \log^2 n\right) \cdot \log n}.$$

A union bound over the above event and the high probability event in Lemma 6 gives us the desired statement.

In fact, Lemma 5 holds for any $\zeta = n^{1-\Omega(1)}$ for sufficiently large n (as long as $n^{\Omega(1)} > 900\log(n)$). We can now finalize the analysis of the additive error of the CANON-APSD algorithm.

Lemma 7. *With high probability, the* CANON-APSD *algorithm has an additive error of at most $O\left(\frac{n^{1/3}}{\varepsilon} \cdot \log^{5/6} n\right)$.*

Proof. We use Lemma 5 by setting the parameter $\zeta = \frac{1}{C} \cdot n^{1/3} \log^{-2/3} n$ with the C in Lemma 5. As such, the total additive error becomes

$$O\left(\frac{1}{\varepsilon} \cdot \sqrt{\left(\frac{n}{n^{1/3}\log^{-2/3} n} + (n^{1/3}\log^{-2/3} n)^2 \cdot \log^2 n\right) \cdot \log n}\right)$$
$$= O\left(\frac{n^{1/3}}{\varepsilon} \cdot \log^{5/6} n\right),$$

as claimed.

This concludes the proof of Theorem 1.

4 A Simple (ε, δ)-DP Algorithm for Counting Queries

Proceeding to the (ε, δ)-DP setting, we show that with the relaxation of approximate-DP, the worst case additive error can be reduced from $\widetilde{O}(n^{1/3})$ to $\widetilde{O}(n^{1/4})$, formally stated as follows.

Theorem 2. *For privacy parameters, $\varepsilon, \delta \in (0,1)$, there exists an (ε, δ)-differentially private efficient algorithm that given a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, w)$ as a range query system $(\mathcal{R} = (X, \mathcal{S}), f, w)$ such that $\mathcal{S}$ is the set of the shortest paths and f is the counting query, with high probability, outputs all pairs counting queries with additive error $O\left(\frac{n^{1/4}\log^{2/3} n \log^{1/4} \frac{1}{\delta}}{\varepsilon}\right)$. That is, the algorithm outputs an estimate $\widehat{f}(\cdot,\cdot)$ such that*

$$\Pr\left(\max_{u,v \in \mathcal{V}} |\widehat{f}(u,v) - ft(u,v)| = O\left(\frac{n^{1/4}\log^{5/4} n \log^{1/2} \frac{1}{\delta}}{\varepsilon}\right)\right) \geq 1 - \frac{1}{n}.$$

At the high level, our algorithm builds single-source shortest path trees (see formal definition in Definition 11) for each vertex sampled uniformly at random, then employs an (ε, δ)-DP algorithm for distances release in the tree graph. Notice that the construction of single-source shortest-path trees follows from folklore algorithms based on Dijkstra's algorithm, which takes $O(m + n\log(n))$ time with the classical Fibonacci heap implementation. Further, our algorithm can be easily extended to guarantee ε-differentially private with slight change of parameters, while this substitution is non-trivial for the algorithm of Muthukrishnan and Nikolov [36], and to the best of our understanding, yields suboptimal error bound.

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

We will use the following result of the (ε, δ)-DP algorithm for tree graphs (see Appendix B).

Lemma 8 ((ε,δ)-DP for tree graph). *Given a tree graph $\mathcal{G}=(\mathcal{V},\mathcal{E},w)$ and privacy parameter $\varepsilon,\delta\in(0,1)$, there exists an (ε,δ)-DP algorithm releasing shortest distances from the root vertex to the rest such that, with high probability, induce additive error at most $O\big(\frac{1}{\varepsilon}\log^{1.5}n\sqrt{\log(\frac{1}{\delta})}\big)$.*

We have three remarks for Lemma 8. First for tree graphs, our problem and the private release of all pairs shortest distances are the same – since there is a unique path between any two vertices in a tree graph. Therefore private release of all pairs shortest distances in a tree graph can be used here. Prior work for this problem ([18,45]) focused on ε-DP. Between [18,45], Fan and Li's algorithm [18] uses heavy-light decomposition of the tree, with a better error bound only when the tree is shallow. Thus we present the version of (ε,δ)-DP based on Sealfon's algorithm [45]. Second, Seafon's algorithm exploits Laplace mechanism, which is replaced by Gaussian mechanism with $\sigma^2:=1/\varepsilon^2\cdot\ln(1.25/\delta)\log n$ in Lemma 8. Third, the additive error bound for ε-DP on tree graph is $O(\frac{1}{\varepsilon}\log^{2.5}n)$ with high probability for single-source distance. Lemma 8 implies that the (ε,δ)-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 the algorithm in Lemma 8 as $\mathsf{PrivateTree}(G)$ with an input tree graph G. Also we use $\mathsf{SSSP}(v)$ 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 (ε,δ)-DP algorithm is presented above.

SSSP-ASRQ: **An (ε,δ)-DP algorithm to release all pairs counting queries**
Input: An n vertices graph, $\mathcal{G}=(\mathcal{V},\mathcal{E},w)$ and privacy parameter $\varepsilon,\delta>0$.

