Differentially Private Counting Queries on Approximate Shortest Paths

Jesse Campbell[1] and Chunjiang Zhu[2]

[1] Duke Kunshan University, Kunshan, Jiangsu, China
jesse.campbell@duke.edu
[2] University of North Carolina at Greensboro, Greensboro, NC, USA
chunjiang.zhu@uncg.edu

Abstract. Given a weighted graph $G = (V, E, \mathbf{w})$ with some private edge attribute function $f : E \to \mathbb{R}^+$, the differentially private counting query problem aims to release the sum of the private edge attributes along a desired path (or set of paths) in G with differential privacy. In this work, we introduce the differentially private counting query problem on approximate shortest paths. A nearly-optimal algorithm with $\widetilde{\Theta}(n^{1/4})$ error is known for releasing the counting queries along all true shortest paths in G [10, 6]. We extend this result by showing if G admits a *t-collective tree spanner* with η_t spanning trees, then for each pair of vertices in V, we can release the counting query over a t-approximate shortest path with $\widetilde{O}(\eta_t/\epsilon)$ additive error in the ε-DP case, and $\widetilde{O}(\sqrt{\eta_t}/\epsilon)$ in the (ε, δ)-DP case. As an example of this, we use the collective tree $O(k \log \log n)$-spanner of [1] to give an algorithm which, for each pair of vertices, releases counting queries over a $O(k \log \log n)$-approximate shortest path in G with $\widetilde{O}(\sqrt{kn^{1/k}}/\epsilon)$ error. Our result is based on the polylogarithmic error in the differential private release of attributes along paths in trees, which is much lower than the $\widetilde{\Omega}(n^{1/4})$ error in the general case.

1 Introduction

Differential privacy (DP) is a mathematical framework which quantifies the leakage of sensitive user information by an algorithm. The definition of differential privacy requires that the probability of a certain outcome being attained from a dataset without the contribution of any single individual is very close to the probability the same outcome is attained from a dataset with the individual's contribution restored. Differential privacy has seen applications in industry and government agencies, with algorithms satisfying differential privacy being adopted by organizations such as Google [3, 5], Microsoft [11], and the US Census Bureau [23, 22].

The *counting query* problem takes on input a weighted, connected graph $G = (V, E, \mathbf{w})$ and some edge attribute function $f : E \to \mathbb{R}^+$. The goal is to release the sum of the attributes along a set of paths in G. Naturally, the *differentially private* counting query on a set of paths aims to release the counting

queries over the set of paths with differential privacy, where the edge attributes are kept private (and are not dependent on the public edge weights). Previous work on this problem has shown it to be $\widetilde{\Theta}(n^{1/4})$-accurate for the set of (unique) shortest paths on general graphs (where $n = |V|$ and $\widetilde{\Theta}(\cdot)$ hides a polylogarithmic factor), for sufficiently small $\varepsilon, \delta > 0$ [10, 6].

Calculating private estimates of counting queries over shortest path systems is motivated by many real-world applications. For instance, consider a graph where edges represent transactions between financial entities, and the edge attributes model the private value of the transactions. Fraud researchers could plausibly query this graph along certain paths to check for abnormal activity. When considering the *shortest path* between two entities, however, the additive error lower bound of $\Omega(n^{1/4})$ may be too high. As another example, consider a graph which models a network of healthcare facilities for the application of patient transfers, with the edge weights corresponding to public travel times between facilities. Suppose the private edge attributes correspond to the number of patients *en route* between facilities. The lower bound of $\widetilde{\Omega}(n^{1/4})$ additive error in releasing the private edge attributes may be too high. It would be useful to have a way of determining a more accurate estimate of patients in-transfer between facilities to get their expected capacities towards a successful patient transfer, at the cost of taking a slightly longer path between facilities. Therefore, it is well-motivated to improve the additive error for applications where the importance of being on a shortest path is small compared to the importance of having an accurate estimate of the private edge attributes. We believe this relaxed problem shares many common practical applications as the original problem [10], e.g., in financial transaction networks, transportation networks, and supply chain networks.

In this setting, we provide a new privacy framework which gives the user a way of releasing counting queries over *approximate* shortest paths while greatly improving the additive error bounds. In specific, we provide an explicit way of releasing counting queries over $O(k \log \log n)$-approximate shortest paths with $\widetilde{O}(kn^{1/k}/\epsilon)$ additive error in the ε-DP case, and $\widetilde{O}(\sqrt{kn^{1/k}}/\epsilon)$ additive error in the (ε, δ)-DP case.

Our results will make use of the notion of *collective tree spanners*. A graph spanner is a sparse subgraph of a graph which preserves approximate pairwise distances. In specific, if H is an α-spanner of G, then the distance between any two vertices in H is at most α times their distance in G, where α is referred to as the *stretch*. In 2004, the notion of *collective tree spanners* was introduced as an extension of *tree t-spanners*, which consist of a single tree and are the sparsest type of graph spanner [2]. A collective tree spanner is a collection of spanning trees of G such that for every pair of vertices, there is a single tree which approximates their distance [15]. We note that the idea of collective tree spanners are highly related to the notion of *tree covers*, as introduced by [24]. A tree cover is a collection of tree subgraphs whose union spans the entire vertex set. Unlike a collective tree spanner, the individual trees in a tree cover are not required to span the entire vertex set.

