
Differentially Private Densest Subgraph Detection

Dung Nguyen^{1 2} Anil Vullikanti^{1 2}

Abstract

Densest subgraph detection is a fundamental graph mining problem, with a large number of applications. There has been a lot of work on efficient algorithms for finding the densest subgraph in massive networks. However, in many domains, the network is private, and returning a densest subgraph can reveal information about the network. Differential privacy is a powerful framework to handle such settings. We study the densest subgraph problem in the edge privacy model, in which the edges of the graph are private. We present the first sequential and parallel differentially private algorithms for this problem. We show that our algorithms have an additive approximation guarantee. We evaluate our algorithms on a large number of real-world networks, and observe a good privacy-accuracy tradeoff when the network has high density.

1. Introduction

Data privacy is a fundamental challenge in many real world applications, e.g., healthcare, social networks, and finance, where there is a risk of revealing private information through adversarial queries. Differential privacy (defined in Section 2), developed through the work of a number of researchers, e.g., (Dwork, 2011; Blum et al., 2005; Dwork et al., 2014; Nissim et al., 2007), has proven to be a very powerful approach to support diverse kinds of computations with rigorous privacy guarantees (see (Zhu et al., 2017; Vadhan, 2017) for extensive surveys on this topic). This framework allows database owners to support queries with a very controlled and rigorous loss of privacy. Differentially private algorithms have now been designed for a number of problems, including supervised and unsupervised machine learning, social network analysis and deep learning (Kasiviswanathan et al., 2013; Blocki et al., 2013; Abadi et al., 2016; Abowd,

2018). However, network problems have been proven to be much harder, and private algorithms for many problems (e.g., in graph mining) remain poorly understood. For some of the problems which have been studied, e.g., (Nguyen et al., 2016), good accuracy bounds are not known. Further, most private algorithms do not scale very well, especially for network problems, e.g., (Kasiviswanathan et al., 2013; Blocki et al., 2013), and there has been limited work on parallel algorithms.

Here, we study the problem of finding the densest subgraph in a graph $G(V, E)$, which is a very basic subroutine in graph mining, and has been used in diverse domains, including bioinformatics, network science, fraud detection, and social network analysis, e.g., (Cadena et al., 2018; Hooi et al., 2016; Khuller & Saha, 2009; Cadena et al., 2016; Tsourakakis et al., 2013; Rozenshtein et al., 2014). There are many notions of density (Cadena et al., 2018); here, we focus on the notion of average density $\rho(S)$ of $S \subset V$, defined as $\rho(S) = (\text{\#edges with both end points in } S)/|S|$. The goal is to find a densest subgraph $S^* = \operatorname{argmax}_{S \subseteq V} \rho(S)$; we use $\rho(G) = \rho(S^*)$ to indicate the density of the densest subgraph in G . This notion of density is one of the most common studied ones, e.g., for anomaly detection in networked data (Cadena et al., 2018; Hooi et al., 2016), primarily because it can be computed very efficiently (Charikar, 2000; Asahiro et al., 2002; Khuller & Saha, 2009; Goldberg, 1984). In particular, the densest subgraph can be computed optimally using linear programming (Goldberg, 1984; Charikar, 2000), and a simple iterative greedy algorithm gives a $\frac{1}{2}$ -approximation (Charikar, 2000; Asahiro et al., 2002). Highly scalable parallel algorithms have also been developed, e.g., (Bahmani et al., 2012; Ghaffari et al., 2019). However, private algorithms are not known for the dense subgraph problem, which is the focus of this paper.

There are two standard approaches for privacy in networks—edge privacy (in which the nodes are public and the edges are private) and node privacy (in which the nodes are also private). As we observe later, the maximum density value has low sensitivity can be computed easily using the Laplace mechanism. However, in graph mining settings, analysts are interested in finding the actual densest subgraph (i.e., the nodes in the subgraph), which is the focus of our paper, and is a much harder problem. It only makes sense in the edge privacy model. Our contributions are summarized below.

¹Department of Computer Science, University of Virginia, Virginia, USA ²Biocomplexity Institute and Initiative, University of Virginia, Virginia, USA. Correspondence to: Dung Nguyen <dungn@virginia.edu>.

- We first present a sequential (ϵ, δ) -differentially private algorithm (SEQDENSEDP) for the densest subgraph problem in the edge privacy model. We adapt Charikar’s algorithm (Charikar, 2000), using the exponential mechanism (McSherry & Talwar, 2007; Gupta et al.) iteratively, and prove that our mechanism achieves the same multiplicative approximation factor as (Charikar, 2000), with an additive logarithmic factor, i.e., the solution S computed by SEQDENSEDP satisfies $\rho(S) \geq \rho(S^*)/2 - O(\log n)$, with high probability.
- We alter the sampling process in SEQDENSEDP, and obtain a parallel algorithm (PARDENSEDP), which involves sampling multiple nodes in parallel, in each iteration. This works well in practice, in some regimes, but can have exponential time in the worst case. Our final algorithm (PHASEDENSEDP) groups multiple iterations into a single sampling phase (basically guessing when a node would be sampled). We prove that PHASEDENSEDP takes $O(\log n)$ phases, with high probability. Both PARDENSEDP and PHASEDENSEDP have similar accuracy as SEQDENSEDP, though the constant factors get worse. We also design a distributed version of PHASEDENSEDP in the Map-Reduce model (Dean & Ghemawat, 2008; Karloff et al., 2010; Beame et al., 2013). We show it has similar accuracy and runtime guarantees as PHASEDENSEDP.
- We evaluate our algorithms on a large number of diverse networks, in order to understand the privacy-accuracy tradeoffs. We find that in networks with high densities, our private algorithms have good accuracy, especially for larger values of ϵ . Our algorithms also have reasonable recall, i.e., the private solution contains a large fraction of the optimal subgraph. SEQDENSEDP has better accuracy than PARDENSEDP, which, in turn, is better than PHASEDENSEDP. In some regimes of ϵ , PARDENSEDP has significantly better accuracy than PHASEDENSEDP, and is comparable to SEQDENSEDP.
- We show an additive lower bound of $\Omega(\sqrt{\log n})$ for any differentially private algorithm for DENSEDP.

Due to the limited space, we have omitted many proofs and other discussion; these are presented in the Appendix. The full version including the Appendix is available at (Nguyen & Vullikanti, 2021). We note that our analysis is not very tight, in order to simplify the discussion; we expect the constant factors can be tightened with a more careful analysis.

We use the exponential mechanism (McSherry & Talwar, 2007) in an iterative manner, which involves choosing elements from a set with probability that is proportional to

a score associated with the element. This approach has been used in private versions of other combinatorial optimization problems, e.g., vertex cover, global minimum cut and set cover, e.g., (Gupta et al.). However, this is inherently sequential, and we develop a parallel version, in which multiple elements can be chosen simultaneously. Consequently, the analysis of the performance becomes much more challenging, and is one of our contributions.

2. Preliminaries

2.1. Problem statement

Densest subgraph. Let $G = (V, E)$ denote an undirected graph. For a subset of nodes $S \subset V$, let $E[S]$ denote the set of edges with both end points in S . The density of a set S in G is defined as $\rho(S) = \frac{|E[S]|}{|S|}$ (Charikar, 2000; Khuller & Saha, 2009). The objective is to find a subset $S^*(G) = \operatorname{argmax}_{S \subset V} \rho(S)$, which achieves the maximum density; we refer to it as S^* when the graph is clear from the context. Let $\rho(G) = \rho(S^*)$. We say that $\rho(S)$ is an α -approximate solution if $\rho(S) \geq \alpha \rho(S^*)$. An optimal densest subgraph (i.e., $\alpha = 1$) can be computed optimally using a linear programming based algorithm, and an iterative algorithm gives a $\alpha = 1/2$ -approximation (Charikar, 2000; Khuller & Saha, 2009). However, this turns out to be difficult to achieve under privacy, and we consider a slightly relaxed version: we say S is an (α, β) -approximation if $\rho(S) \geq \alpha \rho(S^*(G)) - \beta$. Let $N^G(v)$ denote the set of neighbors of v in G , and let $\deg^G(v) = |N^G(v)|$ denote the degree of v . For a subset $S \subset V$, let $\deg_S^G(v) = |N(v) \cap S|$ denote the number of neighbors of v in S in graph G ; when G is clear from the context, we denote it by $\deg_S(v)$.

Differential privacy on graphs. Let \mathcal{G} denote a set of graphs on a fixed set V of nodes. For a graph $G \in \mathcal{G}$, we use $V(G)$ and $E(G)$ to denote the set of nodes and edges of G , respectively. In this paper, we will focus on the notion of *Edge privacy* (Blocki et al., 2013), where all graphs $G \in \mathcal{G}$ have a fixed set of nodes $V(G) = V$, and two graphs $G, G' \in \mathcal{G}$ are considered neighbors, i.e., $G \sim G'$, if they differ in exactly one edge, i.e., $|E(G) - E(G')| = 1$. We note that another notion that has been considered is *Node privacy* (Kasiviswanathan et al., 2013): two graphs $G, G' \in \mathcal{G}$ are considered neighbors, i.e., $G \sim G'$, if they differ in exactly one node, i.e., $|V(G) - V(G')| = 1$. Note that they might differ in many edges.

Definition 2.1. A (randomized) algorithm $M : \mathcal{G} \rightarrow R$ is (ϵ, δ) -differentially private if for all subsets $S \subset R$ of its output space, and for all $G, G' \in \mathcal{G}$, with $G \sim G'$, we have $\Pr[M(G) \in S] \leq e^\epsilon \Pr[M(G') \in S] + \delta$ (Dwork et al., 2014; Dwork, 2011; Vadhan, 2017; Blocki et al., 2013).

Problem statement: densest subgraph detection with edge differential privacy (DENSEDP) Given a family of

graphs \mathcal{G} on a set V of vertices, and parameters ϵ, δ , construct an (ϵ, δ) -differentially private mechanism $M : \mathcal{G} \rightarrow 2^V$, such that: (1) for any graphs $G \sim G'$, and $\mathcal{S} \subset 2^V$, $\Pr[M(G) \in \mathcal{S}] \leq e^\epsilon \Pr[M(G') \in \mathcal{S}] + \delta$, and (2) $\rho(M(G))$ is maximized.

We evaluate the accuracy of the mechanism M (in the second condition above) in terms of the (α, β) -approximation to $\rho(S^*(G))$ for any graph $G \in \mathcal{G}$; therefore, we would like α to be as large as possible, and β to be as small as possible.

Computing density value vs finding a subgraph. The DENSEDP problem involves finding a subset $M(G) \subset V$, where V is not private, but only the edges are private. It can be verified easily that the ‘sensitivity’ (Definition 2.2) $\Delta_\rho = \max_{G \sim G'} |\rho(S^*(G)) - \rho(S^*(G'))| \leq 1$, as the addition or removal of an edge can be shown to alter the density by at most 1. As a result, private computation of $\rho(S^*(G))$ can be done easily with the standard Laplacian mechanism $\text{Lap}(1)$ (Dwork et al., 2014). However, returning a subgraph with high density, and with privacy is a harder problem.

2.2. Additional background on differential privacy

Let $A_1 : \mathcal{D}_1 \rightarrow \mathcal{R}_1$ and $A_2 : \mathcal{D}_2 \rightarrow \mathcal{R}_2$ be (ϵ_1, δ_1) and (ϵ_2, δ_2) -differentially private algorithms, respectively. Let $f : \mathcal{R}_1 \rightarrow \mathcal{R}'_1$ be an arbitrary randomized mapping. We will make extensive use of the following basic results shown in (Dwork et al., 2014), that differential privacy is preserved under:

(1) post-processing (Proposition 2.1 of (Dwork et al., 2014)), i.e., $f \circ A_1 : \mathcal{D} \rightarrow \mathcal{R}'_1$ is also (ϵ, δ) -differentially private, and

(2) composition (Theorem 3.16 of (Dwork et al., 2014)), i.e., $A : (\mathcal{D}_1, \mathcal{D}_2) \rightarrow (\mathcal{R}_1, \mathcal{R}_2)$ defined as $A(x) = (A_1(x), A_2(x))$ is $(\epsilon_1 + \epsilon_2, \delta_1 + \delta_2)$ -differentially private.

Definition 2.2. Sensitivity of utility function (Dwork et al., 2014). Given a dataset space \mathcal{D} and an arbitrary range \mathcal{R} , a function $u : \mathcal{D} \times \mathcal{R} \rightarrow \mathbb{R}$ has a global sensitivity Δ_u defined as: $\Delta_u = \max_{r \in \mathcal{R}} \max_{x \sim x'} |u(x, r) - u(x', r)|$

Definition 2.3. The Exponential Mechanism. (Definition 3.4 of (Dwork et al., 2014)) The exponential mechanism $M(x, u, \mathcal{R})$ that outputs an element $r \in \mathcal{R}$ with probability proportional to $\exp(\frac{\epsilon u(x, r)}{\Delta_u})$ is ϵ -differentially private, if the addition of an element to the data does not decrease the value of the utility function.

Theorem 2.4. Utility of the Exponential Mechanism. (Theorem 3.11 and Corollary 3.12 of (Dwork et al., 2014)) For a given dataset x , let $\text{OPT} = \max_{r \in \mathcal{R}} u(x, r)$. For the exponential mechanism $M(\cdot)$, we have:

$$\Pr[u(x, M(x, u, \mathcal{R})) \leq \text{OPT} - \frac{2\Delta_u}{\epsilon} (\ln |\mathcal{R}| + r)] \leq e^{-r}$$

Adversarial Probabilistic Process. Following (Gupta et al.), consider an adversarial probabilistic process with

n iterations. In each iteration i , the adversary tosses a coin, which gives heads with probability $p_i \in [0, 1]$, adaptively by observing previous $i-1$ iterations. We utilize this process to prove the privacy bounds of our algorithms.

Lemma 2.5. Lemma B.1 of (Gupta et al.). Let Z_i be a random variable, with $Z_i = 1$ indicating that no head appears in the first i iterations, and $Z_i = 0$ otherwise. Let $Y = \sum_{i=1}^n p_i Z_i$. For any q , $\Pr[Y > q] \leq \exp(-q)$.

Corollary 2.5.1. Let \mathcal{T} be the first iteration in which a head appears in the process above. For any $\delta \in (0, 1]$, $\sum_{i=1}^{\mathcal{T}-1} p_i \leq \ln \delta^{-1}$, with probability at least $1 - \delta$.

3. Related Works

Differential privacy has been a very active area of research since its introduction. We briefly summarize the main results and challenges; we focus only on works on privacy for graph that are directly relevant to our paper. Due to space constraints, we refer to (Dwork, 2011; Blum et al., 2005; Dwork et al., 2014; Nissim et al., 2007; Zhu et al., 2017; Vadhan, 2017) for surveys and details of key results in this area, and provide additional discussion in the Appendix.

Much of the initial work on graph privacy involved computing different kinds of statistics, e.g., number of edges, counts of triangles and other subgraphs, and the cost of the minimum spanning tree (Nissim et al., 2007; Karwa et al., 2014a;b; Mir & Wright, 2012; Hay et al., 2009; Kasiviswanathan et al., 2013; Blocki et al., 2013). Graph statistics are challenging due to their high sensitivity. (Nissim et al., 2007) introduce a concept of *smooth sensitivity* and apply it for triangle counting and minimum spanning tree problems in the edge privacy model. (Karwa et al., 2014a) generalize the method for other subgraph counting problems, such as k-star or k-triangle. Many other graph statistics in the edge-privacy model have been studied in (Karwa et al., 2014b; Mir & Wright, 2012; Hay et al., 2009). On the other hand, graph statistics in the node-privacy model are generally more difficult. (Kasiviswanathan et al., 2013; Blocki et al., 2013) are two of the first studies that utilize Lipschitz Extensions to develop node differentially private mechanisms. They transform subgraph counting problems into linear programming optimizations of which the sensitivities of the solutions are restricted. Algorithms for node differentially private graph queries such as degree distributions or random graph model estimations have been established by (Chen & Zhou, 2013; Ding et al., 2018; Day et al., 2016).

In contrast to graph statistics, releasing a subset of nodes in the input graph with differential privacy (as in the st-mincut, k -median, or the vertex cover problems), or a collection of subsets in other combinatorial optimization problems (as in the set cover problem) is much more challenging, e.g., (Mitrovic et al., 2017; Gupta et al.). One of the first

papers on this was by (Gupta et al.), who design private algorithms for different combinatorial optimization problems, including computing global mincuts, vertex cover and set cover. Our algorithms are inspired by the techniques of (Gupta et al.).

