Private Graph Distance Computation with Improved Error Rate

Anonymous Author(s)
Affiliation
Address
email

Abstract

Releasing all pairwise shortest path (APSP) distances between vertices on general graphs under weight Differential Privacy (DP) is known as a challenging task. Previous work achieved DP with the maximal absolute error among all published pairwise distances bounded by $\tilde{O}(n)$ where n is the number of nodes. Whether the approximation error can be reduced to sublinear in n is posted as an interesting open problem. In this paper, we break the linear barrier on the distance approximation error in APSP release, by proposing an algorithm that releases a constructed synthetic graph privately. Computing all pairwise distances on the constructed graph only introduces $\tilde{O}(n^{1/2})$ error in answering all pairwise shortest path distances for fixed privacy parameter. Our method is based on a novel graph diameter (link length) augmentation via constructing "shortcuts" for the paths and the use of Laplace noise with non-zero mean. Numerical examples are also provided. Additionally, we also propose a DP algorithm with error rate $\tilde{O}(k)$, which improves the error of general graphs, when the graph has small feedback vertex set number $k = o(n^{1/2})$.

1 Introduction

In recent years, there has been a growing interest in private data analysis, from academic research to industry practice. Typically, many industrial machine learning applications consist of two steps: 1) collecting data from users; 2) training models using the collected user data. The key question is: throughout this process, how can we protect the privacy of each individual, without leaking sensitive data of each user? In other words, how can we prevent an adversarial attacker from inferring any user's data, given the public information that can be accessed? This has been the overarching question in a line of research that studies private algorithms under various settings. The concept of differential privacy (DP) [Dwork et al., 2006, Blum et al., 2005, Chawla et al., 2005, Dwork, 2006] has been a popular approach for rigorously defining and resolving the problem of keeping useful information for model learning, while protecting privacy for each individual. In the traditional setting, databases D and D' are collections of data records and are considered to be neighboring if they are identical except for a single record (individual information). DP requires that the output of running a randomized algorithm on D and D' should have very close probability distributions. There are many types of databases, and in this work, we will specifically focus on the differential privacy of graphs.

Weight private graphs. In general, there are three types of private graph models, regarding the nodes, the edges and the edge weights, respectively. The node private model requires DP for two adjacent graphs differing in one node. In the edge private model, two neighboring graphs are defined such that they only differ by one edge, while sharing the same set of nodes.

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

In this paper, we focus on the weight private graph model, which was first proposed by Sealfon [2016]. In the weight private graph problem, the topology of the graph is public, which means that the adjacent graphs (databases) when considering DP have same nodes and edges, while this is not the case in other two DP models. Specifically, in the weight private model we consider two graphs being adjacent if the difference between the sums of their edge weights is no more than one unit. One example application of this problem is where one tries to release the transportation volume between several places of interest with privacy constraints, where the roads are regarded as edges. In some sense, it might be hard to keep the graph structure private in practice, since one may easily obtain the map by modern tools like Google Earth. However, in some cases, information like traffic flows could also be sensitive and needs to be kept private. Releasing such private information is well suited for the setting of weight private graph model.

1.1 Open Problem, and Previous Results

In this paper, we study the problem of releasing all pairwise shortest-path (APSP) distances with weight privacy. Two standard strategies in DP are: 1) adding Laplace noise to each edge, 2) adding Laplace noise to the output distance matrix. As discussed in Sealfon [2016], the error of these two strategies is roughly $O(\epsilon^{-1} n \log n)$ by adding noise to each edge and $O(\epsilon^{-1} n \sqrt{\log(1/\delta)})$ using DP composition theorems, respectively, where n is the number of nodes in the graph and (ϵ, δ) is the DP parameter. Here, the approximation error is measured by the largest absolute difference among all released pairwise distances and the ground truth. As we see, both of these errors are roughly $O(n)$, i.e., linear in n when ϵ and δ are fixed. Whether we can achieve differential privacy with error sublinear in n is still an open problem, as recognized by an online repository (DifferentialPrivacy.org[1]) focusing on the research of DP, quoted *"... Is this linear dependence on n inherent, or is it possible to release all-pairs distances with error sublinear in n ?"*

While this linear barrier seems to hold for general graphs, we can derive improved results on several special types of graph. Firstly, for a graph with weights bounded in $[0, M]$, Sealfon [2016] picked a subset $Z \in \mathbb{V}$ of vertices as the "k-covering set" to approximate the original graph. Each vertex u can map to its closest vertex z_u in the covering set. For any node pair u, v, their distance is approximated by the distance between z_u and z_v plus $O(kM)$ additional error. Then, the author proposed to use $O(|Z|^2)$ pairs of distances to approximate all $O(n^2)$ pairwise distances, which leads to $O(\sqrt{nM})$ approximation error for fixed privacy parameters eventually. Secondly, for grid graph with arbitrary positive weights, Fan and Li [2022] proposed to select an intermediate vertex set and divide the shortest path between any node pair into at most three parts depending on its traverse of the set. By connecting the nodes in the intermediate set and applying the standard Laplace mechanism, the authors constructed a DP algorithm with $\tilde{O}(n^{3/4})$ approximation error. Thirdly, for trees, Sealfon [2016] gave a recursive algorithm to release all-pairs distances with error $O(\log^{2.5} n)$, which was improved to $O(\log^{1.5} n \log^{1.5} h)$ by Fan and Li [2022], where h is the depth of tree and can be as small as $O(\log n)$. The general idea was to build a collection $\mathbb{T}$ of subtrees of $\log n$ levels, where the subtrees in each level are disjoint to each other such that each edge appears in at most $O(\log n)$ trees, allowing us to only add $O(\log n)$ units of Laplace noise to each path in $\mathbb{T}$. This leads to $poly \log n/\epsilon$ approximation error on trees.