1.1 Our Results

In this work, we introduce the notion of the *differentially private counting query on approximate shortest paths* (DPCQ on ASP) problem. In particular, we show that we are able to greatly improve upon the lower bound of $\widetilde{\Omega}(n^{1/4})$ for the additive error in general graphs by allowing a counting query to be on a $O(k \log\log n)$ *approximate* shortest path. Due to the key observation that we can release differentially private counting query estimates on tree graphs with $O(\text{polylog}(n))$ error via a recursive algorithm [29, 10], we can release a private estimate of the counting queries over all trees in a collective tree spanner with additive error proportional to the number of trees. Consequently, we can release an estimate over an approximate shortest path for every pair of vertices in V with low error.

Theorem 1. *Let* $\mathbf{T}$ *be a* t*-collective tree spanner of* G *such that* $|\mathbf{T}| = \eta_t$*. For* $\gamma \in (0, 0.5]$ *and* $\varepsilon \in (0, 1]$*, there is an* ε*-DP algorithm for releasing the counting query between* $u, v \in V$ *on a* t*-approximate shortest path in* G *that is* $O(\eta_t \cdot \log^{2.5}(n) \cdot \log(1/\gamma)/\varepsilon)$*-accurate with probability* $1 - \gamma$*.*

Theorem 2. *Let* $\mathbf{T}$ *be a* t*-collective tree spanner of* G *such that* $|\mathbf{T}| = \eta_t$*. For* $\gamma \in (0, 0.5]$ *and* $\varepsilon, \delta \in (0, 1]$*, there is an* (ε, δ)*-DP algorithm for releasing the counting query between* $u, v \in V$ *on a* t*-approximate shortest path that is* $O(\sqrt{\eta_t} \cdot \log^2(n) \cdot \log(1/\delta) \cdot \sqrt{\log(1/\gamma)}/\varepsilon)$*-accurate with probability* $1 - \gamma$*.*

Based on Ramsey spanning trees which are a collective tree $O(k \log\log n)$-spanner with $kn^{1/k}$ trees [1], we immediately arrive at the following results.

Corollary 1. *For* $\gamma \in (0, 0.5]$ *and* $\varepsilon \in (0, 1]$*, there exists an* ε*-DP algorithm for releasing the counting query between* $u, v \in V$ *on a* $O(k \log\log n)$*-approximate shortest path in* G *that is* $O(kn^{1/k} \cdot \log^{2.5}(n) \cdot \log(1/\gamma)/\varepsilon)$*-accurate with probability* $1 - \gamma$*.*

Corollary 2. *For* $\gamma \in (0, 0.5]$ *and* $\varepsilon, \delta \in (0, 1]$*, there exists an* (ε, δ)*-DP algorithm for releasing the counting query between* $u, v \in V$ *on a* $O(k \log\log n)$*-approximate shortest path that is* $O(\sqrt{kn^{1/k}} \cdot \log^2(n) \cdot \log(1/\delta) \cdot \sqrt{\log(1/\gamma)}/\varepsilon)$*-accurate with probability* $1 - \gamma$*.*

1.2 Related Work

Differentially Private Range Query In 2023, Deng et al. [10] were the first to study the application of differential privacy to the range query problem on shortest paths. We follow their DP framework where two graphs are neighboring if they share the graph topology and edge weights, and the vectors of edge attributes differ by a most 1 in the l_1-distance. In particular, [10] showed an algorithm for DP counting queries on shortest paths that is $O(n^{1/3} \cdot \log^{5/6}(n)/\varepsilon)$-accurate in the ε-DP case and $O(n^{1/4} \cdot \log^{2/3}(n) \cdot \log^{1/4}(1/\delta)/\varepsilon)$-accurate in the (ε, δ)-DP case.

The application of differential privacy to graph problems was first studied by Hay et al. [25] in 2009, by giving a mechanism to release a private estimate of the degree distribution in a graph. In their work, they introduce the notions of node- and edge-differential privacy, which defines neighboring graphs based on their underlying graph topology. Algorithms with node- and edge-differential privacy have been studied considerably in recent literature [28, 26, 27]. Adam Sealfon [29] was the first to study the application of differential privacy to the all-pairs shortest distances (APSD) problem [9, 20, 21], defining neighboring graphs based on the l_1 distance of their private edge weight vectors.

Furthermore, Chen et al. [9] proved a lower bound for the additive error of $\Omega(n^{1/6})$ in any algorithm for DP all-pairs distances release on general graphs, for a sufficiently small $\varepsilon, \delta > 0$, by exploiting a reduction from the *linear queries* problem to APSD, and a hereditary discrepancy lower bounded provided by Chazelle and Lvov [8]. It was noted by Deng et al. [10] that this lower-bound technique also applies to the DP range query on shortest paths problem. Soon after, Bodwin et al. [6] improved this lower bound to $\Omega(n^{1/4}/\sqrt{\log n})$ as a result of a better discrepancy lower bound, proving, together with the upper bound by [10], that the DP range query on shortest paths problem is $\widetilde{\Theta}(n^{1/4})$-accurate for general graphs.