Another direction of work is on generation of differentially private synthetic graphs (Xiao et al., 2014; Nguyen et al., 2015; Chen et al., 2014; Gupta et al., 2012), which preserve some specific graph properties, such as cut, shortest path length and degree distribution queries. Such synthetic graphs are useful since post-processing does not cause any privacy violation. Finally, we note that private graph algorithms have also been considered for other notions of privacy, e.g. (Imola et al., 2020).

4. Private Densest Subgraph

Here we design and analyze SEQDENSEDP for the DENSEDP problem. This is an adaptation of the (non private) algorithm Charikar (Charikar, 2000), which gives a $1/2$ -approximation for the densest subgraph problem—this algorithm constructs a sequence of subgraphs S_1, \dots, S_n (by removing a minimum degree node each time), and then chooses $\arg\max_{S_i} \rho(S_i)$. SEQDENSEDP adapts this algorithm, and involves two parts. First, it generates a sequence of candidate subgraphs by removing one node at a time using an Exponential Mechanism, with probability proportional to an exponential function of its current degree times a constant. Therefore, lower-degree nodes have exponentially more chances to be removed than higher-degree nodes. Intuitively, we try to remove low-degree nodes and retain high-degree nodes to construct dense candidates. The second part of the algorithm applies the Exponential Mechanism to randomly select one of the candidates by their densities.

Algorithm 1 SEQDENSEDP(G, ϵ, δ)

Input: $G \in \mathcal{G}$, privacy parameters ϵ, δ .

Output: Subset $S \subseteq V(G)$

- 1: Let $\epsilon' = \frac{\epsilon}{4 \ln(e/\delta)}$, $S_0 = V(G)$
- 2: **for** $t := 1$ to n **do**
- 3: Pick a node $\pi_t \in \{v : v \in S_{t-1}\}$ with probability proportional to $e^{-\epsilon' \deg_{S_{t-1}}(v)}$
- 4: $S_t = S_{t-1} - \pi_t$
- 5: **end for**
- 6: Return S_t from $\{S_t : t = 0, \dots, n-1\}$ with probability proportional to $e^{\epsilon \rho(S_t)/2}$

Lemma 4.1. *The computation of the sequence $\pi^G = \pi_1, \dots, \pi_n$, in the **for** loop in Algorithm 1 (lines 3–5) is $(\epsilon/2, \delta)$ -differentially private.*

Proof. (Short, Full proof in Theorem A.1) Let $G \sim G'$ be two graphs that differ in exactly one edge $e = (u, v)$. Let π^G and $\pi^{G'}$ denote the permutations computed in the for

loop in Algorithm 1. Let \mathcal{T} is the first iteration that one of the endpoints of e (i.e., u or v) is removed. Let $\text{adj}(e)$ define the set of nodes adjacent to edge e . Recall that $\deg_{S_t}^G(j)$ is the degree of node j in subgraph S_t of graph G . Fix a permutation π . Consider the ratio:

$$\begin{aligned} P &= \frac{\Pr[\pi^G = \pi]}{\Pr[\pi^{G'} = \pi]} \\ &= \prod_{t=1}^n \frac{\exp(-\epsilon' \deg_{S_t}^G(\pi_t)) / \sum_j \exp(-\epsilon' \deg_{S_t}^G(j))}{\exp(-\epsilon' \deg_{S_t}^{G'}(\pi_t)) / \sum_j \exp(-\epsilon' \deg_{S_t}^{G'}(j))} \\ &= \frac{\exp(-\epsilon' \deg_{S_{\mathcal{T}}}^G(\pi_{\mathcal{T}}))}{\exp(-\epsilon' \deg_{S_{\mathcal{T}}}^{G'}(\pi_{\mathcal{T}}))} \prod_{t=1}^{\mathcal{T}} \frac{\sum_j \exp(-\epsilon' \deg_{S_t}^G(j))}{\sum_j \exp(-\epsilon' \deg_{S_t}^{G'}(j))} \end{aligned}$$

The last equality follows from the definition of \mathcal{T} , because: (i) for $t = \mathcal{T} + 1..n$, $\deg_{S_t}^G(j) = \deg_{S_t}^{G'}(j)$ for all nodes j , and, (ii) for $t = 1..\mathcal{T} - 1$, $\deg_{S_t}^G(\pi_t) = \deg_{S_t}^{G'}(\pi_t)$. We have two cases for the analysis.

First, suppose $G + e = G'$. In this case, $\frac{\exp(-\epsilon' \deg_{S_{\mathcal{T}}}^G(\pi_{\mathcal{T}}))}{\exp(-\epsilon' \deg_{S_{\mathcal{T}}}^{G'}(\pi_{\mathcal{T}}))} = \exp(\epsilon')$ because $\deg_{S_{\mathcal{T}}}^G(\pi_{\mathcal{T}}) + 1 = \deg_{S_{\mathcal{T}}}^{G'}(\pi_{\mathcal{T}})$. Further, for each t, j , we have $\deg_{S_t}^{G'}(j) \geq \deg_{S_t}^G(j)$ for all j , which implies the product $\prod_{t=1}^{\mathcal{T}} \frac{\sum_j \exp(-\epsilon' \deg_{S_t}^G(j))}{\sum_j \exp(-\epsilon' \deg_{S_t}^{G'}(j))} \leq 1$. Therefore, $P \leq \exp(\epsilon')$

Second, suppose $G - e = G'$. Similar to the first case, $\frac{\exp(-\epsilon' \deg_{S_{\mathcal{T}}}^G(\pi_{\mathcal{T}}))}{\exp(-\epsilon' \deg_{S_{\mathcal{T}}}^{G'}(\pi_{\mathcal{T}}))} = \exp(-\epsilon') \leq 1$. For $j \in \text{adj}(e), t \leq \mathcal{T}$, we have $\deg_{S_t}^G(j) = \deg_{S_t}^{G'}(j) + 1$. For $j \notin \text{adj}(e)$, and for all t , $\deg_{S_t}^G(j) = \deg_{S_t}^{G'}(j)$. Further, for $t > \mathcal{T}$, we have $\deg_{S_t}^G(j) = \deg_{S_t}^{G'}(j)$ for all j . Note that whenever we use $\deg_{S_t}^G(j)$, we only consider $j \in S_{t-1}$, i.e., j would not have been deleted from the graph before step t . Therefore, for $t \leq \mathcal{T}$: we can expand the term $\sum_j \exp(-\epsilon' \deg_{S_t}^G(j))$ as $\sum_j \exp(-\epsilon' \deg_{S_t}^{G'}(j)) = (e^{\epsilon'} - 1) \sum_{j \in \text{adj}(e)} \exp(-\epsilon' \deg_{S_t}^G(j)) + \sum_j \exp(-\epsilon' \deg_{S_t}^{G'}(j))$. Substituting it in the expression for P . Since $\deg_{S_t}^G(j) = \deg_{S_t}^{G'}(j)$ for all j and $t > \mathcal{T}$, we have

$$\begin{aligned} P &\leq \prod_{t=1}^{\mathcal{T}} (1 + (\exp(\epsilon') - 1) \frac{\sum_{j \in \text{adj}(e)} \exp(-\epsilon' \deg_{S_t}^G(j))}{\sum_j \exp(-\epsilon' \deg_{S_t}^{G'}(j))}) \\ &= \prod_{t=1}^{\mathcal{T}} (1 + (\exp(\epsilon') - 1) p_t(G)) \\ &\leq \prod_{t=1}^{\mathcal{T}} \exp((\exp(\epsilon') - 1) p_t(G)), \text{ using } 1 + x \leq e^x \text{ for } x \geq 0, \end{aligned}$$

where $p_t(G)$ is the probability that an endpoint of e is picked in iteration t . We will show in Lemma 4.2, by using the adversarial process (Section 2.2), that $\sum_{t=1}^{\mathcal{T}-1} p_t(G) \leq \ln \delta^{-1}$ with prob. at least $1 - \delta$ and use it to derive a bound on P . We say that π is $\ln \delta^{-1}$ -good if $\sum_{t=1}^{\mathcal{T}-1} p_t(G) \leq \ln \delta^{-1}$ and $\ln \delta^{-1}$ -bad otherwise (*good* and *bad* for short). Therefore,

with probability at least $1 - \delta$, π is good and

$$\begin{aligned} P &\leq \prod_{t=1}^{\mathcal{T}} \exp((\exp(\epsilon') - 1)p_t(G)) \\ &\leq \exp(2\epsilon' \sum_{t=1}^{\mathcal{T}} p_t(G)), \text{ using } e^x \leq 1 + 2x \text{ for } x \leq 1 \\ &\leq \exp(2\epsilon'(\ln \delta^{-1} + p_{\mathcal{T}}(G))) \leq \exp(2\epsilon'(\ln \delta^{-1} + 1)) \\ &= \exp(2\epsilon'(\ln(e/\delta))) = \exp(\epsilon/2), \text{ since } \epsilon' = \frac{\epsilon}{4 \ln(e/\delta)}, \end{aligned}$$

which implies $\Pr[\pi^G = \pi] \leq \exp(\epsilon/2) \Pr[\pi^{G'} = \pi]$.

Let \mathcal{P} be the set of output orderings that might be generated by the **for** loop in SEQDENSEDP. Then, we have

$$\begin{aligned} \Pr[\pi^G \in \mathcal{P}] &= \sum_{\pi \in \mathcal{P}} \Pr[\pi^G = \pi] \\ &= \sum_{\pi \in \mathcal{P}: \pi \text{ is good}} \Pr[\pi^G = \pi] + \sum_{\pi \in \mathcal{P}: \pi \text{ is bad}} \Pr[\pi^G = \pi] \\ &\leq \exp(\epsilon/2) \Pr[\pi^{G'} \in \mathcal{P}] + \delta \end{aligned}$$

The lemma then follows. \square

Lemma 4.2. *Let $p_t(G)$ be the probability that an endpoint of edge e is picked in iteration t and \mathcal{T} is the first iteration that an endpoint of edge e is picked (as stated in Lemma 4.1), $\sum_{t=1}^{\mathcal{T}-1} p_t(G) \leq \ln \delta^{-1}$ with probability at least $1 - \delta$.*

Proof. We map our algorithm to the adversarial probabilistic process in Lemma 2.5 and Corollary 2.5.1. In each iteration t , the adversary chooses heads with probability equal to $p_t(G)$. Since $p_t(G)$ is calculated based on the outcomes of $t - 1$ previous iterations, it satisfies the conditions of the adversarial model. In any iteration t , if no head has appeared in the previous $t - 1$ iterations, we have $p_t(G) = \frac{\sum_{j \in \text{adj}(e)} \exp(-\epsilon' \deg_{S_t}^G(j))}{\sum_j \exp(-\epsilon' \deg_{S_t}^G(j))}$, since each node j is picked with probability proportional to $\exp(-\epsilon' \deg_{S_t}^G(j))$. Therefore, the two processes (i.e., our algorithm and the adversarial probabilistic process) are now equivalent. By Corollary 2.5.1, $\sum_{t=1}^{\mathcal{T}-1} p_t(G) \leq \ln \delta^{-1}$ with probability at least $1 - \delta$. The lemma then follows. \square

Theorem 4.3. *Algorithm 1 is (ϵ, δ) -differentially private.*

Proof. By Lemma 4.1, the sequence $\{S_1, S_2, \dots\}$ is $(\epsilon/2, \delta)$ -differentially private. For each S_t , adding or removing 1 edge to/from the graph changes $\rho(S_t)$ by at most 1, hence $\Delta_\rho = 1$. Applying the Exponential Mechanism, the last command releases an output with $\epsilon/2$ -differential privacy. Using the Composition Theorem, Algorithm 1 is (ϵ, δ) -differentially private. \square

Lemma 4.4. *In each iteration t , the node v_t picked by the exponential mechanism satisfies $\deg_{S_t}(v) \leq 2\rho(S_t) + \frac{24}{\epsilon} \ln(e/\delta) \ln(n)$, with probability at least $1 - 1/n^2$.*

Proof. Since $2\rho(S_t) = \frac{2(\#\text{edges in } S_t)}{|S_t|} = \frac{\sum_{v \in S_t} \deg_{S_t}(v)}{|S_t|}$, it follows that $\min_{v \in S_t} \deg_{S_t}(v) \leq 2\rho(S_t)$. Let v_{\min} be a node achieving the minimum degree in S_t . Then $\deg_{S_t}(v_{\min}) \leq 2\rho(S_t)$.

In the exponential mechanism run in iteration t , the sensitivity $\Delta = 1$, as $|\deg_{S_t}^G(v) - \deg_{S_t}^{G'}(v)| \leq 1$. Applying Theorem 2.4 (with $u(S_t, v) = -\deg_{S_t}^G(v)$ and $r = 2 \ln n$), it follows that $\Pr[\deg_{S_t}(v_t) \geq \deg_{S_t}(v_{\min}) + \frac{6}{\epsilon} \ln(n)] \leq e^{-2 \ln n} \leq 1/n^2$. Substituting $\epsilon' = \frac{\epsilon}{4 \ln(e/\delta)}$, the Lemma follows. \square

Theorem 4.5. *Let S^* be an optimal solution. Assuming $n > 3, \delta < 1/e$, the set S output by Algorithm 1 satisfies $\rho(S) \geq \frac{\rho(S^*)}{2} - \frac{32}{\epsilon} \ln(1/\delta) \ln(n)$, with probability at least $1 - 2/n$.*

Proof. Applying a union bound with Lemma 4.4, it follows that for each iteration t , $\deg_{S_t}(v_t) \leq 2\rho(S_t) + \frac{24}{\epsilon} \ln(e/\delta) \ln(n)$, with probability $1 - 1/n$. Let T denote the first iteration that a node $v_T \in S^*$ is picked. As observed in (Bahmani et al., 2012), $\deg_{S^*}(v) \geq \rho(S^*)$, therefore

$$\begin{aligned} \rho(S^*) &\leq \deg_{S^*}(v) \leq \deg_{S_T}(v) \\ &\leq 2\rho(S_T) + \frac{24}{\epsilon} \ln(e/\delta) \ln(n), \end{aligned}$$

with probability at least $1 - 1/n$. Therefore, $\rho(S_T) \geq \rho(S^*)/2 - \frac{12}{\epsilon} \ln(e/\delta) \ln(n)$, with prob. at least $1 - 1/n$.

Next, applying Theorem 2.4 to the exponential mechanism in the last line of the algorithm, we have $\Pr[\rho(S) \leq \rho(S_T) - 4 \frac{\log n + r}{\epsilon}] \leq e^{-r}$. Choosing $r = \ln n$, we have $\rho(S) \geq \rho(S_T) - 8 \ln n / \epsilon$, with probability at least $1 - 1/n$. Combining this with the bound on $\rho(S_T)$, we have $\rho(S) \geq \rho(S^*)/2 - \frac{12}{\epsilon} \ln(e/\delta) \ln(n) - 8 \ln n / \epsilon$.

Finally, we derive a coarse but simpler bound. For $\delta \leq 1/e$, we have $1 \leq \ln(e/\delta) \leq 2 \ln(1/\delta)$. Using these in the lower bound for $\rho(S)$ gives $\rho(S) \geq \frac{\rho(S^*)}{2} - (24 + \frac{8}{\ln(1/\delta)}) \frac{1}{\epsilon} \ln(1/\delta) \ln(n) \geq \frac{\rho(S^*)}{2} - \frac{32}{\epsilon} \ln(1/\delta) \ln(n)$. \square

5. Parallel Private Densest Subgraph

5.1. Algorithm PARDENSEDP

Instead of picking a single node in each iteration, as in Algorithm SEQDENSEDP, the main idea in PARDENSEDP is to independently sample each node with the right probability. As a result, multiple nodes might be selected simultaneously, especially when the degrees are low. This change makes the privacy analysis more challenging because (1) we have to consider different scenarios of picking the endpoints of the extra edge e and (2) without the normalization terms, the probability P in the proof of Lemma 4.1 loses its symmetry.

Theorem 5.1. *Algorithm 2 is (ϵ, δ) -differentially private.*

Algorithm 2 PARDEnSEDP(G, ϵ, δ)

Input: $G \in \mathcal{G}$, privacy parameters ϵ, δ .