1.2 Our Results and Techniques

In this paper, we propose an algorithm that is able to surpass the linear $O(n)$ error of differentially privately releasing all pairwise distances on general weighted graphs. We make use of the improved and extended techniques from Fan and Li [2022] to general graphs to obtain $\tilde{O}(n^{1/2})$ approximation error. Moreover, when the graph exhibits some special property having small feedback vertex set number, we develop a new algorithm with further improved error bound.

Breaking the linear error barrier for general graphs. Let $G(\mathbb{V}, \mathbb{E}, w)$ denote an undirected graph, where $\mathbb{V}$ is the set of nodes with $|\mathbb{V}| = n$, $\mathbb{E}$ is the set of all edges, and w is the set of corresponding weights. We use $w(e)$ to denote the weight of a specific edge $e \in \mathbb{E}$. The key challenge in this problem is that, we need to bound the error of all shortest paths instead of just one. Since the shortest paths could overlap with each other, the dependency among the shared edges (and the noises) hinders

[1] https://differentialprivacy.org/open-problem-all-pairs/

us from applying standard concentration results. To achieve sublinear error in DP all pairwise distance release problem, our idea, intuitively, is to find "shortcuts" that will reduce the number of edges in long shortest paths (consisting of many edges). The algorithm proceeds as follows. Denote $d(u, v)$ as the true shortest path distance between node u and v w.r.t. graph $G(\mathbb{V}, \mathbb{E}, w)$. First, we randomly sample $n^{1/2}$ vertices from $\mathbb{V}$ to form a subset of vertices $\mathbb{V}_1 \subset \mathbb{V}$. We then create an edge set $\mathbb{E}_1 = \{(u, v) : u \in \mathbb{V}_1, v \in \mathbb{V}_1, u \neq v\}$ and assign each edge the weight $w_1(u, v) = d(u, v)$. Denote $\mathbb{E}_0 =: \{(u, v) \in \mathbb{E} : u \notin \mathbb{V}_1 \text{ or } v \notin \mathbb{V}_1\}$ as the edge set between nodes that are not both in $\mathbb{V}_1$. Next, we add noise to the edges in $\mathbb{E}_0$ and $\mathbb{E}_1$, respectively. For $e \in \mathbb{E}_0$, we add Laplace noise following $Lap(\mu_0, \sigma_0)$ where $\sigma_0 = O(1/\epsilon)$, $\mu_0 = \sigma \log(n/\gamma)$; for $e \in \mathbb{E}_1$, the noise is from $Lap(\mu_1, \sigma_1)$ with $\sigma_1 := O(n^{1/2}\sqrt{\log(1/\delta)})/\epsilon)$ and $\mu_1 = \sigma_1 \log(n/\gamma)$. Note that, the mean of the Laplace noise is non-zero in our approach. We will show that this novel design guarantees that the APSP distances computed directly from the synthetic and noisy graph G' are lower bounded by the true distances for all node pairs (w.h.p.), and the approximation error is at most $\tilde{O}(n^{1/2})$ (Theorem 3.6).

Graph with small feedback vertex set number. A feedback vertex set (FVS) of a graph, also called a loop cutset [Freuder, 1982], is a set of vertices whose removal leaves a graph without cycles, and the feedback vertex set number of a graph is the size of a smallest feedback vertex set. The feedback vertex set problem is NP-hard [Lewis, 1978] and the best known approximation algorithm on undirected graphs is by a factor of two [Becker and Geiger, 1996]. Beyond the results derived for general weighted graphs, we also consider the problem for graphs with small feedback vertex set. We design a new algorithm that releases the distances privately with $\tilde{O}(k)$ error where k is the feedback vertex set number, improving the error bound when $k = o(n^{1/2})$ compared with the result for general weighted graphs.

1.3 More Related Work

The topic of private graphs has attracted substantial research interests in the recent years [Hay et al., 2009, Rastogi et al., 2009, Gupta et al., 2010, Karwa et al., 2011, Gupta et al., 2012, Blocki et al., 2013, Kasiviswanathan et al., 2013, Bun et al., 2015, Sealfon, 2016, Ullman and Sealfon, 2019, Borgs et al., 2018, Arora and Upadhyay, 2019, Fan and Li, 2022]. Our work focuses on weight private graphs and generalizes the previous work [Fan and Li, 2022] which focused on trees and grid graphs.

2 Background: Differential Privacy (DP) on Graph

Consider a connected graph $G = (\mathbb{V}, \mathbb{E}, w)$, with w the collection of the weights of $\mathbb{E}$. We use $w(e)$ to denote the weight of an edge $e \in \mathbb{E}$. Denote $n = |\mathbb{V}|$, $m = |\mathbb{E}|$, with $n \leq m + 1$.

We first define the notion of neighboring graphs. In weight private model, since the nodes and edges of the graph G are unchanged, we can simply use edge weights to represent the graph.

Definition 2.1 (Neighboring)**.** *Graph $G = (\mathbb{V}, \mathbb{E}, w)$ and $G' = (\mathbb{V}, \mathbb{E}, w')$ are called neighboring, noted as $G \sim G'$, if*

$$||w - w'||_1 := \sum_{e \in \mathbb{E}} |w(e) - w'(e)| \leq 1.$$

The Differential Privacy (DP) introduced by Dwork [2006] is defined below adapted to our problem.

Definition 2.2 (Differential Privacy [Dwork, 2006])**.** *If for any two neighboring graphs $G = (\mathbb{V}, \mathbb{E}, w)$ and $G' = (\mathbb{V}, \mathbb{E}, w')$, a randomized algorithm $\mathbb{A}$, and a set of outcomes $O \subset Range(\mathbb{A})$,*

$$Pr[\mathbb{A}(G) \in O] \leq e^{\epsilon} Pr[\mathbb{A}(G') \in O] + \delta,$$

we say algorithm $\mathbb{A}$ is (ϵ, δ)-differentially private.