1. Sample a set $\mathcal{S}$ of $s=\zeta\cdot\log n$ vertices uniformly at random, where $\zeta=O(\sqrt{n}\log^{-2.5}n)$.
2. For each vertex $v\in\mathcal{S}$, compute $T(v)=\mathsf{SSSP}(v)$. Call the set of all trees T.
3. $\mathcal{S}$ **Perturbation:** For each tree $T\in\mathcal{T}$, privatize it by running $\mathsf{PrivateTree}(T)$ with the Gaussian noise $\mathcal{N}\big(\mu=0,\sigma^2:=\frac{1}{\varepsilon_0^2}\ln(1.25/\delta_0)\log n\big)$, ε_0,δ_0 will be specified later, let the output of count query be $f_{\mathsf{T}}(u,v)$.
4. **Non-$\mathcal{S}$ Perturbation:** For each edge in $\mathcal{G}$ add independent Gaussian noise $\mathcal{N}\big(\mu=0,\sigma^2:=\frac{4}{\varepsilon^2}\ln(2.5/\delta)\log n\big)$. For any vertices $u,v\in\mathcal{V}$, let $P(u,v)$ be the shortest path in $\mathcal{G}$ and $f'(u,v)$ be the sum of the noisy attributes of the edges along $P(u,v)$.
5. For each pair of vertices (u,v)
   - If at least one of u,v is in $\mathcal{S}$, release $\widehat{f}(u,v)=f_{\mathsf{T}}(u,v)$.
   - If $u,v\notin\mathcal{S}$ and the path $P(u,v)$ has one vertex $x\in\mathcal{S}$, release $\widehat{f}(u,v)=f_{\mathsf{T}}(u,x)+f_{\mathsf{T}}(x,v)$.
   - Otherwise, release $\widehat{f}(u,v)=f'(u,v)$.

4.1 Proof of Theorem 2

Our analysis mainly hinges on the concentration of Laplace random variables (Lemma 1), a corollary (Proposition 6) of strong composition theorem (Proposition 5) and the observation that any shortest path with length larger than $\frac{n}{\zeta}$ goes through at least one vertex in the sampled set $\mathcal{S}$ with high probability (Lemma 6).

Lemma 9. *The* SSSP-ASRQ *algorithm is* (ε, δ)*-differentially private.*

Proof. First observe that any edge in $\mathcal{G}$ can only appear in at most s trees ($s = |\mathcal{S}|$), since we only build one single-source shortest path tree for each vertex in $\mathcal{S}$. Therefore, the PrivateTree algorithm (Lemma 8) is applied at most s times to any edge. In $\mathcal{S}$ perturbation, the Gaussian mechanism achieves $(\varepsilon_0, \delta_0)$-DP for each tree. Pick ε_0, δ_0 such that $\varepsilon_0 = \frac{\varepsilon}{4\sqrt{2s\ln(4/\delta)}}$ and $\delta_0 = \frac{\delta}{4s}$, using a corollary of strong composition theorem (Proposition 6) on s number of PrivateTree algorithms, we have that the $\mathcal{S}$ perturbation is $(\varepsilon/2, \delta/2)$-differentially private.

Combining with the Non-$\mathcal{S}$ perturbation, which is $(\varepsilon/2, \delta/2)$-differentially private, it is straightforward to see that the SSSP-ASRQ algorithm is (ε, δ)-differentially private.

The analysis of the additive error is again, similar as in Theorem 1 and Lemma 5. 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 10. *With high probability, the* SSSP-ASRQ *algorithm has additive error at most* $O(\frac{n^{1/4}}{\varepsilon} \cdot \log^{1.25} n\sqrt{\log \frac{1}{\delta}})$

Proof. We first show that with high probability, for any vertex pair $(u, v) \in V$, $\left|f(u,v) - \widehat{f}(u,v)\right|$ released by SSSP-ASRQ is at most

$$\max\left\{O(\sqrt{(n/\zeta)\log\frac{1}{\delta}}/\varepsilon), O\left(\sqrt{s\log\frac{2s}{\delta}} \cdot \log^{1.5} n \log\frac{1}{\delta}/\varepsilon\right)\right\}$$

.

Notice that the additive error is once again decomposed into noises from 'output perturbation' ($\mathcal{S}$ perturbation) and 'input perturbation' (Non-$\mathcal{S}$ perturbation). Fix a pair of vertices $(u, v) \in \mathcal{V}$ and denote their shortest path as $P(u, v)$. By Lemma 6 and Lemma 9, the additive noises must be either of the following two cases:

1. At most $\frac{2n}{\zeta}$ independent noises sampled from $\mathcal{N}(\mu, \sigma^2)$, with $\mu = 0, \sigma^2 := \frac{4}{\varepsilon^2}\ln(2.5/\delta)\log n$.
2. At most two independent noises induced by the PrivateTree algorithm, which is upper bounded by $O(\frac{2}{\varepsilon_0}\log^{1.5} n\sqrt{\log\frac{1}{\delta_0}})$.

The first case considers the third bullet point in Step 5. of the SSSP-ASRQ algorithm. From Lemma 6, we know that the additive error is the summation of at most $\frac{2n}{\zeta}$ independent Gaussian noises. The second case considers the first and second points in Step 5. of the SSSP-ASRQ algorithm, where $\widehat{f}(u,v)$ 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 5, 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) for the first case,

$$\Pr\left[\left|f(u,v)-\widehat{f}(u,v)\right|\geq t\right]\leq 2\exp\left(-\frac{t^2}{2n/\zeta\cdot\delta^2}\right),$$

Let $t=(n/\zeta)^{1/4}\log^{0.5}n\cdot\delta$, the above probability is smaller than $\frac{1}{n^4}$. Apply union bound on all vertex pairs, then with high probability, $\left|\mathrm{d}(u,v)-\widehat{\mathrm{d}}(u,v)\right|$ for the first case is at most

$$t=\frac{(n/\zeta)^{1/2}\log^{0.5}n\cdot\delta}{s}=O\left(\frac{1}{\varepsilon}(n/\zeta)^{1/2}\sqrt{\log\frac{1}{\delta}}\right)$$

Next, we show the additive error in the second case. In the $\mathcal{S}$ perturbation that we pick the privacy parameter (ε_0,δ_0) for the Gaussian mechanism where $\varepsilon_0=\frac{\varepsilon}{4\sqrt{2s\ln(2\delta)}}$ and $\delta_0=\frac{\delta}{2s}$.