Output: A subset $S \subset V(G)$

```

1: Let  $\epsilon' = \frac{1-1/e}{8 \ln(e/\delta)} \epsilon$ , and  $c = 1/\epsilon' + 1$ 
2:  $S_0 = V(G)$ ,  $t = 0$ 
3: while  $|S_t| > 0$  do
4:    $\forall j \in S_t$ , assign a random variable  $p_{tj} \in \{0, 1\}$  such that
      $\Pr[p_{tj} = 1] = P_{tj} = \exp(-\epsilon'(\deg_{S_t}(j) + c))$ 
5:    $t := t + 1$ 
6:    $\pi_t = \{j \in S_t : p_{tj} = 1\}$ 
7:    $S_t = S_{t-1} \setminus \pi_t$ 
8: end while
9: Return  $S_t$  from distinct set  $\{S_t\}$  with probability proportional
   to  $e^{\epsilon \rho(S_t)/2}$ 

```

Proof. (Sketch, full proof in Theorem A.4) Using the same notations in Lemma 4.1, we prove the sequence $\{S_1, S_2, \dots\}$ output by the **while** loop is $(\epsilon/2, \delta)$ -differentially private. The remaining part is the same to Theorem 4.3. We construct $P = \frac{\Pr[\pi^G = \pi]}{\Pr[\pi^{G'} = \pi]}$, and define:

$$A_t = \prod_{j:p_{tj}=1} \frac{P_{tj}^G}{P_{tj}^{G'}}, B_t = \prod_{j:p_{tj}=0} \frac{1 - P_{tj}^G}{1 - P_{tj}^{G'}},$$

$$C = \prod_{j:p_{Tj}=1} \frac{P_{Tj}^G}{P_{Tj}^{G'}}, D = \prod_{j:p_{Tj}=0} \frac{1 - P_{Tj}^G}{1 - P_{Tj}^{G'}},$$

then expand $P = \prod_{t=1}^{T-1} (A_t \times B_t) \times C \times D$.

We consider 2 cases (1) $G + e = G'$ and (2) $G - e = G'$ and 2 subcases in each case: (a) both u, v are picked at step T and (b) exact one of them is picked at T . In Case 1, it is straightforward to prove $P \leq \exp(2\epsilon')$, where Case 2 is more difficult to analyze.

In Case 2, we prove $A_t = 1 \forall t < T$. We reduce C and D , and depends on the subcases, they have slightly different forms. We then reduce P to (2a) $P \leq \prod_{t=1}^{T-1} B_t$ and (2b) $P \leq \prod_{t=1}^{T-1} B_t \times \frac{1 - P_{Tj}^G}{1 - P_{Tj}^{G'}}$. Hence, the remaining thing is to bound $\prod_{t=1}^{T-1} B_t$. Since for all $j \notin \text{adj}(e)$, $\deg_{S_t}^G(j) = \deg_{S_t}^{G'}(j)$, the terms corresponding to such nodes cancel out in B_t , which leads to $\prod_{t=1}^{T-1} B_t \leq \exp(\frac{4\epsilon'}{1-e^{-1}} \sum_{t=1}^{T-1} p_t^{uv}(G))$, in which $p_t^{uv}(G)$ is the probability of picking at least one of u, v in iteration t . Mapping the Algorithm's sampling process to the Adversarial Probabilistic Process in Lemma 2.5, we find the upper bound of $\sum_{t=1}^{T-1} p_t^{uv}(G)$ is $\ln \delta^{-1}$ with probability at least $1 - \delta$. It follows that $P \leq \exp(\frac{4\epsilon' \ln(1/\delta)}{1-1/e}) \leq \exp(\epsilon/2)$ in Case 2a and $P \leq \exp(\frac{4\epsilon' \ln(e/\delta)}{1-1/e}) = \exp(\epsilon/2)$ in Case 2b. Using the same argument as in the last part of Lemma 4.1, the proof follows. \square

Theorem 5.2. (Proof in Theorem A.5) Let S^* be an optimal solution. If $\delta \leq 1/e$, the set S output by Algorithm 2

satisfies $\rho(S) \geq \frac{\rho(S^*)}{2} - \frac{56}{\epsilon} \ln(1/\delta) \ln n$, with probability at least $1 - 2/n$.

Running time of PARDEnSEDP. Each node is removed in an iteration with probability $e^{-\epsilon' \deg_{S_{t-1}}(v)}$. This could be as small as $e^{-\Theta(n)}$, e.g., if S_{t-1} has a dense subgraph in which each node has very high degree. Therefore, the number of iterations could be $e^{\Theta(n)}$ in the worst case (this is borne out of our experiments on some networks).

5.2. Algorithm PHASEDENSEDP: Logarithmic Bounded Runtime Algorithm

Algorithm 2 can execute in parallel but it has exponential runtime. This section presents a modified version of PARDEnSEDP that has logarithmic bounded runtime, named PHASEDENSEDP. We group iterations into phases and aim to remove at least a constant fraction of nodes in each phase (Lemma 5.3). Also, we only update node's new degrees at the end of each phase, allowing the sampling process per each node to run independently as its degree remains unchanged in the current phase. The privacy analysis of PHASEDENSEDP differs from PARDEnSEDP's in two key ways: (1) the terms corresponding to the endpoints of the extra edge e are not updated until the end of the current phase and (2) it requires two instances of the Adversarial Probabilistic Process to mimic the sampling process.

Algorithm 3 PHASEDENSEDP(G, ϵ, δ)

Input: $G \in \mathcal{G}$, privacy parameters ϵ, δ .

Output: A subset $S \subseteq V(G)$

```

1: Let  $\epsilon' = \frac{1-1/e}{24 \ln(4/\delta)} \epsilon$ , and  $c = 1/\epsilon' + 1$ 
2: Let  $\pi_0 = V, S_0 = V, T_0 = 0, i = 1$ 
3: while  $S_i \neq \emptyset$  do
4:   if  $|S_i| \leq \ln n$  then
5:      $T_i = T_{i-1} + 1, \pi_{T_i} = S_i, S_{i+1} = \emptyset, i = i + 1$ 
6:   else
7:     For each  $v \in S_i$ , let  $d(v) = \deg_{S_i}(v)$ 
8:     Let  $P_v = \exp(-\epsilon'(d(v) + c))$ 
9:     Let  $T_v$  be chosen from a geometric random process with
       probability  $P_v$ 
10:    Let  $\widehat{\rho}(S_i) = \rho(S_i) + \frac{16}{\epsilon} \ln n + \text{Lap}(\frac{4 \ln n}{|S_i| \epsilon})$ 
11:    Let  $T_i = \exp(\epsilon'(4\widehat{\rho}(S_i) + c)) \cdot 4 \ln n$ 
12:    For  $t \leq T_i$ , let  $\pi_t = \{v : T_v = t\}$ 
13:     $S_{i+1} = S_i \setminus \bigcup_{t'=T_{i-1}+1}^{T_i} \pi_{t'}$ 
14:     $i = i + 1$ 
15:   end if
16: end while
17: Return  $S \in$  distinct set  $\{S_0, S_1, \dots\}$  with probability pro-
   portional to  $e^{\epsilon \rho(S_i)/2}$ 

```

Lemma 5.3. The number of phases is at most $\log n$, with probability at least $1 - \frac{1}{n^2}$.

Proof. Consider any phase i , except the last one, and let

$v \in S_i$ be a node with $d_{S_i}(v) \leq 4\rho(S_i)$. Since T_v is sampled from a geometric distribution, $\Pr[T_v > 4 \ln n / P_v] \leq (1 - P_v)^{4 \ln n / P_v} \leq 1/n^4$. Therefore, with probability at least $1 - 1/n^3$, for all nodes $v \in S_i$, we have $T_v \leq \exp(\epsilon'(d_{S_i}(v) + c))4 \ln n \leq \exp(\epsilon'(4\rho(S_i) + c))4 \ln n \leq \exp(\epsilon'(4\rho(S_i) + c))4 \ln n = T_i$, where the second inequality is because $d_{S_i}(v) \leq 4\rho(S_i)$ and the last inequality follows Lemma A.6. Let $A(S_i) = \cup_{t'=T_{i-1}+1}^{T_i} \pi_{t'}$ denote the set of nodes removed in this phase.

Therefore, with probability at least $1 - 1/n^3$, every node $v \in S_{i+1} = S_i - A(S_i)$ has $d_{S_i}(v) > 4\rho(S_i)$. We have $2|E(S_i)| = \sum_{v \in A(S_i)} d_{S_i}(v) + \sum_{v \in S_{i+1}} d_{S_i}(v) \geq |S_{i+1}|4\rho(S_i) = 4|S_{i+1}|\frac{|E(S_i)|}{|S_i|}$, which implies $|S_{i+1}| \leq |S_i|/2$. Thus, with probability at least $1 - 1/n^3$, the number of nodes in each phase reduces by factor of 2. Therefore, with probability at least $1 - \frac{\log n}{n^3} \geq 1 - \frac{1}{n^2}$, the Lemma follows. \square

Theorem 5.4. *Algorithm 3 preserves (ϵ, δ) -differential privacy, with $\epsilon \in (0, 1]$, $\delta \in (2n^{-2}, 1)$.*

Proof. (Sketch; full proof in Theorem A.10) Similar to Theorem 5.1, we prove the sequence $\{S_1, S_2, \dots\}$ output by the **while** loop is $(\epsilon/2, \delta)$ -differentially private.

First, prove the sequence $\{\pi_0, \pi_1, \dots\}$ is $(\epsilon/4, \delta/2)$ -differentially private. Using the same notations and expansion, we re-analyze the same cases in Theorem 5.1. Analysis in Case 1a and 2a (where u and v are removed at the same iteration) remains unchanged. We re-evaluate Case 1b and 2b. Assume u is removed first at \mathcal{T}_u and v is removed later at $\mathcal{T} = \mathcal{T}_v > \mathcal{T}_u$.

In Case 1b, we prove that $A_t = 1 \forall t \neq \mathcal{T}_u$ and $A_{\mathcal{T}_u} = \exp(\epsilon')$, C and D are the same as in Theorem 5.1. We split the product of B_t into 2 ranges $B_t = \frac{(1-P_{tu}^G)(1-P_{tv}^G)}{(1-P_{tu}^{G'}) (1-P_{tv}^{G'})}$ for $t \in [1, \mathcal{T}_u - 1]$ and $B_t = \frac{1-P_{tv}^G}{1-P_{tv}^{G'}}$ for $t \in [\mathcal{T}_u, \mathcal{T}_v - 1]$. In either case, $B_t \leq 1$. We have $P = \prod_{t=1}^{\mathcal{T}_v-1} (A_t \times B_t) \times C \times D \leq \exp(2\epsilon')$.

In Case 2b, similar to Theorem 5.1, we prove $A_t \leq 1, C \leq 1, D = 1$, and split the product of B_t into two parts, where the latter arises since we do not update the degree of v after removing u : $\prod_{t=1}^{\mathcal{T}_v-1} B_t = \prod_{t=1}^{\mathcal{T}_u-1} \frac{(1-P_{tu}^G)(1-P_{tv}^G)}{(1-P_{tu}^{G'}) (1-P_{tv}^{G'})} \times \prod_{t=\mathcal{T}_u}^{\mathcal{T}_v-1} \frac{1-P_{tv}^G}{1-P_{tv}^{G'}} \leq \prod_{t=1}^{\mathcal{T}_u-1} \exp(\frac{4\epsilon' p_t(G)}{1-e^{-1}}) \times \prod_{t'=\mathcal{T}_u}^{\mathcal{T}_v-1} \exp(\frac{2\epsilon' q_{v'}(G)}{1-e^{-1}})$. We use two instances of the Adversarial Probabilistic Process to calculate the upper bound on the product of B_t as $\prod_{t=1}^{\mathcal{T}_v-1} B_t \leq \exp(\frac{4\epsilon' \ln(4/\delta)}{1-e^{-1}}) \times \exp(\frac{2\epsilon' \ln(4/\delta)}{1-e^{-1}}) = \exp(\frac{6\epsilon' \ln(4/\delta)}{1-e^{-1}})$, with probability at least $1 - \delta/2$.

Second, we prove the sequence $\{\widehat{\rho}(S_0), \widehat{\rho}(S_1), \dots\}$ is $(\epsilon/4, \delta/2)$ -differentially private, using the Laplace mech-

anism on each $\widehat{\rho}(S_i)$ and composition over $\ln n$ elements of the sequence. Finally, Because $\{S_0, S_1, \dots\}$ is composed from $\{\pi_0, \pi_1, \dots\}$ and the the information computed from $\{\widehat{\rho}(S_0), \widehat{\rho}(S_1), \dots\}$, it is $(\epsilon/2, \delta)$ -differentially private. \square

Theorem 5.5. (Proof in Theorem A.11) *Let S^* be an optimal solution. With probability at least $1 - 2/n$, the set S output by Algorithm 3 satisfies $\rho(S) \geq \frac{\rho(S^*)}{4} - \frac{160}{\epsilon} \ln(1/\delta) \ln(n)$ if $\delta \leq 1/e$.*

6. Map-Reduce Implementation

We briefly describe MRDENSEDP (Algorithm 4)—a Map-Reduce implementation of PHASEDENSEDP. We prove in Lemma A.13 that MRDENSEDP preserves the same privacy guarantee and runtime as of PHASEDENSEDP.

Each iteration i of MRDENSEDP contains 3 Reduce subphases and 1 Map subphase and is equivalent to phase i of PHASEDENSEDP. The first Reduce subphase takes input as pairs of nodes that constitute an edge and group them by endpoints. It creates a list of neighbors of each node to keep track of current nodes and calculate their degrees. The second Reduce subphase groups the current nodes together to form a candidate set S_i while the third one gathers the nodes' degrees to calculate the density S_i . The Map subphase performs the sampling process. It samples the removal time of each node and compares to the cut-off time of the current phase to decide if the node is picked or not. If the node is picked, it emits the node's self-edge to signal the next iteration's first Reduce subphase that the node is removed. Otherwise, it emits the edges adjacent to the node for the next iteration to rebuild the graph.

7. Lower bound

Theorem 7.1. *Any (ϵ, δ) -differentially private algorithm for DENSEDP must incur an additive lower bound of $\Omega(\sqrt{\log n/\epsilon})$ for $\delta \leq 1/n$.*

Proof. Let $G = (V, E)$ denote a graph with $|V| = n$ and $E = \emptyset$. Let $M(G)$ be an (ϵ, δ) -differentially private mechanism. Let V be partitioned into sets V_1, \dots, V_N , for $N = n/a$, $\binom{a}{2} = \frac{1}{3\epsilon} \log n$, with each $|V_i| = a$. For $i \in \{1, \dots, N\}$, define $\mathcal{S}_i = \{S \subseteq V : S \cap V_i \neq \emptyset, |S| \leq a^2\}$. We prove in Lemma A.12 that there exists an $i \in \{1, \dots, N\}$ such that $\Pr[M(G) \in \mathcal{S}_i] \leq \frac{2a^3}{n}$.

Next, let $G' = (V, E')$ be a graph such that V_i is a clique in E' , and there are no additional edges. Then, $|E'| = \binom{a}{2} = \frac{1}{3\epsilon} \log n$. By the group privacy property of (ϵ, δ) -differential privacy (Lemma 2.2 of (Vadhan, 2017)), it follows that $\Pr[M(G') \in \mathcal{S}_i] \leq \Pr[M(G) \in \mathcal{S}_i] \exp(\frac{1}{3} \log n) + \frac{1}{3\epsilon} \log n \exp(\frac{1}{3} \log n) \delta \leq \frac{2a^3}{n} n^{1/3} + \frac{\log n}{3\epsilon} n^{1/3} \frac{1}{n} = \frac{2a^3}{n^{2/3}} + \frac{\log n}{3\epsilon n^{2/3}} \leq \frac{3a^3}{n^{2/3}}$.

For any $S \notin \mathcal{S}_i$, we have $\rho(S) \leq 1/2$, since either (1) $|S| > a^2$, which implies $\rho(S) \leq \frac{\binom{a}{2}}{a^2} \leq \frac{1}{2}$, since S may have up to $\binom{a}{2}$ edges. or (2) $|S| \leq a^2$, $S \cap V_i = \emptyset$, which implies $\rho(S) = 0$. If $M(G') \in \mathcal{S}_i$, we have $\rho(M(G')) \leq a/2$. Hence, $\mathbb{E}[\rho(M(G'))] \leq \frac{a}{2} \Pr[M(G') \in \mathcal{S}_i] + \frac{1}{2} \cdot (1 - \Pr[M(G') \in \mathcal{S}_i]) \leq \frac{3a^4}{2n^{2/3}} + \frac{1}{2} \leq 1$, and the theorem follows since $\rho(G') = \frac{a-1}{2}$.

□

8. Experiments

Our experiments address the following questions.