If $\delta = 0$, we say that the algorithm is ϵ-DP. The parameter δ is usually interpreted as the probability allowed for bad cases where ϵ-DP is violated. Intuitively, differential privacy requires that after changing the database by a little (the total weights in our case), the output should not be too different from that of the original database. Differential privacy may be achieved through the introduction of noise to the output. To attain general (ϵ, δ)-DP, we may add Gaussian noise to the output. To achieve ϵ-DP, the noise added typically comes from the Laplace distribution. Smaller ϵ and δ indicate stronger

privacy, which, however, usually sacrifices utility. Thus, one of the central topics in the differential privacy literature is to reduce the scale of noise added, while satisfying the privacy constraint.

In this work, we focus on the popular approach to achieve DP by adding Laplace noises. The Laplace distribution with parameter b has density function $f(x) = \frac{1}{2b}\exp(-|x|/b)$.

Lemma 2.1 (Dwork [2006]). *For a function $f : \mathcal{G} \to \mathbb{R}$ with $\mathcal{G}$ the input space of graphs, define the sensitivity*

$$\triangle_f = \max_{G \sim G'} ||f(G) - f(G')||_1,$$

where G, G' are two neighboring graphs. Let X be a random noise drawn from $Lap(0, \triangle_f/\epsilon)$. The Laplace mechanism outputs

$$M_{f,\epsilon}(G) = f(G) + X,$$

which achieves ϵ-differential privacy.

One important and attractive property of DP is that, different DP algorithms can be easily combined together, also with strict DP guarantee.

Lemma 2.2 (Advanced Composition Theorem [Dwork et al., 2010]). *For any $\epsilon, \delta, \delta' \geq 0$, the adaptive composition of k times (ϵ, δ)-differentially private mechanisms is $(\epsilon', k\delta + \delta')$-differentially private for*

$$\epsilon' = \sqrt{2k\log(1/\delta')} \cdot \epsilon + k \cdot \epsilon(e^{\epsilon} - 1),$$

which is $O(\sqrt{k\log(1/\delta')} \cdot \epsilon)$ when $k \leq 1/\epsilon^2$. In particular, if $\epsilon' \in (0, 1), \delta, \delta' > 0$, the composition of k times $(\epsilon, 0)$-differentially private mechanism is (ϵ', δ')-differentially private for

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

Finally, we introduce a basic tool regarding the Laplace random variables which will be used in the error analysis.

Lemma 2.3. *Consider n i.i.d. random variables $Z_1, Z_2, ..., Z_n$ from $Lap(b)$. With probability at least $1 - \gamma$, $\forall 0 < \gamma < 1$, all n Laplace random variables have magnitude bounded by $b\log(n/\gamma)$.*

3 Private Synthetic Graph Release for APSP Distance

We formally define the problem of interest and some basic concepts as below.

Definition 3.1. *(Approximate Distances Release) Given an graph $G(\mathbb{V}, \mathbb{E}, w)$, our task is to release all pairwise shortest path distances privately. Let $\hat{d}(\cdot, \cdot)$ be the output (approximate) distance function. Our object is to minimize the maximal absolute error over all pairs, namely, $\max\{|\hat{d}(u, v) - d(u, v)| : u, v \in \mathbb{V}\}$, with $\hat{d}(u, v)$, $\forall u, v \in \mathbb{V}$ being DP to weight w. We call $|\hat{d}(u, v) - d(u, v)|$ the additive error of node pair (u, v).*

Definition 3.2 (Path). *A path $\tilde{P}_{v_1,v_L}$ between $v_1, v_L \in \mathbb{V}$ is defined as $\tilde{P}_{v_1,v_L} = \{(v_i, v_{i+1}), 1 \leq i \leq L - 1\}$, the collection of edges between connected node sequence $v_1, v_2, ..., v_k$, with some $L \leq n$. A segment of path $\tilde{P}$ is defined as a **consecutive** sub-path, $\{(v_i, v_i + 1), s \leq i \leq t - 1\}$ for some s, t.*

Definition 3.3 (Shortest path). *Let $S_{u,v}$ be the set of all paths between $u, v \in \mathbb{V}$. For a path $P \in S_{u,v}$ and weights w, denote $d(P, \mathcal{W}) = \sum_{e \in P} w(e)$. The shortest path of $u, v \in \mathbb{V}$ w.r.t. weights w is*

$$P_{u,v} = \operatorname*{argmin}_{P \in S_{u,v}} d(P, w),$$

and the shortest path distance is $d(P_{u,v}, w)$.

Definition 3.4 (Canonical shortest path). *For a given graph $G(\mathbb{V}, \mathbb{E}, w)$, let $\mathbb{V}_1 \subseteq \mathbb{V}$ be a subset of $\mathbb{V}$. Let $P_{x,y} = \{x, v_1, v_2, ..., v_L, y\}$ be the shortest path between $x, y \in \mathbb{V}$. Then a canonical shortest path $P_{x,y}^{\mathbb{V}_1}$ is defined by either of the following:*

1. *$P_{x,y}^{\mathbb{V}_1} \equiv P_{x,y}$, if $P_{x,y}$ contains at most one vertex in $\mathbb{V}_1$;*
2. *$P_{x,y}^{\mathbb{V}_1} = \{x, v_1, ..., p, q, ..., v_L, y\}$, if p, q are the closest nodes in $P_{x,y}$ to x, y respectively, and $p, q \in \mathbb{V}_1$.*