Collective Tree Spanners The notion of collective tree spanners was first introduced in 2004 by Dragan et al. [15], who gave constructions of *additive* collective tree spanners for chordal graphs, co-comparability graphs, and more. Since then, additive collective tree spanners have been developed in the literature, with many results on specific families of graphs such as graphs with bounded parameters [13], bounded tree-breadth graphs [12], and more [16, 14]. *Multiplicative collective tree spanners* were first (informally) studied in 2001 by Gupta et al. [24] as an application to a graph routing problem, showing that all *planar* graphs have a collective multiplicative tree 1-spanner with $O(\sqrt{n})$ trees, and a 3-spanner with $O(\log(n))$ trees. Recently, the topic of multiplicative collective tree spanners has seen an increase in the literature due to the notion of *tree padding spanners*, which is an instance of the collective tree spanner problem such that each vertex has a single tree for which all the α-approximate paths are contained. Abraham et al. [1] gave an algorithm which constructs a tree padding $(k \log\log n)$-spanner which is a collection of $kn^{1/k}$ spanning trees. Similarly, Bartal et al. [4] gave constructions of tree covers for metric spaces with a small amount of trees for general, doubling, and planar metrics in the normal and Ramsey (analogous to the tree padding spanner case for graphs) cases.

2 Preliminaries

First, we present the definition of *collective tree spanners*. For $x, y \in V$, we let $d_G(x, y)$ to be the shortest path distance between x and y in G.

Definition 1. *A collection of spanning trees* $\mathbf{T}$ *of* G *is said to be an* α*-collective tree spanner of* G *if for every* $u, v \in V$*, there is a tree* $\mathcal{T} \in \mathbf{T}$ *such that* $d_{\mathcal{T}}(u, v) \leq \alpha \cdot d_G(u, v)$.

Next, we define the notion of what it means for a dataset to differ in the contribution of one individual. In our case, as in [10], isomorphic, undirected, weighted graphs are neighbors if their edge attribute vectors differ by at most 1 in the l_1 norm.

Definition 2. *Two isomorphic graphs* $G_1, G_2 = (V, E, \mathbf{w})$ *with edge attribute functions* $f_1, f_2 : E \to \mathbb{R}^+$ *are said to be neighboring if*

$$\sum_{e \in E} |f_1(e) - f_2(e)| \leq 1$$

Moreover, we define the notion of the sensitivity of a function as the most the function can differ over neighboring datasets.

Definition 3. *The* l_1 *sensitivity of* $\mathcal{F} : \mathcal{X} \to \mathbb{R}^D$ *is defined as*

$$\Delta_1(f) \coloneqq \max_{X, X'} \|\mathcal{F}(X) - \mathcal{F}(X')\|_1$$

where X, X' *are neighboring datasets.*

We are now ready to present the formal definition of *differential privacy.*

Definition 4. *An algorithm* $\mathcal{M} : \mathcal{X} \to \mathbb{R}^D$ *is said to be* (ε, δ)*-differentially private if, for all outcomes* $S \subseteq \mathbb{R}^D$ *and neighboring datasets* X, X'*,*

$$\mathbb{P}[\mathcal{M}(X) \in S] \leq e^{\varepsilon} \cdot \mathbb{P}[\mathcal{M}(X') \in S] + \delta$$

Furthermore, we present the Laplace mechanism, which is one of the primary tools in the theory of differential privacy. In specific, the Laplace mechanism describes how much one needs to perturb a released statistic in order to guarantee it is ε-DP.

Lemma 1 (Laplace Mechanism, [19]). *Given any function* $f : \mathcal{X} \to \mathbb{R}^k$*, the Laplace mechanism on input* $X \in \mathcal{X}$ *independently samples* $Y_1, ..., Y_k$ *according to* $Lap(\Delta_1(f)/\varepsilon)$ *and outputs,*

$$\mathcal{M}_{f,\varepsilon}(X) = f(X) + (Y_1, ..., Y_k)$$

The Laplace mechanism is ε*-differentially private.*

We also present a useful result about the concentration of independent Laplace random variables.

Lemma 2 ([7]). *Let $X_1, ..., X_t$ be independent random variables distributed according to Lap(b), and let $X = X_1 + ... + X_t$. Then for all $\gamma \in (0, 1)$, with probability at least $1 - \gamma$ we have,*

$$|X| < O(b\sqrt{t} \log (1/\gamma))$$

Next, we present here two key results about the composition of differentially private systems.

Lemma 3 (Basic Composition, [19, 17]). *Let $\varepsilon, \delta \in [0, 1]$ and $k \in \mathbb{N}$. If we run k mechanisms where each mechanism is $(\varepsilon/k, \delta/k)$-differentially private, then the entire algorithm is (ε, δ)-differentially private.*

Lemma 4 (Advanced Composition, [19, 18]). *Let $\varepsilon, \delta \in (0, 1]$ and $k \in \mathbb{N}$. If we run k mechanisms where each mechanism is $\left(\frac{\varepsilon}{2\sqrt{2k \log (2/\delta)}}, \frac{\delta}{2k}\right)$-DP, then the entire algorithm is (ε, δ)-DP.*