Recall Lemma 8, the additive error is at most

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

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

$$O\left(\frac{1}{\varepsilon}\sqrt{\frac{2n}{\zeta}\cdot\log\frac{1}{\delta}}\right)=O\left(n^{1/4}\log^{1.25}n\cdot\sqrt{\log\frac{1}{\delta}}\right)$$

5 Private Algorithms for the Bottleneck Edge Queries

We investigate the problem of private *bottleneck edge* queries under the range query model in this section. The problem has natural motivations in a bulk of applications where the resilience on the shortest path is quantified by a *bottleneck* attribute. For instance, in the Time-to-Stockout problem we discussed in

Sect. 1, the quantity of interest is usually the edge with the *minimum* value of the attribute among the shortest path. We show that we can release such information privately by simply applying the input perturbation technique. More formally, we have:

Theorem 3. *For privacy parameters $\varepsilon, \delta \in (0,1)$, there exist*

- *an ε-differentially private efficient algorithm that given a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, w)$ as a range query system $(\mathcal{R} = (X, \mathcal{S}), f, w)$ such that $\mathcal{S}$ is the set of the shortest paths and f is the bottleneck query, with high probability, outputs all pairs bottleneck queries with additive error $O(\frac{\log n}{\varepsilon})$.*
- *an (ε, δ)-differentially private efficient algorithm that given a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, w)$ as a range query system $(\mathcal{R} = (X, \mathcal{S}), f, w)$ such that $\mathcal{S}$ is the set of the shortest paths and f is the bottleneck query, with high probability, outputs all pairs bottleneck queries with additive error $O\left(\frac{\sqrt{\log n \log \frac{1}{\delta}}}{\varepsilon}\right)$.*

Remark 4. We remark that the bottleneck edge task cannot be trivially solved by the *top-k* selection problem in differential privacy (e.g. [13,35,43] and references therein). Note that although it is possible to directly apply top-1 selection to privately release the bottleneck edge on *a single* shortest path, the $O(n^2)$-many shortest paths may incur significant privacy loss if we simply use composition.

We now present the ε-DP and (ε, δ)-DP algorithms with the input perturbation technique first developed by [45]. Recall that $\gamma(u,v) = \min_{e \in P(u,v)} w(e)$ is the minimum edge weight on the shortest path between u and v. Both algorithms can be presented with only differences on a subroutine as follows.

Algorithms for minimum attribute edge on the shortest path

Input: A range system $\mathcal{R} = (X, \mathcal{S})$ and attribute function w, where X and w specifies a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, w)$, and the ranges $\mathcal{S}$ specifies shortest paths; a privacy budget $\varepsilon, \delta \in (0,1)$.

1. Perform the input perturbation depending on the application:
   - For ε-DP, use the Lap-perturb procedure: add Laplace noise $\mathsf{Lap}(\frac{1}{\varepsilon})$ to every output of w, and obtain $\widetilde{w}$.
   - For (ε, δ)-DP, use the Gaussian-perturb procedure: add Gaussian noise with $\sigma = \frac{\sqrt{2\log(1.25/\delta)}}{\varepsilon}$ to every output of w, and obtain $\widetilde{w}$.
2. For each shortest path $S \in \mathcal{S}$, find $e_S^\star$ the edge with the minimum attribute on each shortest path $S \in \mathcal{S}$ with the *original* attribute function w, i.e. $e_S^\star = \operatorname{argmin}_e \{w(e) \mid e \in S\}$.
3. Report $\widetilde{\gamma} = \widetilde{w}(e_S^\star)$ as the attribute of the bottleneck edge on each shortest path $S = P(u,v)$.

In other words, the whole algorithm can be framed as adding input noise to the attributes (Laplace noise for ε-DP and Gaussian noise for (ε, δ)-DP), identifying the bottleneck edge with the *original* attributes, and release the noisy attribute of that bottleneck edge. We now show that the algorithms are differentially private under their respective setting, and the additive error is small.

The Analysis of ε-DP Bottleneck Edge

The privacy guarantee follows from the input perturbation guarantee and the post-processing theorem (Proposition 3). More formally, we can show the following lemma.

Lemma 11. *The algorithm with Lap-perturb procedure is ε-differentially private.*

Proof. Let $f : \mathcal{E} \to \mathbb{R}^m$ be the attribute function. By the properties of neighboring attributes (Definition 1), it follows that the ℓ_1 sensitivity $\Delta_{f,1}$ is at most 1 since the total change of bottleneck edges can be at most 1. As such, by Propositio 1, the output of $\widetilde{w}$ is ε-DP. Since we only release information as post-processing of $\widetilde{w}$, by Proposition 3, the algorithm is ε-DP.

We now show that the additive error is bounded by $O(\frac{\log n}{\varepsilon})$ with high probability. The argument follows by using the concentration of Laplace distribution and union bound over poly(n) scenarios.