Input: General graph $G = (\mathbb{V}, \mathbb{E}, w)$, private parameter ϵ, δ, γ.
$\epsilon' = \epsilon/2$.
Sample $(n^{1/2})$ vertices from $\mathbb{V}$ uniformly and add them to $\mathbb{V}_1$.
Create an edge set $\mathbb{E}_1 := \{(u, v) : u \in \mathbb{V}_1, v \in \mathbb{V}_1, u \neq v\}$.
For each $e \in \mathbb{E}_1$, let $APSP(e)$ be the exact shortest path distance in G between (u, v) where $e = (u, v)$.
for *each edge* $e \in \mathbb{E}_1$ **do**
  Let $\sigma_1 := (2\sqrt{2}n^{1/2}\sqrt{\log(1/\delta)})/\epsilon'$ and $\mu_1 := \sigma_1 \log(n/\gamma)$.
  Compute $X_e := Lap(\mu_1, \sigma_1)$.
  $w'(e) := APSP(e) + X_e$.
$\mathbb{E}_0 := \mathbb{E} \setminus \mathbb{E}_1$
for *each edge* $e \in \mathbb{E}_0$ **do**
  Let $\sigma_0 := 1/\epsilon', \mu_0 := \sigma_0 \log(n^2/\gamma)$.
  Compute $X_e := Lap(\mu_0, \sigma_0)$.
  $w'(e) := w(e) + X_e$.
Compute all pairwise distances in graph $G' = (\mathbb{V}, \mathbb{E}' = \mathbb{E}_0 \cup \mathbb{E}_1, w')$.
Output : All pairwise distances of G'.

Algorithm 1: Private all pairwise shortest paths distance release.

Intuitively, the canonical shortest path finds a shortcut by directly connecting two nodes in a set $\mathbb{V}_1$. This definition will be the key in our algorithm and construction. One important fact is that, if we connect each pair of nodes in $\mathbb{V}_1$ and assign the true pairwise distance between them as the edge weight (as in our main algorithm), then $d(P_{u,v}, w) \equiv d(P^{\mathbb{V}_1}_{u,v}, w)$ by definition.

3.1 Challenges and the New Algorithm

First, we revisit the prior approach to release all pairwise distances in the private weight model and the challenges. Since neighboring (total) weights differ by at most one unit in l_1 norm, the distance between any two nodes also changes by at most one unit. Releasing a single path can be done by computing the accurate shortest path distance between pairs of inputs and adding Laplace noise proportional to $1/\epsilon$, which is a trivial task in DP. However, releasing all distances privately is much more challenging, since the Laplace mechanism requires $Lap(n^2/\epsilon)$ noise (because of the n^2 queries), resulting in the $O(n)$ error eventually. As introduced in Section 1.1, in prior literature, there are two ways (e.g., Sealfon [2016]) to achieve this error level, either by adding noise to each edge or to the output distances. In this paper, we will focus on the first strategy.

The simple approach is as follows: 1) add $Lap(1/\epsilon)$ noise to each edge, i.e., $w' = w + Lap(1/\epsilon)$, to get graph $G' = (\mathbb{V}, \mathbb{E}, w')$; 2) report all pairwise distances on G' (note that, the shortest paths on G' found might be different from the true shortest paths in G). By the Laplace mechanism [Dwork, 2006], all the weights in graph G become differentially private. By the post-processing property of DP, all the output pairwise distances are also DP. Note that there are $O(n^2)$ pairs of vertices, so the number of edges is bounded by $O(n^2)$. By Lemma 2.3, with probability $1 - \gamma$, all $O(n^2)$ Laplace random variables will have magnitude bounded by $(1/\epsilon)\log(n^2/\gamma)$, so the length of every path in the released synthetic graph is within $n\log(n/\gamma)/\epsilon$ additive error, thus roughly $O(n)$.

Statistically, the problem can be described informally and approximately as: given a set of n^2 i.i.d. Laplace random variables, we want to bound the sum of the n variables among n^2 size-n subsets simultaneously. To our best knowledge, this problem has no solution sublinear in n in literature, and a straightforward additive bound exactly leads to the previous $O(n)$ error as mentioned above. In this section, we propose an algorithm that leverages the concept of canonical shortest paths (Definition Definition 3.4), which finally leads to $\tilde{O}(n^{1/2})$ additive approximation error.

As summarized in Algorithm 1, for a general weighted graph $G(\mathbb{V}, \mathbb{E}, w)$, our algorithm proceeds as follows:

1) Sample $n^{1/2}$ vertices from $\mathbb{V}$ uniformly to form set $\mathbb{V}_1$;
2) Create an edge set $\mathbb{E}_1 := \{(u, v) : u \in \mathbb{V}_1, v \in \mathbb{V}_1, u \neq v\}$. For each $e = (u, v) \in \mathbb{E}_1$, set $w(e) = d(u, v)$ as the true shortest path distance;

3) Add $Lap(\mu_1, \sigma_1)$ noise to each edge $e \in \mathbb{E}_1$, i.e., $w' = w + Lap(\mu_1, \sigma_1)$;

4) Add $Lap(\mu_0, \sigma_0)$ noise to each edge $e \in \mathbb{E}_0$, i.e., $w' = w + Lap(\mu_0, \sigma_0)$ where $\mathbb{E}_0 := \mathbb{E} \backslash \mathbb{E}_1$;

5) Obtain and release the merged graph $G' = (\mathbb{V}, \mathbb{E}' = \mathbb{E}_0 \cup \mathbb{E}_1, w')$. One can simply compute all pairwise distances on G' as the estimation.

Comments. Note that in Algorithm 1, the final estimated distances are calculated by using standard APSP distance computation on graph G'. One implication of this construction is that, beyond outputting the pairwise distances with privacy, we can in fact also publish the graph $G'(\mathbb{V}, \mathbb{E}', w')$ privately and allow the users to compute the private distances using standard algorithms by themselves. This is due to the positive mean of the Laplace noises. If we instead add zero-mean (centered) Laplace noises as in most standard approaches, then computing the APSP distances on G' would not ensure the desired approximation error rate. We will provide more discussion on the role of the shifted noises at the end of this section.

Next, we provide theoretical analysis of our proposed algorithm. Firstly, recall that $\mathbb{V}_1$ contains $n^{1/2}$ uniformly sampled nodes from $\mathbb{V}$. Our analysis starts with the following fact.