3 DP Counting Queries on Approximate Shortest Paths

In this section, we show how we are able to release all-pairs counting queries much more accurately by allowing them to be on t-approximate shortest paths. We achieve a much lower error than the DP range query problem on true shortest paths by leveraging a result which gives an ε-DP recursive algorithm for releasing the counting queries over all-pairs shortest paths in trees with $O(\text{polylog}(n))$ additive error. Previous similar results were on DP releasing all-pairs shortest distances in trees by Sealfon [29], which were adapted to the (ε, δ)-DP and range query setting by Deng et al. [10]. In this paper, we explicitly provide the proof for releasing DP counting queries on all pairs of vertices in a tree under the ε-DP setting. We then apply the result to the studied problem of DP counting queries over approximate shortest paths carefully in order to save a $\log n$ term. Throughout this section, for any two vertices $x, y \in V$, we let $P_G(x, y)$ to be the shortest path between x and y and $\omega_G(x, y) = \sum_{e \in P_G(x,y)} f(e)$ to be the accumulation of edge attributes along the shortest path between x and y, or the *counting query*.

We will first prove the result for releasing counting queries from the root of a rooted tree to all other vertices, and then extend this result to counting queries for all pairs of vertices in a tree.

Lemma 5. *Let $\mathcal{T} = (V, E, \mathbf{w})$ be a tree rooted at z with $\varepsilon \in (0, 1]$ and $\gamma \in (0, 0.5]$. Then there is an ε-DP algorithm for releasing counting queries from the root to all other vertices on $\mathcal{T}$ that is $O(\log^{1.5}(n) \cdot \log (n/\gamma)/\varepsilon)$-accurate with probability $1 - \gamma$.*

Proof. The algorithm works by splitting the tree into subtrees, each with fewer than $n/2$ vertices, computing counting queries between the root and the roots of the subtrees, and then applying the algorithm recursively on each subtree.

It is a well-known result that every tree has either one or two vertices known as *centroids* such that their removal disconnects the tree into subtrees, each with fewer than $n/2$ vertices (see, for example, [31]). If $\mathcal{T}$ has a single centroid, we let z^* be that vertex. If $\mathcal{T}$ has two centroids, we let z^* be the centroid that is closer to the root node z. By choosing it in this way, we guarantee that z^* is the unique vertex such that the subtree rooted at z^* has more than $n/2$ vertices, but the subtree rooted at each of z^*'s children has at most $n/2$ vertices. Let $z_1, ..., z_t$ be the children of z^* and $\mathcal{T}_i = (V_i, E_i)$, $i \in \{1, ..., t\}$, their corresponding subtrees. Additionally, we define $\mathcal{T}_0$ to be the subtree rooted at z with vertex set $V_0 = V - \{V_1, ..., V_t\}$.

Next, we release the counting queries between z and z^*, as well as between z^* and each of it's children by adding Laplace noise from the distribution $\text{Lap}(\log(n)/\varepsilon)$. In particular, we note that since each tree is disjoint by construction, the function which releases these counting queries has sensitivity 1. Hence, by **Lemma 1**, releasing each subtree is $(\varepsilon/\log(n))$-DP. Then, we recursively call this algorithm for each subtree until each subtree consists only of a single vertex.

Since each subtree has at most $n/2$ vertices, the maximum recursion depth is bounded by $\log(n)$. Hence, by basic composition (**Lemma 3**), since the released queries in our algorithm are a composition of $\log(n)$, $(\varepsilon/\log(n))$-DP mechanisms, the total algorithm is ε-DP.

It remains to be shown that the error in each released estimate is bounded above by $O(\text{polylog}(n))$. We note that for every $u \in V$, the estimate of $\omega(z, u)$ is the composition of released queries from root nodes to centroids, and from centroids to their children. By bounding the number of released estimates in the composition, we can bound the total error incurred from each noisy random variable.

We will prove by induction on the number of vertices that the number of released estimates used to calculate an estimate of $\omega(z, u)$ is bounded above by $2\log(n)$. Firstly, the base case of $n = 1$ is vacuously true. Suppose $n > 1$. In the first iteration of the algorithm, the vertex sets $V_0, V_1, ..., V_t$ are a partition of V. Hence, $u \in V_i$ for some $i \in [t]$. By the induction hypothesis, we can release the distance from z_i to u by the composition of at most $2\log(n/2) = 2\log(n) - 2$ noisy distances. Then, we have that $\omega(z, z_i) = \omega(z, z^*) + \omega(z^*, z_i)$, giving a total of $2\log(n)$ noisy estimates.

It follows that the error in each released estimate is the concatenation of $2\log(n)$ random variables distributed according to $\text{Lap}(\log(n)/\varepsilon)$. By **Lemma 2**, with probability at most γ, we have that the error exceeds $O(\log^{1.5}(n) \cdot \log(1/\gamma)/\varepsilon)$. Hence, by a union bound, with probability at least $1 - \gamma$, the error for the estimate from z to all vertices $u \in V$ is bounded above by $O(\log^{1.5}(n) \cdot \log(n/\gamma)/\varepsilon)$, as desired. □

Next, we will prove another result by extending **Lemma 5** to the all-pairs case. We intentionally keep the term $O(\log^{1.5}(n) \cdot \log(n/\gamma)/\varepsilon)$ instead of $O(\log^{2.5}(n) \cdot \log(1/\gamma)/\varepsilon)$, in order to facilitate a more careful analysis in **Theorem 1** to save a $\log n$ term through **Proposition 1**.