Lemma 12. *If the Lap-perturb procedure is applied, with high probability, for each pair of vertices (u, v), the difference between the output of $\widetilde{\gamma}(u, v)$ and the true bottleneck edge attribute $\gamma(u, v)$ is at most $O(\frac{\log n}{\varepsilon})$, i.e.*

$$\Pr\left(\max_{u,v\in\mathcal{V}} |\gamma(u,v) - \widetilde{\gamma}(u,v)| = O\left(\frac{\log n}{\varepsilon}\right)\right) \geq 1 - \frac{1}{n}.$$

Proof. For a fixed vertex pair (u, v), we need to take care of at most $n - 1$ edges on a shortest path. Note that for each edge (x, y) on the path $P(u, v)$, by the tail bound of Laplace distribution, the error induced by a single Laplace noise is at most $5\frac{\log n}{\varepsilon}$ with probability at least $1 - \frac{1}{n^5}$. As such, we have

$$\Pr\left(\max_{e\in P(u,v)} |w(e) - \widetilde{w}(e)| > 5 \cdot \frac{\log n}{\varepsilon}\right) \leq \frac{1}{n^4}.$$

Therefore, the additive error on the bottleneck edge is also at most $5 \cdot \frac{\log n}{\varepsilon}$ with probability at least $1 - \frac{1}{n^4}$. Applying a union bound over $\binom{n}{2}$ pairs gives us the desired statement.

The Analysis of (ε, δ)-DP Bottleneck Edge

We now turn to the algorithm for (ε, δ)-DP. Similar to the case in the ε-DP, we show that the approximate-DP [property holds by the Gaussian noise property and the post-processing theorem. The formal lemma can be shown as follows.

Lemma 13. *The algorithm with Gaussian-perturb procedure is (ε, δ)-differentially private.*

Proof. Similar to the proof of Lemma 11, we let attribute function $w : \mathcal{E} \to \mathbb{R}^m$ be the function of Definition 6. We can then bound the ℓ_2 sensitivity $\Delta_{f,2}$ of the attribute function by 1, again using the properties of neighboring attributes (Defintion 1). As such, by Proposition 2 and Proposition 3 and the right choice of σ, the algorithm is (ε, δ)-DP.

The benefit of allowing approximate-DP is a quadratic improvement on the additive error – conceptually, this follows straightforwardly by considering the lighter tail of the Gaussian distribution. We formalize the result as follows.

Lemma 14. *If the Gaussian-perturb procedure is applied, with high probability, for each pair of vertices (u, v), the difference between the output of $\widetilde{\gamma}(u, v)$ and the true bottleneck edge attribute $\gamma(u, v)$ is at most $O(\frac{\sqrt{\log n \log 1/\delta}}{\varepsilon})$, i.e.*

$$\Pr\left(\max_{u,v\in\mathcal{V}} |\gamma(u,v) - \widetilde{\gamma}(u,v)| = O\left(\frac{\sqrt{\log n \log \frac{1}{\delta}}}{\varepsilon}\right)\right) \geq 1 - \frac{1}{n}.$$

Proof. Again, for a fixed vertex pair (u, v), there are at most $n - 1$ edges among a shortest path. Note that for each edge (x, y) on the path $P(u, v)$, by the tail bound of Gaussian distribution (Lemma 2), there is

$$\Pr\left(|\widetilde{w}((x,y)) - w((x,y))| > 5\sqrt{\log n}\sigma\right) \leq \exp\left(-10 \log n\right) \leq \frac{1}{n^5}.$$

As such, with probability at least $1 - \frac{1}{n^5}$, the attribute of a single edge is only different from the original with an additive error of $5\sqrt{\log n}\sigma$. Therefore, we have

$$\Pr\left(\max_{e\in P(u,v)} |w(e) - \widetilde{w}(e)| > 5\sqrt{\log n}\sigma\right) \leq \frac{1}{n^4}.$$

By the choice of σ, we have $5\sqrt{\log n}\sigma = O(\frac{\sqrt{\log n \log 1/\delta}}{\varepsilon})$. Applying a union bound over $\binom{n}{2}$ pairs gives us the desired statement.

6 VC-Dimension of Shortest Paths Ranges and Generic Algorithms

Under the range query context, it is possible to study the VC-dimension of shortest paths in a graph using a range system. The benefit of such a perspective is that one can apply generic algorithms for private range queries, most notably by the work of Muthukrishnan and Nikolov [36]. We discuss the problem from this perspective in this section.

Recall that we say a subset $A \subseteq X$ to be shattered by $\mathcal{S}$ if each of the subsets of A can be obtained as the intersection of some $S \in \mathcal{S}$ with A, i.e., if $\mathcal{S}|_A = 2^A$. The Vapnik-Chervonenkis (VC) d of a set system $(X, \mathcal{S})$ is defined as the size of the largest subset of X that can be shattered. Formally, the definition can be described as follows.

Definition 12 (Vapnik-Chervonenkis (VC) dimension). *Let $\mathcal{R} = (X, \mathcal{S})$ be a set system and let $A \subseteq X$ be a set. We say A is shattered by $\mathcal{S}$ if $\{S \cap A \mid S \in \mathcal{S}\} = 2^A$, i.e. the union of intersections between sets in S and A covers all subsets of A. The Vapnik-Chervonenkis (VC) dimension d of $\mathcal{R}$ is defined as the size of the largest $A \subseteq X$ that can be shattered by $\mathcal{S}$.*

In an undirected graph G, the VC-dimension of (unique) shortest paths[2] is 2 [1,47]. In a directed graph, the VC-dimension of (unique) shortest paths[3] is 3 [20].

A closely-related notion is the (primal) shatter function of a set system $\mathcal{R} = (X, \mathcal{S})$ (with parameter s), which is defined as the maximum number of distinct sets in $\{A \cap S \mid S \in \mathcal{S}\}$ for some $A \subseteq X$ such that $|A| = S$. More formally, the notion can be defined as follows.