Lemma 3.1 (Sampling Intersection)**.** *Suppose $U \subseteq \mathbb{V}$ is a fixed vertex set and $|U| = n^{1/2} \log(n^2/\gamma)$. With probability at least $1 - \gamma/n^2$, we have that $U \cap \mathbb{V}_1 \neq \emptyset$.*

3.2 Privacy Analysis and Sublinear Error Bound

Lemma 3.2. *Algorithm 1 achieves (ϵ, δ)-DP.*

Proof. The privacy budget is divided into two parts:

(Part 1) The noise added to $\mathbb{E}_0$. It is obvious that for two neighboring inputs differ in the total weights of edges in $\mathbb{E}_0$ by 1, by adding Laplace noises according $Lap(\mu_0, 1/\epsilon')$, it achieve $(\epsilon', 0)$-DP. Note that adding a constant μ_0 independent of the edge weights to the edges does not affect weight privacy.

(Part 2) The noise added to edges in $\mathbb{E}_1$. Similar arguments hold. There are at most $n^{1/2} \cdot n^{1/2} = n$ pairs in $\mathbb{E}_1$. Applying Laplace mechanism with composition theorem (Lemma 2.2), we know that adding noise following $Lap(\mu_1, \sigma_1)$ with $\sigma_1 = \sqrt{8n \log(1/\delta)}/\epsilon' = 2\sqrt{2} n^{1/2} \sqrt{\log(1/\delta)}/\epsilon'$ suffices to achieve (ϵ', δ)-DP. Applying simple DP composition theorem again proves the $(2\epsilon', \delta)$-DP. We conclude by noticing that $\epsilon' = \epsilon/2$. □

We have the following fact that with high probability, any long path in G would contain at least two nodes in $\mathbb{V}_1$ with high probability.

Lemma 3.3. *For a given path in $G(\mathbb{V}, \mathbb{E}, w)$ with number of edges larger than $2n^{1/2} \log(n^2/\gamma)$, it contains at least two vertices in $\mathbb{V}_1$ with probability $1 - 2\gamma/n^2$.*

Recall $G' = (\mathbb{V}, \mathbb{E}' = \mathbb{E}_0 \cup \mathbb{E}_1, w')$ is the constructed synthetic graph. Further define $G'_w = (\mathbb{V}, \mathbb{E}', w)$ as the graph with nodes $\mathbb{V}$, edges $\mathbb{E}'$ and weights w (notice that it is different from $G' = (\mathbb{V}, \mathbb{E}', w')$), and now consider those canonical shortest paths found by the true weights w on $G'_w(\mathbb{V}, \mathbb{E}', w)$. We show that each canonical shortest path $P^{\mathbb{V}_1}_{u,v}$, $u, v \in \mathbb{V}$ in G'_w has error $|d(P_{u,v}, w') - d(P_{u,v}, w)|$ bounded by $\tilde{O}(n^{1/2})$, where $d(P_{u,v}, w)$ is defined by Definition 3.3.

Lemma 3.4. *For any $u, v \in \mathbb{V}$, let $P^{\mathbb{V}_1}_{u,v}$ be the canonical shortest path (Definition 3.4) found by the true weights w. Then, it holds that $|d(P^{\mathbb{V}_1}_{u,v}, w') - d(P^{\mathbb{V}_1}_{u,v}, w)| = O(n^{1/2} \sqrt{\log(1/\delta)} \log^2(n/\gamma)/\epsilon)$ with probability $1 - 4\gamma$ for $\forall u, v \in \mathbb{V}$.*

Additionally, we have the following result stating that the estimated distance is no smaller than the true distance for all pairs of vertices, with high probability.

Lemma 3.5. *Let $P_{u,v}$ be the shortest path between $u, v \in \mathbb{V}$ on G, and $P'_{u,v}$ the shortest path found on G'. Then with probability $1 - 2\gamma$, $d(P'_{u,v}, w') \geq d(P_{u,v}, w)$ for all $u, v \in \mathbb{V}$.*

Now, we are in the position to present the main error guarantee of this work.

Theorem 3.6. *Let $G = (\mathbb{V}, \mathbb{E}, w)$ be general graph with n vertices. For some $\epsilon, \delta > 0$, running Algorithm 1 publishes the graph G' with (ϵ, δ)-differential privacy w.r.t. the weights w. Let $\hat{d}(u, v)$*

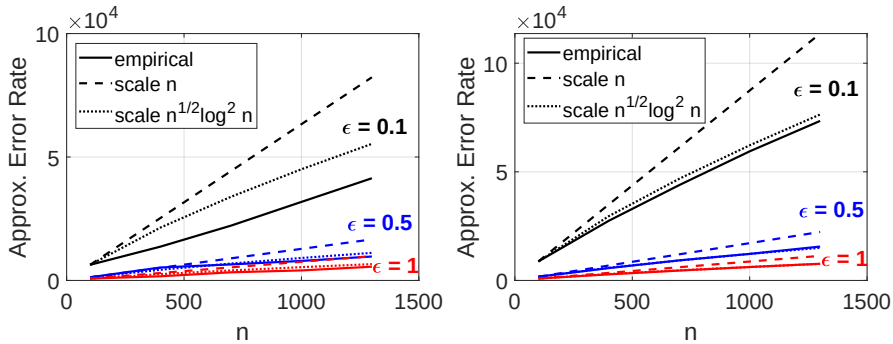

Figure 1: Empirical approx. error rate vs. graph size n (solid curve), with predicted growth rate of $O(n)$ and $O(n^{1/2}\log^2 n)$ (dash and dotted curves). $\delta = 0.01$, $\gamma = 0.01$. Left: each edge weight is $Unif(2000, 3000)$. Right:edge weight from $Unif(1e4, 1e5)$. Averaged over 200 repetitions.

be the estimated shortest path distance between $u, v \in \mathbb{V}$ computed on G'. Then, with probability $1 - 4\gamma$, we have $|\hat{d}(u,v) - d(u,v)| \leq O(\epsilon^{-1} n^{1/2} \log^2(n/\gamma)\sqrt{\log(1/\delta)})$ for all $u, v \in \mathbb{V}$.