Accuracy of our proposed methods. How does the accuracy of the solution computed using our private algorithms vary with ϵ and δ in different graphs?

Recall. To what extent can our private algorithms find the densest subgraph, as quantified by the recall?

Efficiency of the parallel algorithms. How does the number of iterations vary, relative to SEQDENSEDP?

8.1. Baselines and measurements

We use the algorithm of (Charikar, 2000) as a baseline to compute a *non-private densest solution* for performance evaluation in our experiments. Let S_b denote the subgraph output by the baseline algorithm, and let $\mathcal{M}(G)$ denote the solution computed by our private algorithm \mathcal{M} on graph G (for fixed ϵ, δ). We consider the following metrics for evaluating our algorithms. For all these metrics, the closer they are to 1, the better our algorithms perform, in comparison to the baseline.

- *Relative density*: this is defined as $\rho(\mathcal{M}(G))/\rho(S_b)$.
- *Jaccard index*: $|\mathcal{M}(G) \cap S_b|/|\mathcal{M}(G) \cup S_b|$, which captures similarity between $\mathcal{M}(G)$ and S_b .
- *Recall*: $|\mathcal{M}(G) \cap S_b|/|S_b|$, which quantifies what fraction of nodes in S_b are selected by $\mathcal{M}(G)$.

Table 1 lists 6 different networks from SNAP database we use to evaluate our results—these are chosen to be of various sizes, in order to understand the impact of network structure on the results (Leskovec & Krevl, 2014). Due to the space limit, we present the setup and experiments on an extended list of 20 networks (Table 2) in Section B.1. Also, we only show results for SEQDENSEDP and PAR-DENSEDP here; see the Appendix for a detailed comparison with PHASEDENSEDP.

8.2. Experimental Results

Utility analysis of private Algorithms. Figure 1 shows the densities of private subgraphs in relative to the densities of the baselines. In most networks which have average degree larger than 4, we observe that SEQDENSEDP has relative

Network	Nodes	Edges	Description
ca-GrQc	5242	14496	Collab. net. Arxiv General Rel.
musae_DE	9498	153138	Social net. of Twitch (DE)
musae_ENGB	7126	35324	Social net. of Twitch (ENGB)
ca-AstroPh	18772	198110	Collab. net. Arxiv Astro Phy.
musae_squirrel	5201	198493	Wiki page-page net. (squirrel)
facebook	4039	88234	Social circles from Facebook

Table 1. Summary of 6 of the 20 networks used for the experiments (Leskovec & Krevl, 2014); the remaining networks are summarized in Table 2 of the Appendix.

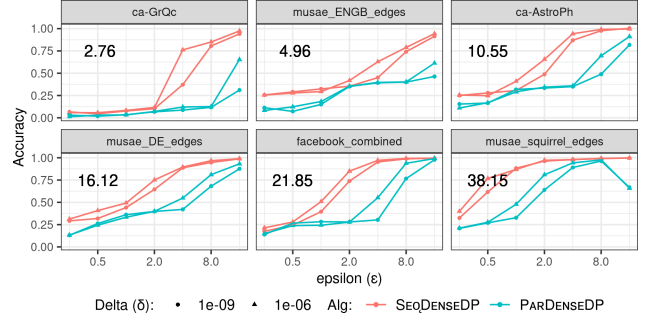


Figure 1. Accuracy in term of relative density of our private algorithms. The number in each graph indicates the density of the graph as whole (which equals the average degree).

density $\geq 75\%$ for $\epsilon = 2$ or 4. For $\epsilon = 2$, SEQDENSEDP has the same density as the non-private baseline in four of six networks and has at least 50% of the density of the baseline in all except one. Figure 5 shows the relationship between the relative density of the solution computed by our private mechanism and the network density (i.e., the average degree) for the twenty networks we study. It confirms that, in general, the accuracy is higher in higher density graphs. In general, we observe the trade-off between accuracy and privacy in all measurements. Higher ϵ , which means less privacy guarantee, yields better accuracy. In contrast, Figures 1, 2, 3 show that δ does not have significant impacts on the quality of solutions. We show in the Appendix that, in general, PAR-DENSEDP outperforms PHASEDENSEDP at $\epsilon \geq 2$. For small ϵ , neither algorithm has good accuracy, though PHASEDENSEDP is slightly better.

Jaccard index and recall. Figure 2 shows the Jaccard similarity between the private subgraphs and S_b . The coefficients vary greatly across networks. Four out of the six networks have Jaccard similarity coefficients at least 0.5 for $\epsilon \geq 2$ with SEQDENSEDP. As with the relative density, networks with higher density tend to have better Jaccard similarity coefficients. Figure 3 shows the recall of our algorithms, i.e., the fraction of nodes in S_b which are selected by our private algorithms. The recall for SEQDENSEDP is at least 75% in all networks for $\epsilon \geq 1$, which indicates that the algorithm successfully finds most nodes in S_b .

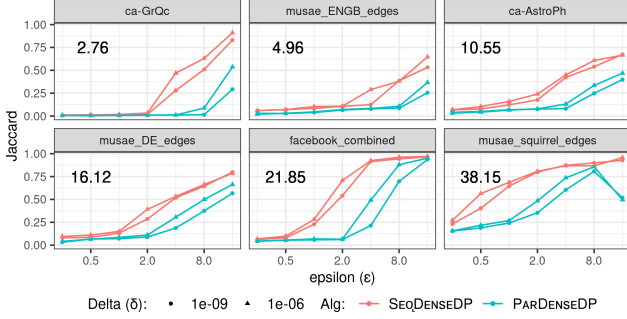


Figure 2. Jaccard similarity coefficient of private subgraphs and non-private baselines.

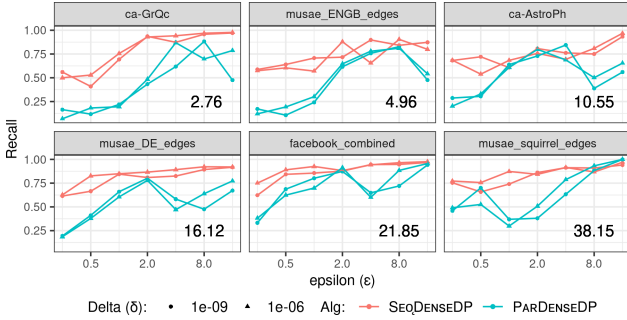


Figure 3. Recall: Fraction of nodes of the baseline's subgraphs are included in our algorithm's outputs.

Performance of parallel algorithms. Figure 4 shows the ratio of the number of iterations taken by PARDENSEDP and SEQDENSEDP (recall that the number of iterations for SEQDENSEDP is n). Due to the nature of the sampling probabilities in PARDENSEDP, the probability of removing a node reduces as ϵ increases. Consequently, PARDENSEDP takes more iterations to remove all nodes with larger ϵ and smaller δ . In all but the *musae_squirrel* network (Figure 4), the number of iterations for PARDENSEDP is about 1% of that for SEQDENSEDP, when $\epsilon \leq 8$. Hence in most cases, PARDENSEDP is much more efficient.

9. Conclusions

In this paper, we design the first sequential and parallel differentially private algorithms for the densest subgraph problem in the edge-privacy model. All of them give $(1/2, O(\log n))$ -approximate solutions, with high probability. In other words, they match the $1/2$ approximation of (Charikar, 2000), with an additive approximation of $O(\log n)$; we also prove a lower bound on the additive term of $\Omega(\sqrt{\log n})$. Our main technical contributions include adaptation of the exponential mechanism to be applied in parallel, and the analysis of privacy and accuracy. Our experiments on 20 networks show that our algorithms have good accuracy in high-density networks and reasonable recall,

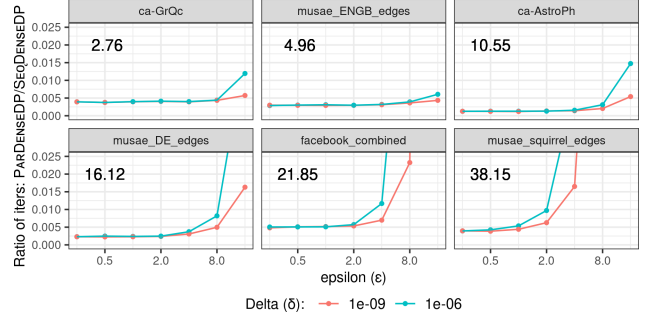


Figure 4. Number of iterations taken by PARDENSEDP, relative to SEQDENSEDP.

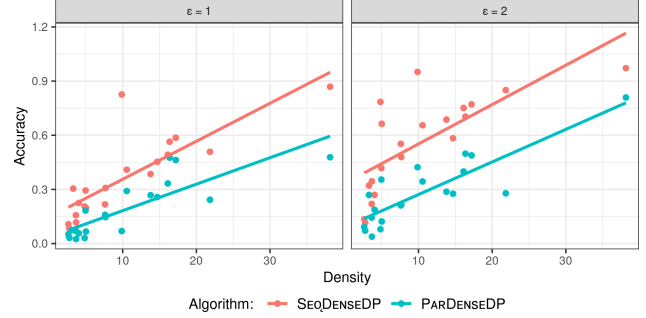


Figure 5. The relationship between accuracy (relative density) and network density. We use $\epsilon \in \{1, 2\}$, $\delta = 10^{-6}$. Each point corresponds to one network. The solid lines show linear models for the relative density and the network density. The plot confirms that our algorithms have better accuracy on higher density networks.

overall, when $\epsilon \geq 2$. SEQDENSEDP has better accuracy while the parallel variants reduce the number of iterations significantly in most settings. Our experiments suggest that PARDENSEDP is good enough for practical uses among two parallel variants. Our paper leads to many open questions. Can we improve the runtime, e.g., by a combination of PARDENSEDP and PHASEDENSEDP? The most intriguing problem is to close the gap between the upper and lower bounds in accuracy. In particular, is an $O(\sqrt{\log n})$ additive approximation to (Charikar, 2000) possible to achieve? Is it possible to obtain a $(1, O(\log n))$ -approximate solution, i.e., a purely additive approximation?

Acknowledgement. We thank anonymous ICML reviewers for helpful suggestions. A. Vullikanti is supported by NSF grants IIS-1931628, CCF-1918656, NSF IIS-1955797, NIH grant R01GM109718.

References

- Abadi, M., Chu, A., Goodfellow, I., McMahan, H. B., Mironov, I., Talwar, K., and Zhang, L. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, CCS '16*, pp. 308–318, New York, NY, USA, 2016. Association for Computing Machinery. ISBN 9781450341394. doi: 10.1145/2976749.2978318. URL <https://doi.org/10.1145/2976749.2978318>.
- Abowd, J. M. The u.s. census bureau adopts differential privacy. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD '18*, pp. 2867, New York, NY, USA, 2018. Association for Computing Machinery. ISBN 9781450355520. doi: 10.1145/3219819.3226070. URL <https://doi.org/10.1145/3219819.3226070>.
- Asahiro, Y., Hassin, R., and Iwama, K. Complexity of finding dense subgraphs. *Discrete Applied Mathematics*, 121(1):15–26, 2002.
- Bahmani, B., Kumar, R., and Vassilvitskii, S. Densest subgraph in streaming and mapreduce. *Proc. VLDB Endow.*, 5(5):454–465, January 2012. ISSN 2150-8097. doi: 10.14778/2140436.2140442. URL <https://doi.org/10.14778/2140436.2140442>.
- Beame, P., Koutris, P., and Suciu, D. Communication steps for parallel query processing. In Hull, R. and Fan, W. (eds.), *Proceedings of the 32nd ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems, PODS 2013, New York, NY, USA - June 22 - 27, 2013*, pp. 273–284. ACM, 2013. doi: 10.1145/2463664.2465224. URL <https://doi.org/10.1145/2463664.2465224>.
- Blocki, J., Blum, A., Datta, A., and Sheffet, O. Differentially private data analysis of social networks via restricted sensitivity. In *Proceedings of the 4th Conference on Innovations in Theoretical Computer Science, ITCS '13*, pp. 87–96, New York, NY, USA, 2013. Association for Computing Machinery. ISBN 9781450318594. doi: 10.1145/2422436.2422449. URL <https://doi.org/10.1145/2422436.2422449>.
- Blum, A., Dwork, C., McSherry, F., and Nissim, K. Practical privacy: The sulq framework. In *Proceedings of the Twenty-Fourth ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems, PODS '05*, pp. 128–138, New York, NY, USA, 2005. Association for Computing Machinery. ISBN 1595930620. doi: 10.1145/1065167.1065184. URL <https://doi.org/10.1145/1065167.1065184>.
- Cadena, J., Vullikanti, A., and Aggarwal, C. On dense subgraphs in signed network streams. In *IEEE International Conference on Data Mining (ICDM)*, 2016.
- Cadena, J., Chen, F., and Vullikanti, A. Graph anomaly detection based on steiner connectivity and density. *Proceedings of the IEEE*, 106(5):829–845, 2018.
- Charikar, M. Greedy approximation algorithms for finding dense components in a graph. volume 1913, pp. 84–95, 09 2000. doi: 10.1007/3-540-44436-X_10.
- Chen, R., Fung, B. C., Philip, S. Y., and Desai, B. C. Correlated network data publication via differential privacy. *The VLDB Journal*, 23(4):653–676, 2014.
- Chen, S. and Zhou, S. Recursive mechanism: towards node differential privacy and unrestricted joins. In *Proceedings of the 2013 ACM SIGMOD International Conference on Management of Data*, pp. 653–664, 2013.
- Day, W.-Y., Li, N., and Lyu, M. Publishing graph degree distribution with node differential privacy. In *Proceedings of the 2016 International Conference on Management of Data*, pp. 123–138, 2016.
- Dean, J. and Ghemawat, S. Mapreduce: simplified data processing on large clusters. *Communications of the ACM*, 51(1):107–113, 2008.
- Ding, X., Zhang, X., Bao, Z., and Jin, H. Privacy-preserving triangle counting in large graphs. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management, CIKM '18*, pp. 1283–1292, New York, NY, USA, 2018. Association for Computing Machinery. ISBN 9781450360142. doi: 10.1145/3269206.3271736. URL <https://doi.org/10.1145/3269206.3271736>.
- Dwork, C. Differential privacy. *Encyclopedia of Cryptography and Security*, pp. 338–340, 2011.
- Dwork, C., Roth, A., et al. The algorithmic foundations of differential privacy. *Foundations and Trends in Theoretical Computer Science*, 9(3-4):211–407, 2014.
- Ghaffari, M., Lattanzi, S., and Mitrović, S. Improved parallel algorithms for density-based network clustering. In Chaudhuri, K. and Salakhutdinov, R. (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 2201–2210, Long Beach, California, USA, 09–15 Jun 2019. PMLR. URL <http://proceedings.mlr.press/v97/ghaffari19a.html>.
- Goldberg, A. V. *Finding a maximum density subgraph*. University of California Berkeley, CA, 1984.