Proposition 1 states a few basic results about asymptotics.

Proposition 1. *Let $\alpha \geq 2$, $\gamma \in (0, 0.5]$, and $\lambda > 0$, then,*

1. $\log(\alpha^{1+\lambda}/\gamma) = O(\log(\alpha/\gamma))$, *and,*
2. $\log(\alpha/\gamma) = O(\log(\alpha)\log(1/\gamma))$.

Lemma 6. *Let $\mathcal{T} = (V, E, \mathbf{w})$ be a tree with $\varepsilon \in (0, 1]$ and $\gamma \in (0, 0.5]$. Then there is an ε-DP algorithm for releasing counting queries on all pairs of vertices in $\mathcal{T}$ that is $O(\log^{1.5}(n) \cdot \log(n/\gamma)/\varepsilon)$-accurate with probability $1 - \gamma$.*

Proof. Choose a vertex $z \in V$ arbitrarily as a root vertex of $\mathcal{T}$, and call the rooted tree $\mathcal{T}_z$. Then, using **Lemma 5**, we can privately release the counting query from z to every other vertex in $\mathcal{T}_z$. We will show that these estimate suffice to compute the estimate for any pair $x, y \in V$ of vertices in V.

Let $v_{x,y} \in V$ be the lowest common ancestor of x and y in the rooted tree $\mathcal{T}_z$. Then, $\omega_{\mathcal{T}}(x, y) = \omega_{\mathcal{T}}(x, v_{x,y}) + \omega_{\mathcal{T}}(v_{x,y}, y) = \omega_{\mathcal{T}_z}(z, x) + \omega_{\mathcal{T}_z}(z, y) - 2\omega_{\mathcal{T}_z}(z, v_{x,y})$. By the result in **Lemma 5**, we condition on the event that for all $w \in V$, $\omega(z, w)$ is $O(\log^{1.5}(n) \cdot \log(n/\gamma)/\varepsilon)$-accurate with probability $1 - \gamma$. Since our query estimate is the sum of four estimates from z, the total error is at most four times this, which is still $O(\log^{1.5}(n) \cdot \log(n/\gamma)/\varepsilon)$. By a union bound, with probability $1 - \gamma$ each counting query among all n^2 pairs counting queries is at most $O(\log^{1.5}(n) \cdot \log(n^3/\gamma)/\varepsilon) = O(\log^{1.5}(n) \cdot \log(n/\gamma)/\varepsilon)$, by **Proposition 1**. □

Applying **Lemma 6** directly to a collective tree spanner of a graph gives the desired result.

*Proof (of **Theorem 1**).* Since $\mathbf{T}$ is constructed only based on the public edge weights and topology of G, each tree $\mathcal{T} \in \mathbf{T}$ is also public. For each tree $\mathcal{T}$ we run the $\varepsilon/|\mathbf{T}|$-DP algorithm from **Lemma 6** to release the counting query over $\mathcal{T}$. By basic composition (**Lemma 3**), releasing the counting queries in all the trees is ε-DP.

By the definition of a collective tree spanner, for every pair of vertices $u, v \in V$, there is a tree $\mathcal{T} \in \mathbf{T}$ such that $P_{\mathcal{T}}(u, v)$ is a t-approximate shortest path between u and v. We let $\widetilde{\omega}(u, v)$ to be the private counting query estimate over the path $P_{\mathcal{T}}(u, v)$. The result of **Lemma 6** ensures that, with probability $1 - \gamma/n^2$, the additive error in the released counting query is,

$$\begin{aligned} \max_{u,v \in V} |\omega_{\mathcal{T}}(u, v) - \widetilde{\omega}(u, v)| &\leq O(|\mathbf{T}| \cdot \log^{1.5}(n) \cdot \log(n^3/\gamma)/\varepsilon) \\ &= O(\eta_t \cdot \log^{1.5}(n) \cdot \log(n/\gamma)/\varepsilon) \\ &= O(\eta_t \cdot \log^{2.5}(n) \cdot \log(1/\gamma)/\varepsilon) \end{aligned} \tag{1}$$

where the two equalities hold because of **Proposition 1**. By a union bound, the upper bound (1) holds for all n^2 pairs of vertices with probability at least $1 - \gamma$. □

For DPCQ on ASP, the (ε, δ) case is identical to the ε-DP case except for two differences. Firstly, we use *advanced composition* to release the private counting query estimates for each tree $\mathcal{T} \in \mathbf{T}$ instead of basic composition (**Lemma 3**).

The second difference is that, instead of applying **Lemma 6** to compute the distance estimates in each tree $\mathcal{T}$, we apply the following lemma by Deng et al. which adds Gaussian noise to the edge weights instead of Laplacian noise to reduce the final error by a factor of $\log(n)$ in the (ε, δ)-DP case. We state the result here but refer the reader to [10] for a detailed proof.

Lemma 7. *Let $\mathcal{T} = (V, E, \mathbf{w})$ be a tree with $\varepsilon, \delta \in (0, 1]$ and $\gamma \in (0, 0.5]$. Then there is an (ε, δ)-DP algorithm for releasing counting queries on all pairs of vertices in $\mathcal{T}$ that is $O(\log(n) \cdot \sqrt{\log(1/\delta)} \cdot \sqrt{\log(n/\gamma)}/\varepsilon)$-accurate with probability $1 - \gamma$.*

The proof of **Theorem 2** is similar to the proof of **Theorem 1** and is therefore deferred to **Appendix A**.