Comments. As we see, the positive mean of Laplace noises implies Lemma 3.5 which provides a key inequality in the proof of Theorem 3.6. This result in fact allows us to only consider the error on the positive side. On the contrary, if we use mean-zero Laplace noises, then we can no longer bound $|d(P'_{u,v}, w') - d(u,v)|$ by $|d(P^{\mathbb{V}_1}_{u,v}, w') - d(u,v)|$ as in Lemma 3.4, since it is possible that $d(P'_{u,v}, w') \leq d(P^{\mathbb{V}_1}_{u,v}, w') < d(u,v)$. Thus, if we simply add zero-mean noises, directly computing the APSP distances on the constructed graph G' would not guarantee the desired error level.

3.3 Numerical Example

We present a numerical example on a multi-stage graph to justify the theory. We simulate a multi-stage graph, where each edge weight is generated from i.i.d. uniform distribution. Each stage contains 1 start node, 1 end node, and 9 nodes each connected to both the start and the end node. We consider multi-stage graphs here because in some sense it is one of the "hardest" cases for private APSP computation, since many shortest paths contain $O(n)$ edges. In Figure 1, we present the empirical largest absolute error over all pairwise distances against the size of graph n. For the theoretical bounds, we start with the first empirical error (black) associated with $n = 100$, and increase at the rate of n (dash) and $n^{1/2}\log^2 n$ (dotted), respectively. We validate that the empirical error grows with rate slower than $n^{1/2}\log^2 n$, as given by our theory, for all ϵ values. Note that the error bound becomes tighter with large edge weights (right) than with small weights (left). When the edge weights follow $Unif(1e4, 1e5)$ which is much larger than the noise magnitude, the shortest path $P'_{u,v}$ found on G' would be the same as $P_{u,v}$ (or $P^{\mathbb{V}_1}_{u,v}$) in most cases. Thus, the LHS and the RHS of (a) would be close, making the bound tighter.

4 Graph with Small Feedback Vertex Set

In previous section, we have investigated the problem of DP all pairwise shortest distance problem on general weighted graphs. In this section, we consider a more special type of graph with small feedback vertex set (FVS), which is a common concept in graph analysis, e.g., [Kratsch and Schweitzer, 2010, Papadopoulos and Tzimas, 2020]. We will design a new algorithm based on FVS computation that privately releases all pairwise distances. We will show that this method improves the error on general graphs when the FVS number is small. To begin with, the definition of FVS is given as below.

Definition 4.1 (Feedback Vertex Set [Becker and Geiger, 1996]). *Let $G = (\mathbb{V}, \mathbb{E})$ be an undirected graph. A set $X \subseteq \mathbb{V}$ is called a feedback vertex set (FVS) if $G \setminus X$ is a forest, where $G \setminus X$ is the induced graph by $\mathbb{V} \setminus X$. The feedback vertex set number is the minimal cardinality over all the possible feedback sets.*

In words, a feedback vertex set (FVS) of a graph is a subset of vertices X such that after removing these nodes from the graph, the subgraph induced by $\mathbb{V} \setminus X$ contains no cycles (i.e., is a forest). The FVS number is the smallest size of all the FVS's.

Input: General graph $G = (\mathbb{V}, \mathbb{E}, w)$, private parameter ϵ, δ, $\epsilon' = \epsilon/3$.
Compute the FVS S using the 2-approx. algorithm in Becker and Geiger [1996]. // Step 1)
Let G' be the induced subgraph of vertex set $\mathbb{V} \setminus S$ in G. Let $k := |S|$.
Compute the differentially private all pairwise shortest path distances $\hat{d}(u, v)$ on trees/forest G' using Algorithm 1 in Sealfon [2016] with privacy parameter ϵ'. // Step 2)
Compute $d(u, v)$ for $u, v \in S$, where $d(u, v)$ is the exact shortest path distance. // Step 3)
for *each* $u \in S, v \in S$ **do**
  Let $\sigma_1 := (2\sqrt{2}k\sqrt{\log(1/\delta)})/\epsilon'$.
  Compute $X_e := Lap(0, \sigma_1)$.
  $\hat{d}(u, v) := d(u, v) + X_e$.
for *each vertex* $u \in S$ **do**
  for *each vertex* $v \notin S$ *adjacent to* u **do**
    // Step 4): To deal with the edges between S and $\mathbb{V} \setminus S$
    Let $\sigma_0 := 1/\epsilon'$.
    Compute $X_e := Lap(0, \sigma_0)$.
    $w'(u, v) := w(u, v) + X_e$.
    $\hat{d}(u, v) := w'(u, v)$.
for *each vertex* u *in* $\mathbb{V} \setminus S$ **do**
  for *each vertex* v *in* S **do**
    // Step 4): To deal with the case that only the last vertex of shortest path is in S
    $\hat{d}(u, v) := \min_{p \notin S}\{\hat{d}(u, p) + w'(p, v)\}$.
for *each vertex* u *in* $\mathbb{V} \setminus S$ **do**
  for *each vertex* v *in* S **do**
    // Step 5): To deal with the case that shortest path passes a vertex p in S
    $\hat{d}(u, v) := \min_{p \in S}\{\hat{d}(u, p) + \hat{d}(p, v)\}$.
for *each pair* $u, v \in \mathbb{V} \setminus S$ **do**
  // Step 6): To deal with the case that shortest path passes some vertex u in S
  $\hat{d}(u, v) := \min_{p \in S}\{\hat{d}(u, p) + \hat{d}(p, v)\}$.
Output : All pairwise distances $\hat{d}(u, v)$.

Algorithm 2: Differentially private all pairwise shortest path distances release for graph with small feedback vertex number.