Definition 13 (Primal Shatter Function). *Let $\mathcal{R} = (X, \mathcal{S})$ be a set system, and s be a positive integer. The primal shatter function of $\mathcal{R}$, denoted as $\pi_{\mathcal{R}}(s)$, is defined as* $\max_{A:|A|=s} |\{A \cap S \mid S \in \mathcal{S}\}|$

It is well known that if the VC-dimension of a range space is d, then $\pi_{\mathcal{R}}(s) = O(s^d)$ [34]. This immediately gives a bound of $O(s^2)$ for shortest paths in undirected graphs. We now show that shortest paths in directed graphs enjoys the same bound as well despite having a higher VC-dimension.

Lemma 15. *For a range query system $\mathcal{R} = (X, \mathcal{S})$ defined by shortest paths in (both directed and undirected) graphs, the primal shatter function is $\pi_{\mathcal{R}}(s) = O(s^2)$ for any positive integer s.*

[2] Any set of three vertices $\{u, v, w\}$ cannot be shattered: if one vertex w stays on the shortest path of the other two vertices u, v, then one cannot obtain the subset u, v; if none of them stays on the shortest path of the other two, then one cannot obtain the subset u, v, w .

[3] In a directed graph, a directed cycle of u, v, w can be shattered.

Proof. Take any set A of size s, any shortest path either does not contain any vertex in A, or contains a first vertex $x \in A$ and the last vertex $y \in A$ along the path. Notice that x, y might be the same vertex. Thus $\mathcal{S}|_A$ contains the subset of A as $A \cup S(x,y), \forall x, y \in A$ where $S(x,y)$ is the set of vertices on the shortest path from x to y. Therefore $\mathcal{S}|_A$ has at most $O(s^2)$ elements.

The benefit of understanding the VC-dimension and the primal shatter function for shortest system is that we can use generic algorithms for private range queries. In particular, Muthukrishnan and Nikolov [36] have developed a differentially private mechanism for answering range queries of bounded VC-dimension. The guarantee of the algorithm is as follows.

Proposition 7 (Muthukrishnan-Nikolov algorithm [36], rephrased). *Let $(\mathcal{R} = (X, \mathcal{S}), f)$ be a range query system, where f is the counting query and the primal shatter function of $\mathcal{R}$ is $\pi_{\mathcal{R}}(s) = O(s^d)$ for any s. There exists an algorithm that outputs* all queries *with*

- *Expected average squared error of* $O\left(\frac{n^{1-1/d}\log\frac{1}{\delta}}{\varepsilon^2}\right)$;
- *With probability at least* $1-\beta$, *worst case squared error of* $O\left(\frac{n^{1-1/d}\log\frac{1}{\delta}\log\frac{n}{\beta}}{\varepsilon^2}\right)$.

The algorithm is (ε, δ)-differentially private.

Using the algorithm of Proposition 7, the bound on the primal shatter functions of Lemma 15, and the fact that counting query sums up the attributes on the shortest paths, we can obtain an (ε, δ)-DP result with additive error $O(n^{1/4}\log^{1/2}(n)\log^{1/2}(1/\delta)/\varepsilon)$ with high constant probability. This matches our (ε, δ)-DP result in Theorem 2 up to lower order terms.

Remark 5. Although it is possible to recover the bound of Theroem 2 using Proposition 7 as a black-box, our constructions still enjoy multiple advantages. In particular, the construction of Proposition 7 does *not* give any non-trivial bound for ε-DP, and it is not trivial to adapt it to pure-DP within the framework. Furthermore, the algorithm of Proposition 7 requires to find a maximal set of ranges with the minimum symmetric differences on different levels, and by the packing lemma bound in [36], it appears that a straightforward implementation could take $\Theta(n^4)$ time in the worst case. On the other hand, our constructions for both Theroem 1 and Theroem 2 can be implemented in $\widetilde{O}(n^2)$ time. Finally, the algorithm of Proposition 7 is much more complicated and counter-intuitive, and our algorithm enjoys much better simplicity.

7 Conclusion and Future Work

We study the private release of shortest path queries under the range query context in this paper, where the graph topology and the shortest paths are public,

and the attributes on the graphs (which do *not* affect shortest paths) are subject to privacy protection. Our upper bounds cannot be applied to the (harder) problem of private release of all pairs shortest distances [45]. Thus improving the bounds of private range query problem (with upper bound $\widetilde{O}(n^{1/3})$ for ε-DP and $\widetilde{O}(n^{1/4})$ for (ε, δ)-DP) and all pairs shortest distances release (with upper bound $\widetilde{O}(n^{2/3})$ for ε-DP and $\widetilde{O}(n^{1/2})$ for (ε, δ)-DP), where both have a lower bound of $\Omega(n^{1/6})$, remains an interesting open problem. Furthermore, since our algorithms are simple to implement, the empirical performances of our algorithms could be another future research direction.

Acknowledgements. We would like to thank Adam Sealfon, Shyam Narayanan, Justin Chen, Badih Ghazi, Ravi Kumar, Pasin Manurangsi, Jelani Nelson and Yinzhan Xu for useful discussion and suggestions. Deng and Gao have been partially supported by NSF through CCF-2118953, CNS-2137245, CCF-2208663, and CRCNS-2207440.

A Range Query on All Paths

When we allow queries along any path in a graph and require differential privacy guarantees, the following result provides a lower bound of $\Omega(n)$ on the additive error. To show the lower bound, we first consider a range query formulated by the incidence matrix A, with m columns corresponding to the m edges in the graph G and rows corresponding to all queries. A query along path P is represented by a row in the matrix with an element of 1 corresponding to edge e if e is on P and 0 otherwise. We will then talk about the discrepancy of matrix A.