To give an explicit instance of **Theorem 1** and **2**, we note that Abraham et al. [1] gave a construction based on Ramsey spanning trees for a collective tree $O(k \log \log n)$-spanner with $kn^{1/k}$ trees. By the result of **Theorem 2**, choosing $k \geq 3$ immediately gives improvement on the additive approximate over the $\widetilde{O}(n^{1/4})$ algorithm on true shortest paths from [10].

Runtime Analysis. Not only does releasing differentially private counting queries over approximate shortest paths allow us to greatly improve the additive error in the released estimates, but also potentially the runtime of the algorithm to release the estimates. In particular, the $\widetilde{\Theta}(n^{1/4})$ algorithm from [10] to release counting queries over exact shortest paths requires first sampling a *hitting set* of vertices of size $\widetilde{O}(\sqrt{n})$, and then computing the shortest path trees spanning from each vertex in the hitting set. Using Dijkstra's algorithm, this time complexity of this process is $\widetilde{O}(n^{2.5})$.

By allowing the counting queries to be along approximate shortest paths, we bypass the need to calculate exact shortest paths from a hitting set of vertices. In specific, since the Laplace noise can be sampled and added to the edges of a tree in $O(1)$ time and the centroid can be found in linear time, the recursive process described in the proof of **Lemma 5** takes $O(n \log n)$ time. It follows that the total algorithm to release the counting queries on approximate paths can run in $O(S(n) + \eta_t \cdot n \log n)$ time, where $S(n)$ is the amount of time needed to construct the collective tree spanner on the graph G.

4 Conclusion and Future Work

In this work, we introduce the problem of differentially private counting queries on approximate shortest paths in graphs. In specific, we show that the additive error lower bound of $\widetilde{\Omega}(n^{1/4})$ for counting queries on true shortest paths can be greatly improved to $\widetilde{O}(\sqrt{kn^{1/k}}/\epsilon)$ at the cost of querying along $\widetilde{O}(k)$-approximate shortest paths. It is still an open question what the optimal trade-off between the quality of shortest paths and the additive error is. We discuss the future work from the perspective of collective tree spanners.

Collective Tree Spanners

The method we provide in this work is based on the existence of *collective tree spanners* for general graphs. Tree graphs have many convenient properties such as being well-suited to recursive algorithms and having unique path systems, emphasizing their convenience in algorithm design. Despite this, until recently there have been few results about multiplicative collective tree spanners for general graphs. Even the collective tree $O(k \log \log n)$-spanner mentioned in this work is an instance of a *tree padding spanner*, making it optimal for routing problems, but sacrificing an optimal tradeoff between stretch and size for a property that is not necessary for applications like ours.

In specific, we note that one clear gap in the current literature is the construction of a constant stretch collective tree spanner with optimal size and stretch tradeoff, namely a collective tree $(2k-1)$-spanner with $O(n^{1/k})$ trees. In what follows, we give an outline of the construction of such a spanner.

In particular, consider the celebrated distance oracle of Thorup and Zwick [30]. Their distance oracle samples nested subsets of vertices of decreasing size $V = A_0 \supseteq A_1 \supseteq ... \supseteq A_{k-1} \supseteq A_k = \varnothing$. For each vertex $w \in A_i$, they define a set of vertices called a *cluster* by $C(w) \coloneqq \{v \in V \mid d(w,v) < d(v, A_{i+1})\}$. Then, they compute the shortest path tree rooted at w spanning each cluster. The shortest path distances within each cluster are then used to output the distance approximations, inducing a *tree cover* of the graph. In expectation, the number of edges in the union of spanning trees of clusters centered at vertices in A_i is $O(n^{1+1/k})$, meaning that the clusters at each level could plausibly be *packed* into $O(n^{1/k})$ spanning trees of G, giving a final collective tree $(2k-1)$-spanner with $O(kn^{1/k})$ trees.

One issue with this approach is that the clusters at each level may have some intersections with one another, and hence we cannot guarantee that packing the clusters together will produce a spanning tree of G. Instead, we can equally partition each set A_i randomly into $\beta \in \mathbb{N}$ subsets A_i^r, $r \in [\beta]$ and then define the clusters $C^*(w) = \{v \in V \mid d(w,v) < d(v, A_i^r - \{w\})\}$. By this definition, clearly the clusters at each level are disjoint. As long as β is chosen to be large enough such that, with high probability $d(v, A_{i+1}) < d(v, A_i^r - \{w\})$ for every $v \in V$, $i \in [k-1]$, and $r \in [\beta]$, we have that $C(w) \subseteq C^*(w)$, implying that the $(2k-1)$ stretch result still holds for the modified definitions of clusters. Furthermore, since for each partition of A_i we construct a single spanning tree of G, the final collective tree spanner would consist of $k \cdot \beta$ trees. Future work can formalize how large β must be in-terms of n to give an explicit construction of a collective tree $(2k-1)$-spanner using this method.

5 Acknowledgements

This material is based upon work supported by NSF Grant CNS-2349369. This work was completed during the UNCG GraLNA 2024 REU, where Jesse Campbell was working under the mentorship of Chunjiang Zhu.