- Gupta, A., Ligett, K., McSherry, F., Roth, A., and Talwar, K. *Differentially Private Combinatorial Optimization*, pp. 1106–1125. doi: 10.1137/1.9781611973075.90. URL <https://epubs.siam.org/doi/abs/10.1137/1.9781611973075.90>.
- Gupta, A., Roth, A., and Ullman, J. Iterative constructions and private data release. In *Theory of cryptography conference*, pp. 339–356. Springer, 2012.
- Hay, M., Li, C., Miklau, G., and 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.
- Hooi, B., Song, H. A., Beutel, A., Shah, N., Shin, K., and Faloutsos, C. Fraudar: Bounding graph fraud in the face of camouflage. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, KDD '16, pp. 895–904, New York, NY, USA, 2016. ACM. ISBN 978-1-4503-4232-2. doi: 10.1145/2939672.2939747. URL <http://doi.acm.org/10.1145/2939672.2939747>.
- Imola, J., Murakami, T., and Chaudhuri, K. Locally differentially private analysis of graph statistics, 2020.
- Karloff, H. J., Suri, S., and Vassilvitskii, S. A model of computation for mapreduce. In Charikar, M. (ed.), *Proceedings of the Twenty-First Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2010, Austin, Texas, USA, January 17-19, 2010*, pp. 938–948. SIAM, 2010. doi: 10.1137/1.9781611973075.76. URL <https://doi.org/10.1137/1.9781611973075.76>.
- Karwa, V., Raskhodnikova, S., Smith, A., and Yaroslavtsev, G. Private analysis of graph structure. *ACM Trans. Database Syst.*, 39(3), October 2014a. ISSN 0362-5915. doi: 10.1145/2611523. URL <https://doi.org/10.1145/2611523>.
- Karwa, V., Slavković, A. B., and Krivitsky, P. Differentially private exponential random graphs. In *International Conference on Privacy in Statistical Databases*, pp. 143–155. Springer, 2014b.
- Kasiviswanathan, S. P., Nissim, K., Raskhodnikova, S., and Smith, A. Analyzing graphs with node differential privacy. In *Proceedings of the 10th Theory of Cryptography Conference on Theory of Cryptography, TCC'13*, pp. 457–476, Berlin, Heidelberg, 2013. Springer-Verlag. ISBN 978-3-642-36593-5. doi: 10.1007/978-3-642-36594-2_26. URL http://dx.doi.org/10.1007/978-3-642-36594-2_26.
- Khuller, S. and Saha, B. On finding dense subgraphs. In Albers, S., Marchetti-Spaccamela, A., Matias, Y., Nikolettseas, S., and Thomas, W. (eds.), *Automata, Languages and Programming*, pp. 597–608, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg.
- Leskovec, J. and Krevl, A. SNAP Datasets: Stanford large network dataset collection. <http://snap.stanford.edu/data>, June 2014.
- McSherry, F. and Talwar, K. Mechanism design via differential privacy. In *Proceedings of the 48th Annual IEEE Symposium on Foundations of Computer Science, FOCS '07*, pp. 94–103, USA, 2007. IEEE Computer Society. ISBN 0769530109. doi: 10.1109/FOCS.2007.41. URL <https://doi.org/10.1109/FOCS.2007.41>.
- Mir, D. and Wright, R. N. A differentially private estimator for the stochastic kronecker graph model. In *Proceedings of the 2012 Joint EDBT/ICDT Workshops*, pp. 167–176, 2012.
- Mitrovic, M., Bun, M., Krause, A., and Karbasi, A. Differentially private submodular maximization: Data summarization in disguise. In Precup, D. and Teh, Y. W. (eds.), *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 2478–2487, International Convention Centre, Sydney, Australia, 06–11 Aug 2017. PMLR. URL <http://proceedings.mlr.press/v70/mitrovic17a.html>.
- Nguyen, D. and Vullikanti, A. Differentially private densest subgraph detection. *CoRR*, abs/2105.13287, 2021. URL <https://arxiv.org/abs/2105.13287>.
- Nguyen, H. H., Imine, A., and Rusinowitch, M. Differentially private publication of social graphs at linear cost. In *2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, pp. 596–599. IEEE, 2015.
- Nguyen, H. H., Imine, A., and Rusinowitch, M. Detecting communities under differential privacy. In *Proceedings of the 2016 ACM on Workshop on Privacy in the Electronic Society, WPES '16*, pp. 83–93, New York, NY, USA, 2016. Association for Computing Machinery. ISBN 9781450345699. doi: 10.1145/2994620.2994624. URL <https://doi.org/10.1145/2994620.2994624>.
- Nissim, K., Raskhodnikova, S., and Smith, A. Smooth sensitivity and sampling in private data analysis. In *Proceedings of the Thirty-Ninth Annual ACM Symposium on Theory of Computing, STOC '07*, pp. 75–84, New York, NY, USA, 2007. Association for Computing Machinery. ISBN 9781595936318. doi: 10.1145/1250790.1250803. URL <https://doi.org/10.1145/1250790.1250803>.

- Rozenshtein, P., Anagnostopoulos, A., Gionis, A., and Tatti, N. Event detection in activity networks. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, 2014.
- Tsourakakis, C., Bonchi, F., Gionis, A., Gullo, F., and Tsiarli, M. Denser than the densest subgraph: extracting optimal quasi-cliques with quality guarantees. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pp. 104–112, 2013.
- Vadhan, S. *The Complexity of Differential Privacy*, pp. 347–450. Springer International Publishing, Cham, 2017. ISBN 978-3-319-57048-8. doi: 10.1007/978-3-319-57048-8_7. URL https://doi.org/10.1007/978-3-319-57048-8_7.
- Xiao, Q., Chen, R., and Tan, K.-L. Differentially private network data release via structural inference. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 911–920, 2014.
- Zhu, T., Li, G., Zhou, W., and Yu, P. S. Differentially private data publishing and analysis: A survey. *IEEE Transactions on Knowledge and Data Engineering*, 29(8): 1619–1638, Aug 2017. ISSN 2326-3865. doi: 10.1109/TKDE.2017.2697856.

A. Missing Proofs

A.1. SEQDENSEDP

Lemma A.1. *The computation of the sequence $\pi^G = \pi_1, \dots, \pi_n$, in the **for** loop in Algorithm 1 (lines 3–5) is $(\epsilon/2, \delta)$ -differentially private.*

Proof. Let $G \sim G'$ be two graphs that differ in exactly one edge $e = (u, v)$. Let π^G and $\pi^{G'}$ denote the permutations computed in the **for** loop in Algorithm 1. Let \mathcal{T} is the first iteration that one of the endpoints of e (i.e., u or v) is removed. Let $\text{adj}(e)$ define the set of nodes adjacent to edge e . Recall that $\deg_{S_t}^G(j)$ is the degree of node j in subgraph S_t of graph G .

Fix a permutation π . Consider the following ratio:

$$\begin{aligned} P &= \frac{\Pr[\pi^G = \pi]}{\Pr[\pi^{G'} = \pi]} \\ &= \prod_{t=1}^n \frac{\exp(-\epsilon' \deg_{S_t}^G(\pi_t)) / \sum_j \exp(-\epsilon' \deg_{S_t}^G(j))}{\exp(-\epsilon' \deg_{S_t}^{G'}(\pi_t)) / \sum_j \exp(-\epsilon' \deg_{S_t}^{G'}(j))} \\ &= \frac{\exp(-\epsilon' \deg_{S_{\mathcal{T}}}^G(\pi_{\mathcal{T}}))}{\exp(-\epsilon' \deg_{S_{\mathcal{T}}}^{G'}(\pi_{\mathcal{T}}))} \prod_{t=1}^{\mathcal{T}} \frac{\sum_j \exp(-\epsilon' \deg_{S_t}^{G'}(j))}{\sum_j \exp(-\epsilon' \deg_{S_t}^G(j))} \end{aligned}$$

The last equality follows from the definition of \mathcal{T} , because: (i) for $t = \mathcal{T} + 1..n$, $\deg_{S_t}^G(j) = \deg_{S_t}^{G'}(j)$ for all nodes j , and, (ii) for $t = 1..\mathcal{T} - 1$, $\deg_{S_t}^G(\pi_t) = \deg_{S_t}^{G'}(\pi_t)$. We have two cases for the analysis.

First, suppose, $G + e = G'$. In this case, we have $\frac{\exp(-\epsilon' \deg_{S_{\mathcal{T}}}^G(\pi_{\mathcal{T}}))}{\exp(-\epsilon' \deg_{S_{\mathcal{T}}}^{G'}(\pi_{\mathcal{T}}))} = \exp(\epsilon')$ because $\deg_{S_{\mathcal{T}}}^G(\pi_{\mathcal{T}}) + 1 = \deg_{S_{\mathcal{T}}}^{G'}(\pi_{\mathcal{T}})$.

Further, for each t, j , we have $\deg_{S_t}^{G'}(j) \geq \deg_{S_t}^G(j)$ for all j , which implies the product $\prod_{t=1}^{\mathcal{T}} \frac{\sum_j \exp(-\epsilon' \deg_{S_t}^{G'}(j))}{\sum_j \exp(-\epsilon' \deg_{S_t}^G(j))} \leq 1$. Therefore, $P \leq \exp(\epsilon')$

In the second case, $G - e = G'$. Similar to the first case, $\frac{\exp(-\epsilon' \deg_{S_{\mathcal{T}}}^G(\pi_{\mathcal{T}}))}{\exp(-\epsilon' \deg_{S_{\mathcal{T}}}^{G'}(\pi_{\mathcal{T}}))} = \exp(-\epsilon') \leq 1$. For $j \in \text{adj}(e), t \leq \mathcal{T}$, we have $\deg_{S_t}^G(j) = \deg_{S_t}^{G'}(j) + 1$. For $j \notin \text{adj}(e)$, and for all t , $\deg_{S_t}^G(j) = \deg_{S_t}^{G'}(j)$. Further, for $t > \mathcal{T}$, we have $\deg_{S_t}^G(j) = \deg_{S_t}^{G'}(j)$ for all j . Note that whenever we use $\deg_{S_t}^G(j)$, we only consider $j \in S_{t-1}$, i.e., j would not have been deleted from the graph before step t .

Therefore, for $t \leq \mathcal{T}$:

$$\begin{aligned} &\sum_j \exp(-\epsilon' \deg_{S_t}^{G'}(j)) \\ &= \sum_{j \in \text{adj}(e)} \exp(-\epsilon' \deg_{S_t}^{G'}(j)) + \sum_{j \notin \text{adj}(e)} \exp(-\epsilon' \deg_{S_t}^{G'}(j)) \\ &= \sum_{j \in \text{adj}(e)} \exp(-\epsilon' (\deg_{S_t}^G(j) - 1)) + \sum_{j \notin \text{adj}(e)} \exp(-\epsilon' \deg_{S_t}^G(j)) \\ &= \exp(\epsilon') \sum_{j \in \text{adj}(e)} \exp(-\epsilon' \deg_{S_t}^G(j)) + \sum_{j \notin \text{adj}(e)} \exp(-\epsilon' \deg_{S_t}^G(j)) \\ &= (\exp(\epsilon') - 1) \sum_{j \in \text{adj}(e)} \exp(-\epsilon' \deg_{S_t}^G(j)) + (\sum_{j \in \text{adj}(e)} \exp(-\epsilon' \deg_{S_t}^G(j)) + \sum_{j \notin \text{adj}(e)} \exp(-\epsilon' \deg_{S_t}^G(j))) \\ &= (\exp(\epsilon') - 1) \sum_{j \in \text{adj}(e)} \exp(-\epsilon' \deg_{S_t}^G(j)) + \sum_j \exp(-\epsilon' \deg_{S_t}^G(j)) \end{aligned}$$

We now return to the expression for P . Since $\deg_{S_t}^G(j) = \deg_{S_t}^{G'}(j)$ for all j and $t > \mathcal{T}$, we have

$$\begin{aligned}
 P &\leq \prod_{t=1}^{\mathcal{T}} \left(\frac{(\exp(\epsilon') - 1) \sum_{j \in \text{adj}(e)} \exp(-\epsilon' \deg_{S_t}^G(j))}{\sum_j \exp(-\epsilon' \deg_{S_t}^G(j))} \right. \\
 &\quad \left. + \frac{\sum_j \exp(-\epsilon' \deg_{S_t}^G(j))}{\sum_j \exp(-\epsilon' \deg_{S_t}^G(j))} \right) \\
 &= \prod_{t=1}^{\mathcal{T}} \left(1 + (\exp(\epsilon') - 1) \frac{\sum_{j \in \text{adj}(e)} \exp(-\epsilon' \deg_{S_t}^G(j))}{\sum_j \exp(-\epsilon' \deg_{S_t}^G(j))} \right) \\
 &= \prod_{t=1}^{\mathcal{T}} (1 + (\exp(\epsilon') - 1) p_t(G)) \\
 &\leq \prod_{t=1}^{\mathcal{T}} \exp((\exp(\epsilon') - 1) p_t(G)), \\
 &\quad \text{using the inequality } 1 + x \leq e^x \text{ for all } x \geq 0,
 \end{aligned}$$

where $p_t(G)$ is the probability that an endpoint of edge e is picked in iteration t . We will show in Lemma 4.2, by using the adversarial process (Section 2.2), that $\sum_{t=1}^{\mathcal{T}-1} p_t(G) \leq \ln \delta^{-1}$ with probability at least $1 - \delta$. Assuming this bound, we first derive a bound on P . We say that π is $\ln \delta^{-1}$ -good if $\sum_{t=1}^{\mathcal{T}-1} p_t(G) \leq \ln \delta^{-1}$ and $\ln \delta^{-1}$ -bad otherwise. Therefore, with probability at least $1 - \delta$, the ordering π is $\ln \delta^{-1}$ -good and

$$\begin{aligned}
 P &\leq \prod_{t=1}^{\mathcal{T}} \exp((\exp(\epsilon') - 1) p_t(G)) \\
 &\leq \exp(2\epsilon' \sum_{t=1}^{\mathcal{T}} p_t(G)), \\
 &\quad \text{using the inequality } e^x \leq 1 + 2x \text{ for } x \leq 1 \\
 &\leq \exp(2\epsilon' (\ln \delta^{-1} + p_{\mathcal{T}}(G))) \\
 &\leq \exp(2\epsilon' (\ln \delta^{-1} + 1)) \\
 &= \exp(2\epsilon' (\ln(e/\delta))) \\
 &= \exp(\epsilon/2), \text{ since } \epsilon' = \frac{\epsilon}{4 \ln(e/\delta)},
 \end{aligned}$$

which implies $\Pr[\pi^G = \pi] \leq \exp(\epsilon/2) \Pr[\pi^{G'} = \pi]$ in this case.

Let \mathcal{P} be the set of output orderings that might be generated by the **for** loop in SEQDENSEDP. Then, we have

$$\begin{aligned}
 &\Pr[\pi^G \in \mathcal{P}] \\
 &= \sum_{\pi \in \mathcal{P}} \Pr[\pi^G = \pi] \\
 &= \sum_{\pi \in \mathcal{P}: \pi \text{ is } (\ln \delta^{-1})\text{-good}} \Pr[\pi^G = \pi] + \sum_{\pi \in \mathcal{P}: \pi \text{ is } (\ln \delta^{-1})\text{-bad}} \Pr[\pi^G = \pi] \\
 &\leq \sum_{\pi \in \mathcal{P}: \pi \text{ is } (\ln \delta^{-1})\text{-good}} \exp(\epsilon/2) \Pr[\pi^{G'} = \pi] + \delta \\
 &\leq \exp(\epsilon/2) \Pr[\pi^{G'} \in \mathcal{P}] + \delta
 \end{aligned}$$

The lemma then follows. \square

A.2. PARDENSEDP

Lemma A.2. Let $G = G' + e$, where $e = (u, v)$. Let $E_{tj} = \frac{1 - P_{tj}^G}{1 - P_{tj}^{G'}}$ be the ratio of the probability that node j is not picked in iteration t in G , to the corresponding probability in G' . Let $p_t^{uv}(G)$ be the probability that at least one of the nodes u or v is removed in iteration t from graph G . Then, for $j \in \text{adj}(e)$, $E_{tj} \leq \exp(\frac{e^{\epsilon'} - 1}{1 - e^{-1}} P_{tj}^G)$, and $E_{tu}E_{tv} \leq \exp(\frac{4\epsilon'}{1 - e^{-1}} p_t^{uv}(G))$.

Proof. For $j \in \text{adj}(e)$, we have $\deg_{S_t}^G(j) = \deg_{S_t}^{G'}(j) + 1$. Therefore,

$$\begin{aligned} E_{tj} &= \frac{1 - P_{tj}^G}{1 - \exp(\epsilon') P_{tj}^G} \\ &= \frac{1 - P_{tj}^G (\exp(\epsilon') + 1 - \exp(\epsilon'))}{1 - \exp(\epsilon') P_{tj}^G} \\ &= \frac{1 - P_{tj}^G \exp(\epsilon') - (1 - \exp(\epsilon')) P_{tj}^G}{1 - \exp(\epsilon') P_{tj}^G} \\ &= 1 + \frac{(\exp(\epsilon') - 1) P_{tj}^G}{1 - \exp(\epsilon') P_{tj}^G} \\ &\leq 1 + \frac{(\exp(\epsilon') - 1) P_{tj}^G}{1 - e^{-1}} \\ &\leq \exp\left(\frac{\exp(\epsilon') - 1}{1 - e^{-1}} P_{tj}^G\right), \end{aligned}$$

where the second last inequality follows from the following argument

$$\begin{aligned} \deg_{S_t}(j) + c &\geq \frac{1}{\epsilon'} + 1 \\ \Rightarrow \epsilon'(\deg_{S_t}(j) + c) - \epsilon' &\geq 1 \\ -\epsilon'(\deg_{S_t}(j) + c) + \epsilon' &\leq -1 \\ P_{tj}^G e^{\epsilon'} = \exp(-\epsilon'(\deg_{S_t}(j) + c)) e^{\epsilon'} &\leq e^{-1} \\ \Rightarrow 1 - P_{tj}^G e^{\epsilon'} &\geq 1 - e^{-1} \\ \Rightarrow \frac{1}{1 - P_{tj}^G e^{\epsilon'}} &\leq \frac{1}{1 - e^{-1}}, \end{aligned}$$

where the last inequality requires $P_{tj}^G e^{\epsilon'} - 1 < 1 - e^{-1}$. This holds because $P_{tj}^G \leq \exp(-\epsilon' - 1) \Rightarrow P_{tj}^G e^{\epsilon'} - 1 \leq e^{-1} - 1 < 1 - e^{-1}$. This implies

$$\begin{aligned} E_{tu}E_{tv} &\leq \exp\left(\frac{\exp(\epsilon') - 1}{1 - e^{-1}} (P_{tu}^G + P_{tv}^G)\right) \\ &\leq \exp\left(\frac{\exp(\epsilon') - 1}{1 - e^{-1}} 2p_t(G)\right) \\ &\leq \exp\left(\frac{4\epsilon'}{1 - e^{-1}} p_t(G)\right), \end{aligned}$$

where the second inequality follows because $\max(P_{tu}^G, P_{tv}^G) \leq p_t(G)$, where $p_t(G)$ is the probability that at least one of u or v is removed at step t , i.e., $p_t(G) = P_{tu} + P_{tv} - P_{tu}P_{tv}$. The last inequality follows because $e^{\epsilon'} < 1 + 2\epsilon'$, if $\epsilon' \leq 1$. \square

Lemma A.3. The computation of the sequence S_1, S_2, \dots by the **while** loop (line 4-8) of PARDENSEDP is $(\epsilon/2, \delta)$ -differentially private.