Next, we propose a differentially private algorithm that releases the APSP distances of graphs based on the feedback vertex set computation. The steps of Algorithm 2 are summarized below:

1) We compute a feedback vertex set S of G by using the 2-approximation algorithm in Becker and Geiger [1996], with $|S| = k$; the induced subgraph G' of vertex set $\mathbb{V} \setminus S$ is a forest;

2) We use Algorithm 1 in Sealfon [2016] to obtain the private all pairwise shortest path distances in G'. For any pair (u, v) in G whose shortest path does not pass any vertex in S, $\hat{d}(u, v)$ on G' is already a good estimation for $d(u, v)$;

3) For the k^2 (recall $k = |S|$) pairwise distances of the node pairs in S, we can compute the APSP distances directly and add $Lap(0, \sigma_1)$ to achieve DP. We still use the $\hat{d}(u, v)$ to represent them;

4) For each edge (u, v) with $u \in S, v \in \mathbb{V} \setminus S$, we add noise according $Lap(0, \sigma_0 = 1/\epsilon')$ to $w(u, v)$. For a pair in $\{(u, v) | u \in \mathbb{V} \setminus S, v \in S\}$, if the shortest path between u and v does not pass other vertex in S except v, then there exist some neighbor p of v such that $d_G(u, v) = d_{G'}(u, p) + w(p, v)$ based on Lemma 4.1, where $d_G(u, v), d_{G'}(u, v)$ represent the shortest path distance between u and v in G and G' respectively. Hence we use $\hat{d}(u, v) := \min_{p \notin S}\{\hat{d}(u, p) + w'(p, v)\}$ to estimate $d(u, v)$;

5) For a pair in $\{(u, v) : u \in \mathbb{V} \setminus S, v \in S\}$, if the shortest path between u and v passes some vertex $p \in S$ except v, then there exist $p \in S, p' \notin S$ such that $d_G(u, v) = d_G(u, p) + d_G(p, v)$, and the shortest path between u and p does not pass any vertex in S except p based

on Lemma 4.1. We obtained $\hat{d}(u,p)$ in step 4). We then use $\hat{d}(u,v) := \min_{p\in S}\{\hat{d}(u,p) + \hat{d}(p,v)\}$ to estimate $d(u,v)$;

6) For pairs in $\{(u,v) \in (\mathbb{V}\setminus S)^2\}$, if the shortest path between u and v passes some vertex p in S, then $d_G(u,v) = d_G(u,p) + d_G(p,v)$. We can use $\hat{d}(u,v) := \min_{p\in S}\{\hat{d}(u,p) + \hat{d}(p,v)\}$ to estimate $d(u,v)$, also $\hat{d}(u,p), \hat{d}(p,v)$ had been computed in 4), 5).

The auxiliary lemma mentioned in step 4) and 5) is given below.

Lemma 4.1. *For a given graph $G(\mathbb{V},\mathbb{E},w)$, let G' be the induced subgraph of G with vertex set $\mathbb{V}\setminus S$. For any pair $\{(u,v)|u \in \mathbb{V}\setminus S, v \in S\}$: 1) if the shortest path between u and v does not pass other nodes in S, then there exist a neighbor p of v such that $d_G(u,v) = d_{G'}(u,p) + w(p,v)$; 2) if the shortest path $P_{u,v}$ between u and v does pass other nodes in S except v, then there exist vertices $p \in S, p' \notin S$ such that $d_G(u,v) = d_G(u,p) + d_G(p,v)$ and $d_G(u,p) = d_{G'}(u,p') + w(p',p)$.*

4.1 Privacy and Error Analysis

Lemma 4.2. *Algorithm 2 achieves (ϵ,δ)-DP.*

Proof. The privacy budget ϵ is divided into three parts:

(Part 1) For the distances between all pairs in $\mathbb{V}\setminus S$, our method achieves $(\epsilon',0)$-DP by the result on trees from [Sealfon, 2016].

(Part 2) For the k^2 distances of all node pairs in S, by adding to each edge weights i.i.d. $Lap(0,\sigma_1)$ noises with $\sigma_1 = \sqrt{8n\log(1/\delta)}/\epsilon' = 2\sqrt{2}n^{1/2}\sqrt{\log(1/\delta)}/\epsilon'$, we can achieve (ϵ',δ)-DP according to Lemma 2.2.

(Part 3) For each edge (u,v) with $u \in \mathbb{V}\setminus S, v \in S$, obviously by Laplace mechanism $w'(u,v) := w(u,v) + Lap(1/\epsilon')$, we achieve $(\epsilon',0)$-DP to release all the pairwise distances.

Composing three privacy budges up and using simple composition theorem of DP, we show that Algorithm 2 achieves (ϵ,δ)-DP. □

The error bound of Algorithm 2 is given as below.

Theorem 4.3. *Let $G = (\mathbb{V},\mathbb{E},w)$ be a general graph with n vertices. For some $\epsilon,\delta > 0$, running Algorithm 2 publishes all APSP distances that is (ϵ,δ)-differentially private w.r.t. the weights w. With probability $1-3\gamma$, the additive error is bounded by $O(\epsilon^{-1}k\log(k/\gamma)\sqrt{\log(1/\delta)} + \epsilon^{-1}(\log^{2.5} n)\log(1/\gamma))$.*

Comments. We see that when $k = |S| = o(n^{1/2})$, the error in Theorem 4.3 is smaller than the $O(n^{1/2}\log^2 n)$ error in Theorem 3.6 for the general weighted graphs. Therefore, our result implies improved error bounds when the graph has small feedback vertex set such that $k = o(n^{1/2})$. Further, if $k = o(\log n)$, i.e., the graph is "almost" a forest, then the first term vanishes and the error reduces to $\tilde{O}(\log^{2.5} n)$ for trees [Sealfon, 2016].