A Proof of Theorem 2

Theorem 2. *Let* $\mathbf{T}$ *be a* t*-collective tree spanner of* G *such that* $|\mathbf{T}| = \eta_t$*. For* $\gamma \in (0, 0.5]$ *and* $\varepsilon, \delta \in (0, 1]$*, there is an* (ε, δ)*-DP algorithm for releasing the counting query between* $u, v \in V$ *on a* t*-approximate shortest path that is* $O(\sqrt{\eta_t} \cdot \log^2(n) \cdot \log(1/\delta) \cdot \sqrt{\log(1/\gamma)}/\varepsilon)$*-accurate with probability* $1 - \gamma$*.*

Proof. For each tree $\mathcal{T} \in \mathbf{T}$ we run the $(\varepsilon/(2\sqrt{2|\mathbf{T}|\log(2/\delta)}), \delta/(2|\mathbf{T}|))$-DP algorithm from **Lemma 7** to release the counting query over $\mathcal{T}$. By advanced composition (**Lemma 4**), releasing the counting queries in all the trees is (ε, δ)-DP.

By the definition of collective tree spanners, for every pair of vertices $u, v \in V$, there is a tree $\mathcal{T} \in \mathbf{T}$ such that $P_{\mathcal{T}}(u, v)$ is a t-approximate shortest path between u and v. We let $\widetilde{\omega}(u, v)$ to be the private counting query estimate over the path $P_{\mathcal{T}}(u, v)$. The result of **Lemma 7** ensures that, with probability $1 - \gamma/n^2$, the additive error in the released counting query is,

$$\begin{aligned}
&\max_{u,v \in V} |\omega_{\mathcal{T}}(u, v) - \widetilde{\omega}(u, v)| \\
&\quad \leq O(\sqrt{|\mathbf{T}|} \cdot \sqrt{\log(2/\delta)} \cdot \log(n) \cdot \sqrt{\log(1/\delta)} \cdot \sqrt{\log(n^3/\gamma)}/\varepsilon) \\
&\quad = O(\sqrt{\eta_t} \cdot \log^2(n) \cdot \log(1/\delta) \cdot \sqrt{\log(1/\gamma)}/\varepsilon) \quad (2)
\end{aligned}$$

by **Proposition 1**. By a union bound, the upper bound (2) holds for all n^2 pairs of vertices with probability at least $1 - \gamma$. □

B Proof of Proposition 1

Proposition 1. *Let* $\alpha \geq 2$*,* $\gamma \in (0, 0.5]$*, and* $\lambda > 0$*, then,*

1. $\log(\alpha^{1+\lambda}/\gamma) = O(\log(\alpha/\gamma))$*, and,*
2. $\log(\alpha/\gamma) = O(\log(\alpha)\log(1/\gamma))$*.*

Proof (of 1).

$$\begin{aligned}
\log(\alpha^{1+\lambda}/\gamma) = \log((\alpha/\gamma^{1/(1+\lambda)})^{1+\lambda}) &= (1+\lambda) \cdot \log(\alpha/\gamma^{1/(1+\lambda)}) \\
&\leq (1+\lambda) \cdot \log(\alpha/\gamma) = O(\log(\alpha/\gamma))
\end{aligned}$$

Proof (of 2). Suppose the base of our logarithm is $b = \{2, e\}$. Then,

$$\log(\alpha/\gamma) = \log(\alpha) + \log(1/\gamma) = O(\max\{\log(\alpha), \log(1/\gamma)\}) = O(\log(\alpha)\log(1/\gamma))$$

as $\alpha, 1/\gamma \geq 2$. □

References

[1] Ittai Abraham et al. "Ramsey Spanning Trees and Their Applications". In: *ACM Trans. Algorithms* 16.2 (2020). ISSN: 1549-6325.

[2] Reyan Ahmed et al. "Graph spanners: A tutorial review". In: *Computer Science Review* 37 (2020).

[3] Ahmet Aktay et al. "Google COVID-19 Community Mobility Reports: Anonymization Process Description (version 1.1)". In: *ArXiv preprint* arXiv: 2004.04145 (2020).

[4] Yair Bartal, Ora Nova Fandina, and Ofer Neiman. "Covering metric spaces by few trees". In: *Journal of Computer and System Sciences* 130 (2022), pp. 26–42.

[5] Shailesh Bavadekar et al. "Google COVID-19 Search Trends Symptoms Dataset: Anonymization Process Description (version 1.0)". In: *ArXiv preprint* arXiv: 2009.01265 (2020).

[6] Greg Bodwin et al. "The Discrepancy of Shortest Paths". In: *ArXiv preprint* arXiv:2401.15781 (2024).

[7] Hubert Chan, Elaine Shi, and Dawn Song. "Private and Continual Release of Statistics". In: *ACM Trans. Inf. Syst. Secur.* 14.3 (2011). ISSN: 1094-9224.

[8] B. Chazelle and A. Lvov. "A Trace Bound for the Hereditary Discrepancy". In: *Discrete & Computational Geometry* 26.2 (2001), pp. 221–231.

[9] Justin Y. Chen et al. "Differentially Private All-Pairs Shortest Path Distances: Improved Algorithms and Lower Bounds". In: *Proceedings of the 2023 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)* (2023), pp. 5040–5067.