Proof. Let $G \sim G'$ be two neighboring graphs which differ in a single edge $e = (u, v)$. Let π^G and $\pi^{G'}$ denote the outputs in the case of graphs G and G' , respectively. As in the proof of Lemma 4.1, let \mathcal{T} be the first iteration in which a node in u, v is removed, and let P_{tj}^G be the probability that node j of graph G is sampled at step t . We note that for every node $j \neq u, v$, in every iteration t : $\deg_{S_t}^G(j) = \deg_{S_t}^{G'}(j)$ and hence $P_{tj}^G = P_{tj}^{G'}$. Fix a permutation π , and consider the following ratio:

$$\begin{aligned} P &= \frac{\Pr[\pi^G = \pi]}{\Pr[\pi^{G'} = \pi]} \\ &= \prod_{t=1}^{\mathcal{T}} \frac{\Pr[\pi_t^G = \pi]}{\Pr[\pi_t^{G'} = \pi]}, \text{ since } \deg_{S_t}^G(j) = \deg_{S_t}^{G'}(j) \text{ for } t > \mathcal{T}, \text{ for all } j \\ &= \prod_{t=1}^{\mathcal{T}} \frac{\prod_{j:p_{tj}=1} P_{tj}^G \times \prod_{j:p_{tj}=0} (1 - P_{tj}^G)}{\prod_{j:p_{tj}=1} P_{tj}^{G'} \times \prod_{j:p_{tj}=0} (1 - P_{tj}^{G'})} \\ &= \prod_{t=1}^{\mathcal{T}-1} \left(\prod_{j:p_{tj}=1} \frac{P_{tj}^G}{P_{tj}^{G'}} \times \prod_{j:p_{tj}=0} \frac{1 - P_{tj}^G}{1 - P_{tj}^{G'}} \right) \times \prod_{j:p_{\mathcal{T}j}=1} \frac{P_{\mathcal{T}j}^G}{P_{\mathcal{T}j}^{G'}} \times \prod_{j:p_{\mathcal{T}j}=0} \frac{1 - P_{\mathcal{T}j}^G}{1 - P_{\mathcal{T}j}^{G'}} \end{aligned}$$

$$\text{Define } A_t = \prod_{j:p_{tj}=1} \frac{P_{tj}^G}{P_{tj}^{G'}}, B_t = \prod_{j:p_{tj}=0} \frac{1 - P_{tj}^G}{1 - P_{tj}^{G'}},$$

$$C = \prod_{j:p_{\mathcal{T}j}=1} \frac{P_{\mathcal{T}j}^G}{P_{\mathcal{T}j}^{G'}}, D = \prod_{j:p_{\mathcal{T}j}=0} \frac{1 - P_{\mathcal{T}j}^G}{1 - P_{\mathcal{T}j}^{G'}}$$

We have two cases: $G + e = G'$ and $G - e = G'$.

Case 1 $G + e = G'$. For all t , $A_t = 1$ because for all nodes j and $t < \mathcal{T}$, $\deg_{S_t}^G(j) = \deg_{S_t}^{G'}(j)$. Next,

$$B_t = \prod_{j:p_{tj}=0} \frac{1 - P_{tj}^G}{1 - P_{tj}^{G'}} = \frac{(1 - P_{tu}^G)(1 - P_{tv}^G)}{(1 - P_{tu}^{G'})(1 - P_{tv}^{G'})} \leq 1,$$

because $P_{tj}^{G'} = \exp(-\epsilon') P_{tj}^G \leq P_{tj}^G$ for $j \in \{u, v\}$ and $P_{tj}^{G'} = P_{tj}^G$ for $j \notin \{u, v\}$. Therefore, $\prod_{t=1}^{\mathcal{T}-1} (A_t \times B_t) \leq 1$. We have two sub-cases.

Case 1.a Both u, v are picked at step \mathcal{T} : $p_{\mathcal{T}u} = p_{\mathcal{T}v} = 1$. In that case, none of the terms in D involve u or v , which means $D = 1$. We have

$$\begin{aligned} C &= \prod_{j:p_{\mathcal{T}j}=1} \frac{P_{\mathcal{T}j}^G}{P_{\mathcal{T}j}^{G'}} = \frac{P_{\mathcal{T}u}^G \times P_{\mathcal{T}v}^G}{P_{\mathcal{T}u}^{G'} \times P_{\mathcal{T}v}^{G'}} \\ &= \frac{P_{\mathcal{T}u}^G \times P_{\mathcal{T}v}^G}{\exp(-2\epsilon') \times P_{\mathcal{T}u}^G \times P_{\mathcal{T}v}^G} = \exp(2\epsilon'), \end{aligned}$$

since $\deg_{S_{\mathcal{T}}}^{G'}(j) = \deg_{S_{\mathcal{T}}}^G(j) + 1$, for $j \in \text{adj}(e)$. Therefore, $P = \prod_{t=1}^{\mathcal{T}-1} (A_t \times B_t) \times C \times D \leq \exp(2\epsilon')$.

Case 1.b Exact one node in $\{u, v\}$ is picked at step \mathcal{T} . Assume v is picked, so that $p_{\mathcal{T}u} = 0, p_{\mathcal{T}v} = 1$, and $C = \exp(\epsilon')$ and $D = \frac{1 - P_{\mathcal{T}u}^G}{1 - P_{\mathcal{T}u}^{G'}} \leq 1$. Therefore, $P \leq \exp(\epsilon')$ in this case as well, which implies that in **Case 1**, $P \leq \exp(2\epsilon') < \exp(\epsilon/2)$.

Case 2. $G = G' + e$: by the same argument as in Case 1, we have $A_t = 1$ for all $t < \mathcal{T}$. We have similar two sub-cases.

Case 2.a Both u, v are picked at step \mathcal{T} : then $p_{\mathcal{T}u} = p_{\mathcal{T}v} = 1$, and $C = \exp(-2\epsilon') \leq 1$ and $D = 1$. This implies $P \leq \prod_{t=1}^{\mathcal{T}-1} B_t$

Case 2.b Only one of u, v is picked at step \mathcal{T} . Assume v is picked, so that $p_{\mathcal{T}u} = 0, p_{\mathcal{T}v} = 1$. Then $C = \exp(-\epsilon') \leq 1$, $D = \frac{1-P_{\mathcal{T}u}^G}{1-P_{\mathcal{T}u}^{G'}}$, and $P \leq \prod_{t=1}^{\mathcal{T}-1} B_t \times \frac{1-P_{\mathcal{T}u}^G}{1-P_{\mathcal{T}u}^{G'}}$

Now, we consider the product B_t . Since for all $j \notin \text{adj}(e)$, $\deg_{S_t}^G(j) = \deg_{S_t}^{G'}(j)$, the terms corresponding to such nodes cancel out in B_t , and we have

$$\begin{aligned} \prod_{t=1}^{\mathcal{T}-1} B_t &= \prod_{t=1}^{\mathcal{T}-1} \frac{(1-P_{tu}^G)(1-P_{tv}^G)}{(1-P_{tu}^{G'})(1-P_{tv}^{G'})} \\ &= \prod_{t=1}^{\mathcal{T}-1} (E_{tu} \times E_{tv}) \\ &\leq \prod_{t=1}^{\mathcal{T}-1} \exp\left(\frac{4\epsilon'}{1-e^{-1}} p_t^{uv}(G)\right) \text{ from Lemma A.2} \\ &= \exp\left(\frac{4\epsilon'}{1-e^{-1}} \sum_{t=1}^{\mathcal{T}-1} p_t^{uv}(G)\right). \end{aligned}$$

As in the proof of Lemma 4.1, we will prove later that $\sum_{t=1}^{\mathcal{T}-1} p_t(G) \leq \ln(1/\delta)$. with probability at least $1 - \delta$. Assuming this bound, we complete the rest of proof. With probability at least $1 - \delta$, we have $\prod_{t=1}^{\mathcal{T}-1} B_t \leq \exp\left(\frac{4\epsilon'}{1-e^{-1}} \ln(1/\delta)\right)$. We now apply this inequality to both cases 2.a and 2.b.

Case 2.a $P = \prod_{t=1}^{\mathcal{T}-1} B_t \leq \exp\left(\frac{4\epsilon'}{1-1/e} \ln(1/\delta)\right) \leq \exp(\epsilon/2)$.

Case 2.b $P = \prod_{t=1}^{\mathcal{T}-1} B_t \times D$

$$\begin{aligned} P &= \prod_{t=1}^{\mathcal{T}-1} B_t \times \frac{1-P_{\mathcal{T}u}^G}{1-P_{\mathcal{T}u}^{G'}} \\ &\leq \exp\left(\frac{4\epsilon'}{1-1/e} \ln(1/\delta)\right) \times E_{\mathcal{T}u} \\ &\leq \exp\left(\frac{4\epsilon'}{1-1/e} \ln(1/\delta)\right) \times \exp\left(\frac{4\epsilon'}{1-1/e} p_{\mathcal{T}}(G)\right) \\ &\leq \exp\left(\frac{4\epsilon'}{1-1/e} \ln(1/\delta)\right) \times \exp\left(\frac{4\epsilon'}{1-1/e}\right) \\ &= \exp\left(\frac{4\epsilon'}{1-1/e} (\ln(1/\delta) + 1)\right) \\ &= \exp\left(\frac{4\epsilon'}{1-1/e} \ln(e/\delta)\right) \\ &= \exp(\epsilon/2), \end{aligned}$$

where the second inequality follows because of Lemma A.2.

Therefore, in **Case 2**, we have $P \leq \exp(\epsilon/2)$, with probability at least $1 - \delta$. Using the same argument as in the proof of Lemma 4.1, we have that for any set of outputs \mathcal{P} that can be generated by the **while** loop of PARDENSEDP, $\Pr[\pi^G \in \mathcal{P}] \leq \exp(\epsilon/2) \Pr[\pi^{G'} \in \mathcal{P}] + \delta$, and hence the lemma follows.

Finally, we prove the upper bound of the sum $\sum_{t=1}^{\mathcal{T}-1} p_t^{uv}(G)$ is $\ln(1/\delta)$, with probability at least $1 - \delta$, by mapping our algorithm to the adversarial probabilistic process in Lemma 2.5. The adversarial process is defined as following: in each iteration t , the adversary chooses heads with probability equal to $p_t^{uv}(G)$. If a head is tossed, each node $j \in \{u, v\}$ is picked independently with probability P_{tj} . If a tail is tossed, the adversary does not pick any node of $\{u, v\}$. Every other node

$j \notin \{u, v\}$ is picked independently with probability P_{tj} , no matter how the coin is tossed. Therefore, in case of heads, the outcomes are $\{(u), (v), (u, v)\}$ (which means picking only u , picking only v and picking both u and v), with probabilities proportional to $P_{tu} - P_{tu}P_{tv}$, $P_{tv} - P_{tu}P_{tv}$ and $P_{tu}P_{tv}$, respectively. In order to prove the adversarial process and our algorithm are equivalent, we have to prove that probabilities of picking a node j are the same in both processes.

For $j \notin \{u, v\}$, the adversarial picks j with probability P_{tj} , so the probability of picking j is the same as in Algorithm 2.

We consider the probability of picking u (v is similar) at iteration t is $\Pr[u \text{ is picked}]$:

$$\begin{aligned} &= \Pr[\text{head}](\Pr[(u)|\text{head}] + \Pr[(u, v)|\text{head}]) \\ &= p_t(G) \frac{(P_{tu} - P_{tu}P_{tv}) + (P_{tu}P_{tv})}{P_{tu} - P_{tu}P_{tv} + P_{tv} - P_{tu}P_{tv} + P_{tu}P_{tv}} \\ &= (P_{tu} + P_{tv} - P_{tu}P_{tv}) \frac{P_{tu}}{P_{tu} + P_{tv} - P_{tu}P_{tv}} \\ &= P_{tu}, \end{aligned}$$

which is the same with probability of picking u in Algorithm 2

Similarly, we also have the probability that v picked at iteration t is P_{tv} . Therefore, the two processes (our algorithm and the adversarial probabilistic process) are equivalent. By Corollary 2.5.1 that $\sum_{t=1}^{T-1} p_t(G) \leq \ln(1/\delta)$ with probability at least $1 - \delta$, we have:

$$\prod_{t=1}^{T-1} B_t \leq \exp\left(\frac{4\epsilon'}{1 - e^{-1}} \ln(1/\delta)\right),$$

□

Theorem A.4. *Algorithm 2 is (ϵ, δ) -differentially private.*

Proof. By Lemma A.3, the sequence $\{S_1, S_2, \dots\}$ output by the **while** loop is $(\epsilon/2, \delta)$ -differentially private. For each S_t , adding or removing 1 edge to the graph changes $\rho(S_t)$ by at most 1, hence $\Delta_\rho = 1$. Applying the Exponential Mechanism, the last step of PARDENSEDP releases output S with $\epsilon/2$ -differential privacy. Using the Composition Theorem, Algorithm 2 preserves (ϵ, δ) -differential privacy. □

Theorem A.5. (Theorem 5.2) *Let S^* be an optimal solution. If $\delta \leq 1/e$, the set S output by Algorithm 2 satisfies $\rho(S) \geq \frac{\rho(S^*)}{2} - \frac{56}{\epsilon} \ln(1/\delta) \ln n$, with probability at least $1 - 2/n$.*

Proof. In each iteration t , observe that if a node j has $\deg_{S_t}(j) \geq 2\rho(S_t) + \frac{3}{\epsilon} \ln n \geq \frac{3}{\epsilon} \ln n$, then $\Pr[p_{tj} = 1] \leq \exp(-\epsilon'(\deg_{S_t}(j) + c)) \leq \exp(-\epsilon' \deg_{S_t}(j)) \leq 1/n^3$. Therefore, with probability at least $1 - 1/n$, for all t and j , $\deg_{S_t}(j) \leq 2\rho(S_t) + \frac{3}{\epsilon} \ln n$. Suppose T is the first iteration in which a node $v \in S^* \cap S_T$ is picked. Then, As in the proof of Theorem 4.5, we have $\rho(S^*) \leq 2\rho(S_T) + \frac{3}{\epsilon} \ln n$, which implies $\rho(S_T) \geq \rho(S^*)/2 - \frac{3}{2\epsilon} \ln n$.