The classical notion of discrepancy of a matrix A is the minimum value of $||Ax||_\infty$, where x is a vector with elements taking values $+1$ or -1. And the hereditary discrepancy of A is the maximum discrepancy of A limited on any subset of columns. As shown in [36], both discrepancy and hereditary discrepancy of A provides a lower bound on the additive error of differentially private range query using incidence matrix A.

Theorem 4. *A (ε, δ)-differential privacy mechanism that answers range queries where ranges are defined on any path of an input graph has to incur additive error of $\Omega(n)$.*

Proof. Consider a graph of $n+1$ vertices $v_1, v_2, \cdots, v_{n+1}$ and $2n$ edges. Between vertices v_i and v_{i+1} there are two parallel edges e_i and e_i'. On this graph there are 2^n paths from v_1 to v_{n+1}. We consider only queries along these paths and the incidence matrix is a tall matrix A of 2^n rows and $2n$ columns, corresponding to the $2n$ edges in the graph. Now we take a submatrix of A with only the columns corresponding to edges e_i. This gives a matrix A' of $2^n \times n$, with the rows corresponding to all subsets of $[n]$. A' has discrepancy of $\Omega(n)$. To see that, consider the specific vector x that minimizes $||A'x||_\infty$. Suppose x has k entries of $+1$ and $n-k$ entries of -1. Without loss of generality, we assume $k \geq n/2$, The row of A that has value 1 corresponding to the positive entries of x and value 0 corresponding to the negative entries of x, gives a value of $k \geq n/2$.

Thus $||Ax||_\infty$ is at least $n/2$. This means that the hereditary discrepancy of A is at least $\Omega(n)$.

By the same argument and use Corollary 1 in [36], we conclude that any (ε, δ)-differentially private mechanism has to have error of $\Omega(n)$.

B Proof of Lemma 8 – (ε, δ) Algorithm for Tree Graphs

Proof. We first claim that we can answer all pairs shortest distance on a tree with (α, β)-accuracy for

$$\alpha = O\left(\frac{1}{\varepsilon}\log n\sqrt{\log\left(\frac{n}{\beta}\right)\log\left(\frac{1}{\delta}\right)}\right)$$

showing the utility guarantee of Lemma 8. 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 $O\left(\frac{1}{\varepsilon}\log^{1.5} n\sqrt{\log\left(\frac{1}{\delta}\right)}\right)$.

In Sealfon's algorithm [45], 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 $\mathsf{Lap}(\log n/\varepsilon)$ 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 privatizes each of the subtrees recursively. Using Sealfon's algorithm, we know that for a given root node v_0, computing the single source (with the 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 compositions we need is $O(\log n)$.

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

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

$$\mathbb{E}[\widehat{d}(u, v_0)] = d(u, v_0).$$

Using the standard concentration of Gaussian distribution [50] implies that

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

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

$$\begin{aligned}&\Pr\left(\left|\widehat{d}(u, v_0) - \mathbb{E}\left[\widehat{d}(u, v_0)\right]\right| > \frac{C}{\varepsilon} \log n \sqrt{\log\left(\frac{2n}{\beta}\right) \log\left(\frac{1}{\delta}\right)}\right) \\ \leq & 2e^{-C \log(2n/\beta)} \leq \frac{\beta}{n}.\end{aligned}$$

Now union bound gives that

$$\Pr\left(\max_{u \in \mathcal{V}} \left|\widehat{d}(u, v_0) - d(u, v_0)\right| \leq \frac{C}{\varepsilon} \log n \sqrt{\log\left(\frac{n}{\beta}\right) \log\left(\frac{1}{\delta}\right)}\right) \geq 1 - \beta.$$

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

$$\widehat{d}(u, v) = \widehat{d}(u, v^*) + \widehat{d}(v, v*) - 2\widehat{d}(z, v^*).$$

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,

$$\Pr\left(\max_{u,v \in \mathcal{V}} |\widehat{\mathrm{d}}(u, v) - \mathrm{d}(u, v)| = O\left(\frac{1}{\varepsilon} \log n \sqrt{\log\left(\frac{n}{\beta}\right) \log\left(\frac{1}{\delta}\right)}\right)\right) \geq 1 - \beta$$

completing the proof of the claim.

C Proof of Lemma 6

Proof. (Proof of Lemma 6). The lemma is proved by a simple application of the Chernoff bound. For each path $P(u, v)$ with more than $\frac{n}{\zeta}$ edges, let v' be the $\left(\frac{n}{\zeta} + 1\right)$-th vertices on the path $P(u, v)$ from u. Similarly, let u' be the $\left(\frac{n}{\zeta} + 1\right)$-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 $\frac{n}{\zeta}$ vertices, and we are sampling $s = 100 \log n \cdot \zeta$ 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

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

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

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

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.

D A Remark on Range Query Shortest Path Lower Bound

For counting range queries with (ε, δ)-DP guarantee, there is a lower bound of $\Omega(n^{1/6})$ on the additive error, adapted from the construction of the lower bound for private all pairs shortest distances [10]. Specifically, the construction uses a graph where vertices are points in the plane and edges map to line segments between two points that do not contain other vertices. The edge length is the Euclidean length and therefore the shortest path between two vertices is the path corresponding to a straight line. The range query problem can be now formulated as a (special case) of linear queries, as in Sect. 6 and Sect. A, where the matrix A corresponds to the incidence matrix of the shortest paths and the edges in the graph. It is known that this matrix has a discrepancy lower bound of $\Omega(n^{1/6})$ [34]. By the connection of the discrepancy and linear query lower bounds [36], this is a lower bound for our problem.