[10] Chengyuan Deng et al. "Differentially Private Range Query on Shortest Paths". In: *Algorithms and Data Structures - 18th International Symposium, WADS 2023, Proceedings*. Springer Science and Business Media Deutschland GmbH, 2023, pp. 340–370.

[11] Bolin Ding, Janardhan Kulkarni, and Sergey Yekhanin. "Collecting Telemetry Data Privately". In: *Advances in Neural Information Processing Systems*. Ed. by I. Guyon et al. Vol. 30. Curran Associates, Inc., 2017.

[12] Feodor F. Dragan and Muad Abu-Ata. "Collective additive tree spanners of bounded tree-breadth graphs with generalizations and consequences". In: *Theoretical Computer Science* 547 (2014), pp. 1–17.

[13] Feodor F. Dragan and Chenyu Yan. "Collective Tree Spanners in Graphs with Bounded Parameters". In: *Algorithmica* 57.1 (2010), pp. 22–43.

[14] Feodor F. Dragan, Chenyu Yan, and Derek G. Corneil. "Collective Tree Spanners and Routing in AT-free Related Graphs". In: *Graph-Theoretic Concepts in Computer Science*. Springer Berlin Heidelberg, 2005, pp. 68–80.

[15] Feodor F. Dragan, Chenyu Yan, and Irina Lomonosov. "Collective Tree Spanners of Graphs". In: *Algorithm Theory - SWAT 2004*. Springer Berlin Heidelberg, 2004, pp. 64–76.

[16] Feodor F. Dragan et al. "Collective additive tree spanners for circle graphs and polygonal graphs". In: *Discrete Applied Mathematics* 160.12 (2012), pp. 1717–1729.

[17] Cynthia Dwork and Jing Lei. "Differential privacy and robust statistics". In: *Proceedings of the Forty-First Annual ACM Symposium on Theory of Computing*. STOC '09. Bethesda, MD, USA: Association for Computing Machinery, 2009, pp. 371–380. ISBN: 9781605585062.

[18] Cynthia Dwork and Aaron Roth. "The Algorithmic Foundations of Differential Privacy". In: *Foundations and Trends in Theoretical Computer Science* 9.3–4 (2014), pp. 211–407. ISSN: 1551-305X.

[19] Cynthia Dwork et al. "Calibrating Noise to Sensitivity in Private Data Analysis". In: *Theory of Cryptography*. Ed. by Shai Halevi and Tal Rabin. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 265–284. ISBN: 978-3-540-32732-5.

[20] Chenglin Fan, Ping Li, and Xiaoyun Li. "Breaking the linear error barrier in differentially private graph distance release". In: NeurIPS. 2022.

[21] Chenglin Fan, Ping Li, and Xiaoyun Li. "Private graph all-pairwise-shortest-path distance release with improved error rate". In: *Proceedings of the 36th International Conference on Neural Information Processing Systems*. NIPS '22. Curran Associates Inc., 2024. ISBN: 9781713871088.

[22] Andrew David Foote, Ashwin Machanavajjhala, and Kevin McKinney. "Releasing Earnings Distributions using Differential Privacy: Disclosure Avoidance System For Post-Secondary Employment Outcomes (PSEO)". In: *Journal of Privacy and Confidentiality* 9.2 (2019).

[23] Simson L. Garfinkel, John M. Abowd, and Sarah Powazek. "Issues Encountered Deploying Differential Privacy". In: *Proceedings of the 2018 Workshop on Privacy in the Electronic Society*. WPES'18. Toronto, Canada: Association for Computing Machinery, 2018, pp. 133–137. ISBN: 9781450359894.

[24] A. Gupta, A. Kumar, and R. Rastogi. "Traveling with a Pez dispenser (or, routing issues in MPLS)". In: *Proceedings 42nd IEEE Symposium on Foundations of Computer Science*. 2001, pp. 148–157.

[25] Michael Hay et al. "Accurate Estimation of the Degree Distribution of Private Networks". In: *2009 Ninth IEEE International Conference on Data Mining*. 2009, pp. 169–178.

[26] Xun Jian, Yue Wang, and Lei Chen. "Publishing Graphs Under Node Differential Privacy". In: *IEEE Transactions on Knowledge and Data Engineering* 35.4 (2023), pp. 4164–4177.

[27] Ganghong Liu, Xuebin Ma, and Wuyungerile Li. "Publishing Node Strength Distribution With Node Differential Privacy". In: *IEEE Access* 8 (2020), pp. 217642–217650.

[28] Wenfen Liu et al. "Graph Node Strength Histogram Publication Method with Node Differential Privacy". In: *Journal of Physics: Conference Series* 1757.1 (2021), p. 012186.

[29] Adam Sealfon. "Shortest Paths and Distances with Differential Privacy". In: PODS '16. New York, NY, USA: Association for Computing Machinery, 2016, pp. 29–41. ISBN: 9781450341912.

[30] M. Thorup and U. Zwick. "Approximate distance oracles". In: *Journal of the ACM* 52.1 (2005), pp. 1–24.

[31] Hua Wang. "Centroid, Leaf-centroid, and Internal-centroid". In: *Graphs and Combinatorics* 31.3 (2015), pp. 783–793.