Next, in the last line of the algorithm, suppose $S_\tau = \arg\max_{S_t} \rho(S_t)$. Selecting the distinct set of S_t is a post-processing of the private list of all S_t and hence, the set is private and has at most n elements. As in the proof of Theorem 4.5, we have $\rho(S) \geq \rho(S_\tau) - \frac{8}{\epsilon} \ln n \geq \rho(S^*)/2 - \frac{12 \ln(e/\delta)}{(1-1/e)\epsilon} \ln n - \frac{8}{\epsilon} \ln n \geq \rho(S^*)/2 - \left(\frac{24}{1-1/e} + \frac{8}{\ln(1/\delta)}\right) \frac{1}{\epsilon} \ln(1/\delta) \ln(n) \geq \rho(S^*)/2 - \left(\frac{24}{1-1/2} + 8\right) \frac{1}{\epsilon} \ln(1/\delta) \ln(n) \geq \rho(S^*)/2 - \frac{56}{\epsilon} \ln(1/\delta) \ln(n)$, with probability at least $1 - 2/n$. □

A.3. PHASEDENSEDP

Lemma A.6. *For each phase i , except the last, with probability at least $1 - 1/n^4$, $\rho(S_i) \leq \widehat{\rho(S_i)} \leq \rho(S_i) + \frac{32}{\epsilon} \ln n$*

Proof. By the property of the Laplace distribution (Dwork et al., 2014), $\Pr\left[\left|Lap\left(\frac{4 \ln n}{|S_i| \epsilon}\right)\right| > \frac{16 \ln^2 n}{|S_i| \epsilon}\right] \leq \exp(-4 \ln n) = 1/n^4$. Therefore, with probability at least $1 - 1/n^4$, we have $|Lap(\frac{4 \ln n}{|S_i| \epsilon})| \leq \frac{16 \ln^2 n}{|S_i| \epsilon} \leq \frac{16 \ln n}{\epsilon}$, since $|S_i| > \ln n$ for all i , except the last phase. This implies that with probability at least $1 - 1/n^4$, we have $-\frac{16 \ln n}{\epsilon} \leq Lap(\frac{4 \ln n}{|S_i| \epsilon}) \leq \frac{16 \ln n}{\epsilon}$. By the definition of $\widehat{\rho(S_i)}$, the Lemma follows. □

Lemma A.7. *With probability at least $1 - \frac{4 \ln n}{n^2}$, in every phase i (except the last one), for any node v with $d_{S_i}(v) > 4\rho(\widehat{S_i}) + \frac{4}{\epsilon'} \ln n$, we have $T_v > T_i$.*

Proof. Consider any phase i , and node $v \in S_i$. We have $P_v = \exp(-\epsilon'(d_{S_i}(v) + c)) \leq \exp(-\epsilon'(4\rho(\widehat{S_i}) + c) - 4 \ln n) = \exp(-\epsilon'(4\rho(\widehat{S_i}) + c))/n^4$, in which the inequality is because $d_{S_i}(v) > 4\rho(\widehat{S_i}) + \frac{4}{\epsilon'} \ln n$. Since T_v is sampled from a geometric process, we have $\Pr[T_v \leq T_i] = \sum_{j=1}^{T_i} (1 - P_v)^{j-1} P_v \leq T_i P_v \leq \frac{4 \ln n}{n^4}$. By taking a union bound over all nodes, and all phases, the lemma follows. \square

Lemma A.8. *In Algorithm 3, let $p_t(G)$ denote the probability that one of u, v is removed at time t . Let $q_t^u(G)$ denote the probability that node u is removed at time t , similarly $q_t^v(G)$. Let $\mathcal{T}_u, \mathcal{T}_v$ are iterations in which u and v are removed respectively. Assume $\mathcal{T}_u < \mathcal{T}_v$, with probability at least $1 - \delta/2$, $\sum_{t=1}^{\mathcal{T}_u-1} p_t(G) \leq \ln 4/\delta$ and $\sum_{t=\mathcal{T}_u}^{\mathcal{T}_v-1} q_t^v(G) \leq \ln 4/\delta$.*

Proof. We use the same notations as in Lemma A.3. Let $\mathcal{T}_u \leq \mathcal{T}_v$ are iterations when each node in $\{u, v\}$ is removed. Let $\delta' = \delta/4$.

For iteration $t \in [1, \mathcal{T}_u]$, set $p_t(G) = P_{tu} + P_{tv} - P_{tu}P_{tv}$ as the same as in the random process of Lemma A.3. Therefore, the algorithm from iteration $1 \rightarrow \mathcal{T}_1$ is equivalent to 1 instance of the Adversarial Probabilistic Process. Applying Corollary 2.5.1, with probability at least $1 - \delta'$, $\sum_{t=1}^{\mathcal{T}_u-1} p_t(G) \leq \ln 1/\delta' = \ln 4/\delta$.

In the second process after u is removed ($t \in [\mathcal{T}_u + 1, \mathcal{T}_v]$), for all node $j \neq v$, the probabilities of picking j are set to be independent from the coin tosses, hence their probabilities are the same. At each iteration $t \in [\mathcal{T}_u + 1, \mathcal{T}_v]$, set $t' = t - \mathcal{T}_u$, the adversary chooses head probability $q_{t'}^v(G) = P_{tv}$ and picks v if a head is tossed and does not pick v if a tail is tossed. Then, the probability to pick v at step t is still P_{tv} and the sampling process is equivalent to another instance of the Adversarial Probabilistic Process. Applying Corollary 2.5.1, with probability at least $1 - \delta'$, $\sum_{t=\mathcal{T}_u}^{\mathcal{T}_v-1} q_t^v(G) = \sum_{t'=1}^{\mathcal{T}_v-\mathcal{T}_u-1} q_{t'}^v(G) \leq \ln 1/\delta' = \ln 4/\delta$.

Substituting $\delta = 4\delta'$, with probability at least $(1 - \delta')^2 \geq 1 - \delta/2$, $\sum_{t=1}^{\mathcal{T}_u-1} p_t(G) \leq \ln 4/\delta$ and $\sum_{t=\mathcal{T}_u}^{\mathcal{T}_v-1} q_t^v(G) \leq \ln 4/\delta$. \square

Lemma A.9. *The computation of the list $\{S_0, S_1, \dots\}$ by the outer **while** loop is $(\epsilon/2, \delta)$ -differentially private, with $\epsilon \in (0, 1]$ and $\delta \in (2n^{-2}, 1)$.*

Proof. We outline the proof as following: we prove that the sequence $\{\pi_0, \pi_1, \dots\}$ is produced with $(\epsilon/4, \delta/2)$ -differential privacy, no matter when we start a new phase. At each phase i , we only use private information derived from private $\rho(\widehat{S_i})$ to end a phase and calculate S_i . Next, we prove that the sequence $\{\rho(S_0), \rho(S_1), \dots\}$ is also $(\epsilon/4, \delta/2)$ -differentially private. In the end, the sequence $\{S_0, S_1, \dots\}$ is $(\epsilon/2, \delta)$ -differentially private due to the Composition Theorem by composing $\{\pi_0, \pi_1, \dots\}$ and $\{\rho(S_0), \rho(S_1), \dots\}$.

First, we prove that $\{\pi_0, \pi_1, \dots\}$ is produced with $(\epsilon/4, \delta/2)$ -differential privacy.

We use the same notations and analyses of Lemma A.3. **Case 1.a** and **Case 2.a** remain unchanged, when both node u and v are removed at the same iteration $\mathcal{T} = \mathcal{T}_u = \mathcal{T}_v$. We have to re-evaluate **Case 1.b** and **Case 2.b**, when only one of $\{u, v\}$ is removed at iteration \mathcal{T} , and the other is removed later. We assume that u is removed first at $\mathcal{T} = \mathcal{T}_u$ and v is removed at $\mathcal{T}_v > \mathcal{T}_u$. The reason for the differences is in Lemma A.3, after \mathcal{T}_u , we update the degree of v with respect to the remaining nodes and hence, this quantity is the same in both G and G' , at every remaining iteration. In Algorithm 3, we do not update the degree of v until the end of the current phase. It means that in the worst case, degree of v remains the same until it is removed, or $\forall t \in \{\mathcal{T}_u, \mathcal{T}_v\}$, $P_{tv}^G \leq \exp(\epsilon') P_{tv}^{G'}$ if $G' = G + \{u, v\}$.

Recall that all terms are the same in G and G' after v is removed:

$$P = \prod_{t=1}^{\mathcal{T}_v-1} \left(\prod_{j:p_{tj}=1} \frac{P_{tj}^G}{P_{tj}^{G'}} \times \prod_{j:p_{tj}=0} \frac{1 - P_{tj}^G}{1 - P_{tj}^{G'}} \right) \times \prod_{j:p_{\mathcal{T}_v j}=1} \frac{P_{\mathcal{T}_v j}^G}{P_{\mathcal{T}_v j}^{G'}} \times \prod_{j:p_{\mathcal{T}_v j}=0} \frac{1 - P_{\mathcal{T}_v j}^G}{1 - P_{\mathcal{T}_v j}^{G'}}$$

$$\text{Let } A_t = \prod_{j:p_{tj}=1} \frac{P_{tj}^G}{P_{tj}^{G'}}, B_t = \prod_{j:p_{tj}=0} \frac{1 - P_{tj}^G}{1 - P_{tj}^{G'}},$$

$$C = \prod_{j:p_{\mathcal{T}_v j}=1} \frac{P_{\mathcal{T}_v j}^G}{P_{\mathcal{T}_v j}^{G'}}, D = \prod_{j:p_{\mathcal{T}_v j}=0} \frac{1 - P_{\mathcal{T}_v j}^G}{1 - P_{\mathcal{T}_v j}^{G'}}$$

Case 1.b. $G + (u, v) = G'$.

Product A_t involves all terms of nodes j picked in iteration t . All terms involve in this product are the same for all nodes except for u (which is picked at \mathcal{T}_u), hence $A_t = 1$ for all $t \neq \mathcal{T}_u$, and $A_{\mathcal{T}_u} = \frac{P_{\mathcal{T}_u}^G}{P_{\mathcal{T}_u}^{G'}} = \exp(\epsilon')$.

Product B_t contains terms of remaining nodes that are not picked at iteration t . Before \mathcal{T}_u , the numerator and denominator differ by the probabilities of picking u and v in G and G' , while the other terms are the same. After iteration \mathcal{T}_u , because u is already removed, there is only v 's term left in the difference. $B_t = \frac{(1-P_{tu}^G)(1-P_{tv}^G)}{(1-P_{tu}^{G'})(1-P_{tv}^{G'})}$ for $t \in [1, \mathcal{T}_u - 1]$ and $B_t = \frac{1-P_{tv}^G}{1-P_{tv}^{G'}}$ for $t \in [\mathcal{T}_u, \mathcal{T}_v - 1]$. In either case, $B_t \leq 1$, since $P_{tu}^G \geq P_{tu}^{G'}$ and $P_{tv}^G \geq P_{tv}^{G'}$.

Product C contains terms of nodes picked at \mathcal{T}_v , hence $C = \prod_{j:p_{\mathcal{T}_v j}=1} \frac{P_{\mathcal{T}_v j}^G}{P_{\mathcal{T}_v j}^{G'}} = \frac{P_{\mathcal{T}_v}^G}{P_{\mathcal{T}_v}^{G'}} = \exp(\epsilon')$ and $D = 1$ because other than u (already removed) and v (in C), all other node's probabilities must be the same in G and G' .

Then $P = \prod_{t=1}^{\mathcal{T}_v-1} (A_t \times B_t) \times C \times D \leq \exp(2\epsilon')$.

Case 2.b $G' + (u, v) = G$

Using similar argument as in **Case 1.b**, we can calculate the values of A_t, C, D , with $\exp(\epsilon')$ is reversed to $\exp(-\epsilon')$, since we reverse the roles of G and G' . Then, $A_t = 1$ for all $t \neq \mathcal{T}_u$, and $A_{\mathcal{T}_u} = \frac{P_{\mathcal{T}_u}^G}{P_{\mathcal{T}_u}^{G'}} = \exp(-\epsilon') \leq 1$.

Similarly, $C = \prod_{j:p_{\mathcal{T}_v j}=1} \frac{P_{\mathcal{T}_v j}^G}{P_{\mathcal{T}_v j}^{G'}} = \frac{P_{\mathcal{T}_v}^G}{P_{\mathcal{T}_v}^{G'}} = \exp(-\epsilon') \leq 1$ and $D = 1$.

Now we analyze $\prod_{t=1}^{\mathcal{T}_v-1} B_t$:

$$\begin{aligned} \prod_{t=1}^{\mathcal{T}_v-1} B_t &= \prod_{t=1}^{\mathcal{T}_u-1} \frac{(1 - P_{tu}^G)(1 - P_{tv}^G)}{(1 - P_{tu}^{G'})(1 - P_{tv}^{G'})} \times \prod_{t=\mathcal{T}_u}^{\mathcal{T}_v-1} \frac{1 - P_{tv}^G}{1 - P_{tv}^{G'}} \\ &= \prod_{t=1}^{\mathcal{T}_u-1} (E_{tu} \times E_{tv}) \prod_{t=\mathcal{T}_u}^{\mathcal{T}_v-1} (E_{tv}) \\ &\leq \prod_{t=1}^{\mathcal{T}_u-1} \exp\left(\frac{4\epsilon'}{1 - e^{-1}} p_t(G)\right) \times \prod_{t'=\mathcal{T}_u}^{\mathcal{T}_v-1} \exp\left(\frac{2\epsilon'}{1 - e^{-1}} q_{t'}^v(G)\right) \\ &= \exp\left(\frac{4\epsilon'}{1 - e^{-1}} \sum_{t=1}^{\mathcal{T}_u-1} p_t(G)\right) \times \exp\left(\frac{2\epsilon'}{1 - e^{-1}} \sum_{t'=\mathcal{T}_u}^{\mathcal{T}_v-1} q_{t'}^v(G)\right) \\ &\leq \exp\left(\frac{4\epsilon' \ln(1/\delta')}{1 - e^{-1}}\right) \times \exp\left(\frac{2\epsilon' \ln(1/\delta')}{1 - e^{-1}}\right) \\ &= \exp\left(\frac{6\epsilon' \ln(1/\delta')}{1 - e^{-1}}\right) \\ &= \exp\left(\frac{6\epsilon' \ln(4/\delta)}{1 - e^{-1}}\right), \end{aligned}$$

where the first inequality is the result of Lemma A.2 and the last inequality is the result of Lemma A.8.

Then $P = \prod_{t=1}^{\mathcal{T}_v-1} (A_t \times B_t) \times C \times D \leq \exp\left(\frac{6\epsilon' \ln(4/\delta)}{1 - e^{-1}}\right)$ with probability at least $1 - \delta/2$.

Using the same argument as in Algorithm 2 and substituting $\epsilon = \frac{24\epsilon' \ln(4/\delta)}{1-e^{-1}}$, the sequence $\{\pi_0, \pi_1, \dots\}$ is $(\epsilon/4, \delta/2)$ -differentially private. For all t , we only care about non-empty π_t where at least 1 node is removed at t . Retaining all non-empty π_t preserves the $(\epsilon/4, \delta/2)$ -differential privacy, as we treat the process of dropping all empty π_t is a post-process of the private sequence $\{\pi_0, \pi_1, \dots\}$ as Post-Processing Proposition.

Next, we prove that the sequence $\{\widehat{\rho(S_0)}, \widehat{\rho(S_0)}, \dots\}$ is $(\epsilon/4, \delta/2)$ -differentially private. Since the modification of an edge of the input graph will change the value of each $\rho(S_i)$ by at most $1/|S_i|$, using Laplace mechanism, adding a noise with magnitude of $4 \ln n / (|S_i| \epsilon)$ satisfies $\epsilon / (4 \ln n)$ -differential privacy for each $\widehat{\rho(S_i)}$. Lemma 5.3 proves that the sequence $\{\widehat{\rho(S_0)}, \widehat{\rho(S_0)}, \dots\}$ has at most $\ln n$ elements with probability at least $1 - n^2 \geq 1 - \delta/2$ when $\delta \geq 2n^{-2}$. The sequence $\{\widehat{\rho(S_0)}, \widehat{\rho(S_0)}, \dots\}$ is $\epsilon/4$ -differentially private with probability at least $1 - \delta/2$ due to applying Composition Theorem on $\ln n$ elements. It implies the sequence is $(\epsilon/4, \delta/2)$ -differentially private because any mechanism which is ϵ -differentially private with probability at least $1 - \delta$ is also (ϵ, δ) -differentially private (Vadhan, 2017).

Finally, the sequence $\{S_0, S_1, \dots\}$ is constructed by using the sequence $\{\pi_0, \pi_1, \dots\}$ and the information computed from $\{\widehat{\rho(S_0)}, \widehat{\rho(S_1)}, \dots\}$. Because both sequences above are $(\epsilon/4, \delta/2)$ -differentially private, applying the Composition Theorem, $\{S_0, S_1, \dots\}$ is $(\epsilon/2, \delta)$ -differentially private. \square

Theorem A.10. Algorithm 3 preserves (ϵ, δ) -differential privacy, with $\epsilon \in (0, 1]$, $\delta \in (2n^{-2}, 1)$.