References

1. Abraham, I., Delling, D., Fiat, A., Goldberg, A.V., Werneck, R.F.: VC-dimension and shortest path algorithms. In: Aceto, L., Henzinger, M., Sgall, J. (eds.) ICALP 2011. LNCS, vol. 6755, pp. 690–699. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-22006-7_58
2. Acs, G., Castelluccia, C., Chen, R.: Differentially private histogram publishing through lossy compression. In: 2012 IEEE 12th International Conference on Data Mining, pp. 1–10. IEEE (2012)
3. Beimel, A., Moran, S., Nissim, K., Stemmer, U.: Private center points and learning of halfspaces. In: Conference on Learning Theory, pp. 269–282. PMLR (2019)

4. Beimel, A., Nissim, K., Stemmer, U.: Private learning and sanitization: pure vs. approximate differential privacy. In: Raghavendra, P., Raskhodnikova, S., Jansen, K., Rolim, J.D.P. (eds.) APPROX/RANDOM -2013. LNCS, vol. 8096, pp. 363–378. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-40328-6_26
5. Bhaskara, A., Dadush, D., Krishnaswamy, R., Talwar, K.: Unconditional differentially private mechanisms for linear queries. In: Proceedings of the forty-fourth annual ACM Symposium on Theory of computing, pp. 1269–1284 (2012)
6. Blum, A., Dwork, C., McSherry, F., Nissim, K.: Practical privacy: the SuLQ framework. In: Proceedings of the Twenty-Fourth ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems, pp. 128–138 (2005)
7. Blum, A., Ligett, K., Roth, A.: A learning theory approach to noninteractive database privacy. J. ACM **60**(2), 12 (2013)
8. Bun, M., Ullman, J., Vadhan, S.: Fingerprinting codes and the price of approximate differential privacy. SIAM J. Comput. **47**(5), 1888–1938 (2018)
9. Chan, T.H.H., Shi, E., Song, D.: Private and continual release of statistics. ACM Trans. Inf. Syst. Secur. (TISSEC) **14**(3), 1–24 (2011)
10. Chen, J.Y., et al.: Differentially private all-pairs shortest path distances: improved algorithms and lower bounds. In: 2023 Symposium on Discrete Algorithm (SODA 2023) (2023)
11. Cormode, G., Kulkarni, T., Srivastava, D.: Answering range queries under local differential privacy. Proc. VLDB Endowment **12**(10), 1126–1138 (2019)
12. Cormode, G., Procopiuc, C., Srivastava, D., Shen, E., Yu, T.: Differentially private spatial decompositions. In: 2012 IEEE 28th International Conference on Data Engineering, pp. 20–31. IEEE (2012)
13. Durfee, D., Rogers, R.M.: Practical differentially private top-k selection with pay-what-you-get composition. In: Wallach, H.M., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E.B., Garnett, R. (eds.) Advances in Neural Information Processing Systems, vol. 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019 (December), pp. 8–14. Vancouver, BC, Canada, pp. 3527–3537 (2019). https://proceedings.neurips.cc/paper/2019/hash/b139e104214a08ae3f2ebcce149cdf6e-Abstract.html
14. Dwork, C., Kenthapadi, K., McSherry, F., Mironov, I., Naor, M.: Our data, ourselves: privacy via distributed noise generation. In: Vaudenay, S. (ed.) EUROCRYPT 2006. LNCS, vol. 4004, pp. 486–503. Springer, Heidelberg (2006). https://doi.org/10.1007/11761679_29
15. Dwork, C., McSherry, F., Nissim, K., Smith, A.: Calibrating noise to sensitivity in private data analysis. J. Priv. Confidentiality **7**(3), 17–51 (2016)
16. Dwork, C., Roth, A.: The algorithmic foundations of differential privacy. Found. Trends® Theor. Comput. Sci. **9**(3–4), 211–407 (2014)
17. Dwork, C., Rothblum, G.N., Vadhan, S.: Boosting and differential privacy. In: 2010 IEEE 51st Annual Symposium on Foundations of Computer Science, pp. 51–60. IEEE (2010)
18. Fan, C., Li, P.: Distances release with differential privacy in tree and grid graph. arXiv preprint arXiv:2204.12488 (2022)
19. Fan, C., Li, P., Li, X.: Breaking the linear error barrier in differentially private graph distance release. arXiv preprint arXiv:2204.14247 (2022)
20. Funke, S., Nusser, A., Storandt, S.: On k-path covers and their applications. Proc. VLDB Endowment **7**(10), 893–902 (2014)
21. Ghane, S., Kulik, L., Ramamoharao, K.: A differentially private algorithm for range queries on trajectories. Knowl. Inf. Syst. **63**(2), 277–303 (2021)