5 Conclusion

In literature, to achieve (ϵ,δ)-differential privacy (DP) in general weight private graphs, releasing all pairwise shortest path (APSP) distances incurs maximal approximation error $\tilde{O}(n)$, linear in the number of vertices. Recently, Fan and Li [2022] studied this problem on grid graphs and trees, but the linear error barrier for general weighted graphs is still an open problem [Sealfon, 2020]. In this work, we propose a differentially private algorithm to release a carefully constructed graph, and computing the APSP distance on the synthetic graph achieves $\tilde{O}(n^{1/2})$ error sublinear in n, thus answering the open question. With improved and extended techniques from Fan and Li [2022], the idea of our approach is to augment the diameter of graph by a set of shortcuts along with adding shifted Laplace noise. Moreover, for graphs with small feedback vertex set number k, we also propose a DP algorithm to answer all pairwise shortest path distances each with error $\tilde{O}(k)$. This improves the result for general graphs when k, the feedback vertex set number, is small.

References

Raman Arora and Jalaj Upadhyay. On differentially private graph sparsification and applications. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 13378–13389, Vancouver, Canada, 2019.

Ann Becker and Dan Geiger. Optimization of pearl's method of conditioning and greedy-like approximation algorithms for the vertex feedback set problem. *Artif. Intell.*, 83(1):167–188, 1996.

Jeremiah Blocki, Avrim Blum, Anupam Datta, and Or Sheffet. Differentially private data analysis of social networks via restricted sensitivity. In *Proceedings of the Innovations in Theoretical Computer Science (ITCS)*, pages 87–96, Berkeley, CA, 2013.

Avrim Blum, Cynthia Dwork, Frank McSherry, and Kobbi Nissim. Practical privacy: the sulq framework. In *Proceedings of the Twenty-fourth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems (PODS)*, pages 128–138, Baltimore, MD, 2005.

Christian Borgs, Jennifer T. Chayes, Adam D. Smith, and Ilias Zadik. Revealing network structure, confidentially: Improved rates for node-private graphon estimation. In *Proceedings of the 59th IEEE Annual Symposium on Foundations of Computer Science (FOCS)*, pages 533–543, Paris, France, 2018.

Mark Bun, Kobbi Nissim, Uri Stemmer, and Salil P. Vadhan. Differentially private release and learning of threshold functions. In *Proceedings of the IEEE 56th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 634–649, Berkeley, CA, 2015.

Shuchi Chawla, Cynthia Dwork, Frank McSherry, Adam D. Smith, and Hoeteck Wee. Toward privacy in public databases. In *Proceedings of the Second Theory of Cryptography Conference (TCC)*, pages 363–385, Cambridge, MA, 2005.

Cynthia Dwork. Differential privacy. In *Proceedings of the 33rd International Colloquium on Automata, Languages and Programming (ICALP), Part II*, pages 1–12, Venice, Italy, 2006.

Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam D. Smith. Calibrating noise to sensitivity in private data analysis. In *Proceedings of the Third Theory of Cryptography Conference (TCC)*, pages 265–284, New York, NY, 2006.

Cynthia Dwork, Guy N. Rothblum, and Salil P. Vadhan. Boosting and differential privacy. In *Proceedings of the 51th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 51–60, Las Vegas, NV, 2010.

Chenglin Fan and Ping Li. Distances release with differential privacy in tree and grid graph. *arXiv preprint arXiv:2204.12488, to appear in ISIT*, 2022.

Eugene C. Freuder. A sufficient condition for backtrack-free search. *J. ACM*, 29(1):24–32, 1982.

Anupam Gupta, Katrina Ligett, Frank McSherry, Aaron Roth, and Kunal Talwar. Differentially private combinatorial optimization. In *Proceedings of the Twenty-First Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1106–1125, Austin, TX, 2010.

Anupam Gupta, Aaron Roth, and Jonathan R. Ullman. Iterative constructions and private data release. In *Proceedings of the 9th Theory of Cryptography Conference (TCC)*, pages 339–356, Taormina, Sicily, Italy, 2012.

Michael Hay, Chao Li, Gerome Miklau, and David D. Jensen. Accurate estimation of the degree distribution of private networks. In *Proceedings of the Ninth IEEE International Conference on Data Mining (ICDM)*, pages 169–178, Miami, FL, 2009.

Vishesh Karwa, Sofya Raskhodnikova, Adam D. Smith, and Grigory Yaroslavtsev. Private analysis of graph structure. *Proc. VLDB Endow.*, 4(11):1146–1157, 2011.

Shiva Prasad Kasiviswanathan, Kobbi Nissim, Sofya Raskhodnikova, and Adam D. Smith. Analyzing graphs with node differential privacy. In *Proceedings of the 10th Theory of Cryptography Conference (TCC)*, pages 457–476, Tokyo, Japan, 2013.

Stefan Kratsch and Pascal Schweitzer. Isomorphism for graphs of bounded feedback vertex set number. In *Proceedings of the 12th Scandinavian Symposium and Workshops on Algorithm Theory (SWAT)*, pages 81–92, Bergen, Norway, 2010.

John M. Lewis. On the complexity of the maximum subgraph problem. In *Proceedings of the 10th Annual ACM Symposium on Theory of Computing (STOC)*, pages 265–274, San Diego, CA, 1978.

Charis Papadopoulos and Spyridon Tzimas. Subset feedback vertex set on graphs of bounded independent set size. *Theor. Comput. Sci.*, 814:177–188, 2020.

Vibhor Rastogi, Michael Hay, Gerome Miklau, and Dan Suciu. Relationship privacy: output perturbation for queries with joins. In *Proceedings of the Twenty-Eigth ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems (PODS)*, pages 107–116, Providence, RI, 2009.

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

Adam Sealfon. Open problem - private all-pairs distances, 2020. https://differentialprivacy.org/open-problem-all-pairs/.

Jonathan R. Ullman and Adam Sealfon. Efficiently estimating erdos-renyi graphs with node differential privacy. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 3765–3775, Vancouver, Canada, 2019.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes]
   (c) Did you discuss any potential negative societal impacts of your work? [Yes]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes]
   (b) Did you include complete proofs of all theoretical results? [Yes]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results? [Yes]
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes]
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes]
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [N/A]
   (b) Did you mention the license of the assets? [N/A]
   (c) Did you include any new assets either in the supplemental material or as a URL? [N/A]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]