Proof. Lemma A.9 proves that the sequence $\{S_1, S_2, \dots\}$ output by the **while** loop is $(\epsilon/2, \delta)$ -differentially private. For each S_i , for any pair of neighbor graphs $G \sim G'$, the value of any $\rho(S_i)$ on G and G' differ by at most 1, hence $\Delta_\rho = 1$. Using the Exponential Mechanism, the last commands release an output with $\epsilon/2$ -differential privacy. Applying the Composition Theorem on the sequence $\{S_1, S_2, \dots\}$ which is $(\epsilon/2, \delta)$ -differentially private and the output of the Exponential Mechanism which is $\epsilon/2$ -differentially private, Algorithm 3 preserves (ϵ, δ) -differential privacy. \square

Theorem A.11. (Theorem 5.5) Let S^* be an optimal solution. With probability at least $1 - 2/n$, the set S output by Algorithm 3 satisfies $\rho(S) \geq \frac{\rho(S^*)}{4} - \frac{160}{\epsilon} \ln(1/\delta) \ln(n)$ if $\delta \leq 1/e$.

Proof. Suppose i is the first phase in which a node $v \in S^*$ is removed. Then, $S^* \subseteq S_i$. As in the proof of Theorem 4.5, we have $\rho(S^*) \leq d_{S^*}(v) \leq d_{S_i}(v)$. From Lemma A.7, it follows, with probability at least $1 - \frac{4 \ln n}{n^2}$, that $d_{S_i}(v) \leq 4\widehat{\rho(S_i)} + \frac{4}{\epsilon} \ln n \leq 4\rho(S_i) + \frac{128}{\epsilon} \ln n + \frac{4}{\epsilon} \ln n$. Therefore, $\rho(S_i) \geq \frac{\rho(S^*)}{4} - \frac{32}{\epsilon} \ln n - \frac{1}{\epsilon} \ln n$, with probability at least $1 - \frac{4 \ln n}{n^2} \geq 1 - 1/n$.

As in the proof of Theorem 4.5, with probability at least $1 - 1/n$, we have $\rho(S) \geq \max_{i'} \rho(S_{i'}) - \frac{8}{\epsilon} \ln n \geq \rho(S_i) - \frac{8}{\epsilon} \ln n \geq \frac{\rho(S^*)}{4} - \frac{40}{\epsilon} \ln n - \frac{24 \ln(4/\delta)}{\epsilon(1-1/e)} \ln n \geq \frac{\rho(S^*)}{4} - \frac{40}{\epsilon} \ln n - \frac{2.5 \cdot 24 \ln(1/\delta)}{1-1/e} \frac{1}{\epsilon} \ln n \geq \frac{\rho(S^*)}{4} - \left(\frac{40}{\ln(1/\delta)} + \frac{60}{1-1/e} \right) \frac{1}{\epsilon} \ln(1/\delta) \ln n \geq \frac{\rho(S^*)}{4} - \frac{160}{\epsilon} \ln(1/\delta) \ln n$, and the Theorem follows. \square

A.4. Lower bound

Lemma A.12. Let $G = (V, E)$ denote a graph with $|V| = n$ and $E = \emptyset$. Let V be partitioned into sets V_1, \dots, V_N , for $N = n/a$, with each $|V_i| = a$. For $i \in \{1, \dots, N\}$, define $\mathcal{S}_i = \{S \subseteq V : S \cap V_i \neq \emptyset, |S| \leq a^2\}$. Let $M(G)$ be an ϵ -differentially private mechanism. There exists an $i \in \{1, \dots, N\}$ such that $\Pr[M(G) \in \mathcal{S}_i] \leq \frac{2a^3}{n}$.

Proof. We now prove that there exists an $i \in \{1, \dots, N\}$ such that $\Pr[M(G) \in \mathcal{S}_i] \leq \frac{2a^3}{n}$. Observe that

$$\mathbb{E} \left[|M(G)| \mid |M(G)| \leq a^2 \right] = \sum_{i=1}^N \mathbb{E} \left[|M(G) \cap V_i| \mid |M(G)| \leq a^2 \right] \leq a^2,$$

which implies that there exists i such that $\mathbb{E} \left[|M(G) \cap V_i| \mid |M(G)| \leq a^2 \right] \leq \frac{a^2}{N} = \frac{a^3}{n}$.

By the Markov's inequality, we have $\Pr \left[|M(G) \cap V_i| > \frac{n}{2a^3} \frac{a^3}{n} \mid |M(G)| \leq a^2 \right] \leq \frac{2a^3}{n}$, which implies $\Pr[M(G) \cap V_i \neq \emptyset \mid |M(G)| \leq a^2] \leq \Pr[|M(G) \cap V_i| \geq 1 \mid |M(G)| \leq a^2] \leq \frac{2a^3}{n}$. Finally,

$$\begin{aligned} & \Pr[M(G) \in \mathcal{S}_i] \\ &= \Pr[M(G) \cap V_i \neq \emptyset, |M(G)| \leq a^2] \\ &= \Pr[M(G) \cap V_i \neq \emptyset \mid |M(G)| \leq a^2] \Pr[|M(G)| \leq a^2] \\ &\leq \frac{2a^3}{n} \end{aligned}$$

□

A.5. Map-Reduce Implementation

Algorithm 4 MRDENSEDP(G, ϵ, δ)

Input: $G \in \mathcal{G}$, privacy parameters ϵ, δ .

Output: A subset $S \subseteq V(G)$

```

1: For all edges  $(u, v) \in E(G)$ , emit  $\langle u, v \rangle$  and initialize variables as in PHASEDENSEDP
2: while  $|S_i| \neq \emptyset$  do
3:   /*Start a new phase  $i$ */
4:   Reduce sub-phase: Input  $\langle u; v_1, v_2, \dots \rangle$ 
5:   if  $u \in \{v_1, v_2, \dots\}$  then
6:     Ignore node  $u$ , it was sampled out in the last phase
7:   else
8:     Emit  $\langle u; \{v_1, v_2, \dots\} \rangle$ 
9:     Emit  $\langle S_i; u \rangle$ 
10:    Emit  $\langle \rho(S_i); \deg_{S_i}(u) = |\{v_1, v_2, \dots\}| \rangle$ 
11:  end if
12:  Reduce sub-phase: Input  $\langle S_i; u_1, u_2, \dots \rangle$ 
13:   $S_i = \{u_1, u_2, \dots\}$ 
14:  Reduce sub-phase: Input  $\langle \rho(S_i); \deg_{S_i}(u_1), \deg_{S_i}(u_2), \dots \rangle$ 
15:   $\rho(S_i) = \sum_i \deg_{S_i}(u_i) / 2$ 
16:  Calculate  $\widehat{\rho(S_i)}$  and  $T_i$  as in PHASEDENSEDP, line 12-13
17:  Map phase sub-phase: Input  $\langle u, \{v_1, v_2, \dots\} \rangle$ 
18:  Calculate  $\deg_{S_i}(u) = |\{v_1, v_2, \dots\}|$ 
19:  Sample  $T_u$  as in PHASEDENSEDP, line 11
20:  if  $T_u \leq T_i$  then
21:    Emit  $\langle u, u \rangle$ 
22:  else
23:    For  $v \in \{v_1, v_2, \dots\}$ , emit  $\langle v, u \rangle$ 
24:  end if
25:   $i = i + 1$ 
26: end while
27: Return  $S_i$  from distinct  $\{S_i\}$  with probability proportional to  $e^{\epsilon \rho(S_i)/2}$ 

```

Lemma A.13. MRDENSEDP preserves (ϵ, δ) -differential privacy, for $\epsilon \in (0, 1]$, $\delta \in (2n^{-2}, 1)$ and runs in $O(\ln n)$ phases, with probability at least $1 - 1/n^2$.

Proof. We prove that for all i , the i^{th} phase of MRDENSEDP is equivalent to the i^{th} phase of PHASEDENSEDP. The lemma then follows.

In the first reduce sub-phase, with its input is every edge (u, v) , the sub-phase detects a removed node by looking for the node in its neighbor sets. It is because in lines 19-20, the algorithm marks a removed node in the previous phase by adding an edge that points to the node itself. MRDENSEDP constructs three entities for each remained node u : the neighbor nodes

of $u : \{v_1, v_2, \dots\}$, assigning u to candidate subgraph S_i and the degree of u .

In the second and third reduce sub-phases, they gather all remained node u and their degrees to construct the candidate subgraph S_i and $\rho(S_i)$ as the results of the phase. They are the same as S_i and $\rho(S_i)$ produced by Algorithm 3.

The map sub-phase takes input as pairs of node u and u 's neighbors $\{v_1, v_2, \dots\}$. The sub-phase calculates the degree of node u with respect to the current nodes in phase i and samples the removal time T_u of each node u as in line 11 of Algorithm 3. It compares T_u to T_i (of phase i) to determine if u is removed in this phase. If it is removed, the sub-phase only emits an self-edge to mark that the node is removed. It guarantees that in the next phases, there is no edge (v, u) for any $v \neq u$ and hence, u will not exist in the neighbor set of any node v . Otherwise, it emits all edge (v, u) which will be used to construct the neighbor set of each node v in the next phase. \square

Network	Nodes	Edges	Description	Network	Nodes	Edges	Description
ca-GrQc	5242	14496	Collab. net. Arxiv Gen. Relativity	deezer_europe	28281	92752	Social net. Deezer users from Europe
ca-HepTh	9877	25998	Collab. net. Arxiv HE Phy. Theory	musae_DE	9498	153138	Social net. of Twitch users (DE)
ca-HepPh	12008	118521	Collab. net. Arxiv HE Phy.	musae_ENGB	7126	35324	Social net. of Twitch users (ENGB)
ca-AstroPh	18772	198110	Collab. net. Arxiv Astro Phy.	musae_facebook	22470	171002	Facebook page-page net.
ca-CondMat	23133	93497	Collab. net. Arxiv Cond. Matter	musae_FR	6549	112666	Social net. of Twitch users (FR)
email-Enron	36692	183831	Email net. from Enron	musae_git	37700	289003	Social net. of Github developers.
loc-gowalla	196591	950327	Gowalla location based social net	musae_squirrel	5201	198493	Wiki page-page net. on topic squirrel
loc-brightkite	58228	214078	Brightkite location based social net	musae_crocodile	11631	170918	Wiki page-page net. on topic crocodile
facebook	4039	88234	Social circles from Facebook	musae_chameleon	2277	31421	Wiki page-page net. on topic chameleon
lastfm_asia	7624	27806	Social net. LastFM users from Asia	musae_PTBR	1912	31299	Social net. of Twitch users (PTBR)

Table 2. Summary of networks in our experiments. All networks are undirected, with sizes range from thousands to hundreds thousands. (Leskovec & Krevl, 2014)

B. Experiments

B.1. Experimental Setup

In private algorithms, there are two parameters to control the privacy guarantee: ϵ and δ . We can interpret ϵ as the upper bound on the absolute value of *privacy loss*. While there is no strict range for the values of ϵ , we are usually interested in small ϵ , since it provides better privacy guarantee (Dwork, 2011). In the experiments, we test our algorithms with ϵ varying in the range $[0.5, 16]$, which is similar to the range of ϵ in (Abadi et al., 2016). We infer δ as the “upper bound on the probability of catastrophic failure (e.g., the entire dataset being published in the clear)” (Vadhan, 2017). Hence, all applications must choose δ to be cryptographically negligible ($\delta \leq n^{-\omega(1)}$) (Vadhan, 2017) (Dwork, 2011). Since the networks we study have several hundreds of thousands of edges, we select $\delta \in \{10^{-6}, 10^{-9}\}$ for our experiments.

We perform each experiment (as a combination of algorithm with parameters) repeatedly 10 times to report the average of estimations. We must note that running multiple instances of the same algorithm is only for the sole purpose of performance analysis. In practice, multiple executions of the same private mechanism will degrade the privacy guarantee, in the worst case linearly to the number of executions (Dwork et al., 2014).

B.2. Experimental Results

Performance of parallel algorithms. Figure 6 shows the accuracy comparison of PARDENSEDP and PHASEDENSEDP. In general, PARDENSEDP outperforms PHASEDENSEDP at $\epsilon \geq 2$. For small ϵ , neither algorithm has good accuracy, though PHASEDENSEDP is slightly better. As we can see from Figure 4, PARDENSEDP is efficient in many real networks and settings, it is reasonable to choose it over PHASEDENSEDP in practice.

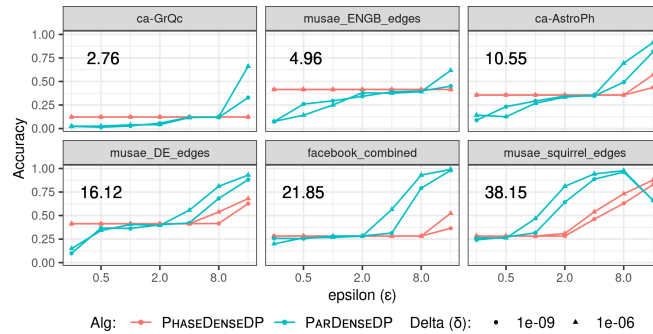


Figure 6. Comparison of relative density between PARDENSEDP and PHASEDENSEDP.

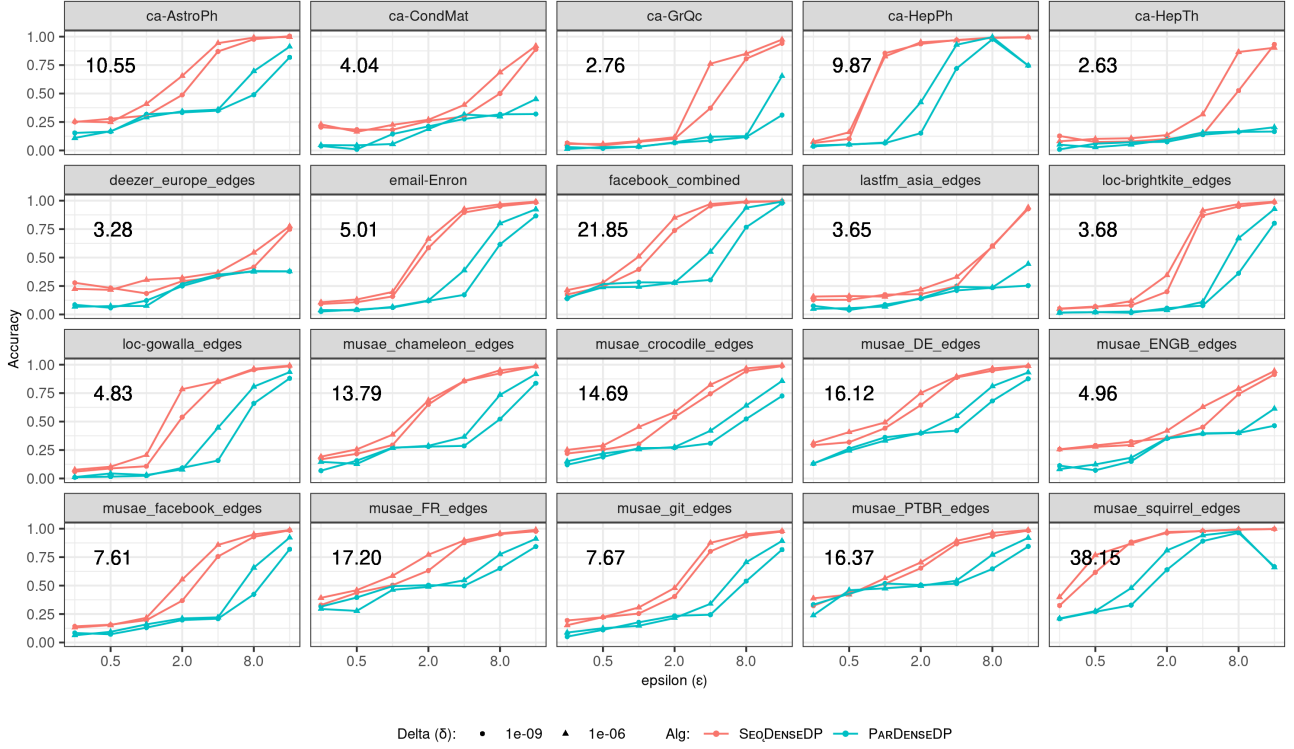


Figure 7. Accuracy of our private algorithms. The accuracy is measured in terms of the relative density. The number in each graph indicates the density of the graph as whole (which equals the average degree). An accuracy close to 1.0 means the private subgraph's density is as good as the non-private baseline's.