22. Ghosh, A., Ding, J., Sarkar, R., Gao, J.: Differentially private range counting in planar graphs for spatial sensing. In: Proceedings of the 39th Annual IEEE International Conference on Computer Communications (INFOCOM 2020), pp. 2233–2242 (2020)
23. Gupta, A., Roth, A., Ullman, J.: Iterative constructions and private data release. In: Cramer, R. (ed.) TCC 2012. LNCS, vol. 7194, pp. 339–356. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-28914-9_19
24. Hardt, M., Ligett, K., McSherry, F.: A simple and practical algorithm for differentially private data release. In: Advances in Neural Information Processing Systems, vol. 25 (2012)
25. Hardt, M., Rothblum, G.N.: A multiplicative weights mechanism for privacy-preserving data analysis. In: 2010 51st Annual IEEE Symposium on Foundations of Computer Science (FOCS), pp. 61–70. IEEE (2010)
26. Hardt, M., Talwar, K.: On the geometry of differential privacy. In: Proceedings of the Forty-Second ACM Symposium on Theory of Computing, pp. 705–714. ACM (2010)
27. Hay, M., Li, C., Miklau, G., Jensen, D.: Accurate estimation of the degree distribution of private networks. In: 2009 Ninth IEEE International Conference on Data Mining, pp. 169–178. IEEE (2009)
28. Hay, M., Rastogi, V., Miklau, G., Suciu, D.: Boosting the accuracy of differentially-private histograms through consistency. arXiv preprint arXiv:0904.0942 (2009)
29. Hong, Y.C., Chen, J.: Graph database to enhance supply chain resilience for industry 4.0. IJISSCM **15**(1), 1–19 (2022)
30. Kaplan, H., Mansour, Y., Stemmer, U., Tsfadia, E.: Private learning of halfspaces: simplifying the construction and reducing the sample complexity. Adv. Neural. Inf. Process. Syst. **33**, 13976–13985 (2020)
31. Li, C., Hay, M., Rastogi, V., Miklau, G., McGregor, A.: Optimizing linear counting queries under differential privacy. In: Proceedings of the Twenty-Ninth ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems, pp. 123–134. ACM (2010)
32. Li, C., Miklau, G.: Optimal error of query sets under the differentially-private matrix mechanism. In: Proceedings of the 16th International Conference on Database Theory, pp. 272–283 (2013)
33. Li, Y., Purcell, M., Rakotoarivelo, T., Smith, D., Ranbaduge, T., Ng, K.S.: Private graph data release: a survey. ACM Comput. Surv. **55**(11), 1–39 (2023). https://doi.org/10.1145/3569085
34. Matoušek, J.: Geometric Discrepancy. Springer, Berlin Heidelberg (1999). https://doi.org/10.1007/978-3-642-03942-3
35. McSherry, F., Talwar, K.: Mechanism design via differential privacy. In: 48th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2007), pp. 94–103. IEEE (2007)
36. Muthukrishnan, S., Nikolov, A.: Optimal private halfspace counting via discrepancy. In: Proceedings of the Forty-Fourth Annual ACM Symposium on Theory of Computing, pp. 1285–1292 (2012)
37. Nikolov, A., Talwar, K., Zhang, L.: The geometry of differential privacy: the sparse and approximate cases. In: Proceedings of the Forty-Fifth Annual ACM Symposium on Theory of Computing, pp. 351–360 (2013)
38. Ogbuke, N.J., Yusuf, Y.Y., Dharma, K., Mercangoz, B.A.: Big data supply chain analytics: ethical, privacy and security challenges posed to business, industries and society. Prod. Plan. Control **33**(2–3), 123–137 (2022)
39. Pourhabibi, T., Ong, K.L., Kam, B.H., Boo, Y.L.: Fraud detection: a systematic literature review of graph-based anomaly detection approaches. Decis. Support Syst. **133**, 113303 (2020)

40. Qardaji, W., Yang, W., Li, N.: Differentially private grids for geospatial data. In: 2013 IEEE 29th International Conference on Data Engineering (ICDE), pp. 757–768. IEEE (2013)
41. Qardaji, W., Yang, W., Li, N.: Understanding hierarchical methods for differentially private histograms. Proc. VLDB Endowment **6**(14), 1954–1965 (2013)
42. Qardaji, W., Yang, W., Li, N.: Priview: practical differentially private release of marginal contingency tables. In: Proceedings of the 2014 ACM SIGMOD International Conference on Management of Data, pp. 1435–1446 (2014)
43. Qiao, G., Su, W.J., Zhang, L.: Oneshot differentially private top-k selection. In: Meila, M., Zhang, T. (eds.) Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18–24 July 2021, Virtual Event. Proceedings of Machine Learning Research, vol. 139, pp. 8672–8681. PMLR (2021). http://proceedings.mlr.press/v139/qiao21b.html
44. Sadigurschi, M., Stemmer, U.: On the sample complexity of privately learning axis-aligned rectangles. Adv. Neural. Inf. Process. Syst. **34**, 28286–28297 (2021)
45. Sealfon, A.: Shortest paths and distances with differential privacy. In: Proceedings of the 35th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems, pp. 29–41 (2016)
46. Sharma, S., Chen, K., Sheth, A.: Toward practical privacy-preserving analytics for IoT and cloud-based healthcare systems. IEEE Internet Comput. **22**(2), 42–51 (2018)
47. Tao, Y., Sheng, C., Pei, J.: On k-skip shortest paths. In: Proceedings of the 2011 ACM SIGMOD International Conference on Management of data, SIGMOD 2011, pp. 421–432. Association for Computing Machinery, New York (2011)
48. Toth, C.D., O'Rourke, J., Goodman, J.E.: Handbook of Discrete and Computational Geometry (2017)
49. Tukey, J.W.: Mathematics and the picturing of data. In: Proceedings of the International Congress of Mathematicians, Vancouver, 1975, vol. 2, pp. 523–531 (1975)
50. Wainwright, M.J.: High-Dimensional Statistics: A Non-asymptotic Viewpoint, vol. 48. Cambridge University Press, Cambridge (2019)
51. Xiao, X., Wang, G., Gehrke, J.: Differential privacy via wavelet transforms. IEEE Trans. Knowl. Data Eng. **23**(8), 1200–1214 (2010)
52. Xiao, Y., Xiong, L., Fan, L., Goryczka, S.: DPCube: Differentially private histogram release through multidimensional partitioning. arXiv preprint arXiv:1202.5358 (2012)
53. Zhang, J., Xiao, X., Xie, X.: Privtree: A differentially private algorithm for hierarchical decompositions. In: Proceedings of the 2016 International Conference on Management of Data, pp. 155–170 (2016)