Privately Estimating Graph Parameters in Sublinear Time

Jeremiah Blocki
Department of Computer Science, Purdue University, West Lafayette, IN, USA

Elena Grigorescu
Department of Computer Science, Purdue University, West Lafayette, IN, USA

Tamalika Mukherjee
Department of Computer Science, Purdue University, West Lafayette, IN, USA

Abstract

We initiate a systematic study of algorithms that are both differentially-private and run in sublinear time for several problems in which the goal is to estimate natural graph parameters. Our main result is a differentially-private $(1+\rho)$-approximation algorithm for the problem of computing the average degree of a graph, for every $\rho > 0$. The running time of the algorithm is roughly the same (for sparse graphs) as its non-private version proposed by Goldreich and Ron (Sublinear Algorithms, 2005). We also obtain the first differentially-private sublinear-time approximation algorithms for the maximum matching size and the minimum vertex cover size of a graph.

An overarching technique we employ is the notion of *coupled global sensitivity* of randomized algorithms. Related variants of this notion of sensitivity have been used in the literature in ad-hoc ways. Here we formalize the notion and develop it as a unifying framework for privacy analysis of randomized approximation algorithms.

2012 ACM Subject Classification Security and privacy → Privacy-preserving protocols; Theory of computation → Streaming, sublinear and near linear time algorithms

Keywords and phrases differential privacy, sublinear time, graph algorithms

Digital Object Identifier 10.4230/LIPIcs.ICALP.2022.26

Category Track A: Algorithms, Complexity and Games

Related Version *Full Version*: `https://arxiv.org/abs/2202.05776`

Funding *Jeremiah Blocki*: supported in part by NSF CNS-1931443 and NSF CCF-1910659.
Elena Grigorescu: supported in part by NSF CCF-1910659 and NSF CCF-1910411.
Tamalika Mukherjee: supported in part by NSF CCF-1910659 and NSF CCF-1910411.

Acknowledgements We thank several anonymous reviewers for their valuable feedback on a preliminary version of this work. We thank Soheil Behnezhad for bringing to our attention the subtlety in the analysis of [26], first discovered by [10], and finally resolved in his recent work [3]. We also thank Jakub Tetek for bringing up several points for clarification, which we have incorporated in the current version.

1 Introduction

Graphs are frequently used to model massive data sets (e.g., social networks) where the users are the nodes, and their relationships are the edges of the graphs. These relationships often consist of sensitive information, which drives the need for privacy in this setting.

Differential Privacy (DP) [12] has become the gold standard in privacy-preserving data analysis due to its compelling privacy guarantees and mathematically rigorous definition. Informally, a randomized function computed on a graph is *differentially private* if the distribution of the function's output does not change significantly with the presence or

absence of an individual edge (or node). See [13] for a comprehensive tutorial on differential privacy.

▶ **Definition 1** (Differential-privacy). *Let $\mathcal{G}_n$ denote the set of all n-node graphs. An algorithm $\mathcal{A}$ is (ε, δ) node-DP (resp. edge-DP) if for every pair of node-neighboring (resp. edge-neighboring)*[1] *graphs $G_1, G_2 \in \mathcal{G}_n$, and for all sets $\mathcal{S}$ of possible outputs, we have that $\Pr[\mathcal{A}(G_1) \in \mathcal{S}] \leqslant e^{\varepsilon} \Pr[\mathcal{A}(G_2) \in \mathcal{S}] + \delta$. When $\delta = 0$ we simply say that the algorithm is ε-DP.*

Since the graphs appearing in modern applications are massive, it is also often desirable to design *sublinear-time* algorithms that approximate natural combinatorial properties of the graph, such as the average degree, the number of connected components, the cost of a minimum spanning tree, the number of triangles, the size of a maximum matching, the size of a minimum vertex cover, etc. For an excellent survey on sublinear-time algorithms for approximating graph parameters, we refer the reader to [29].

There has been a lot of work in developing differentially-private algorithms for estimating graph parameters in polynomial-time, with respect to *edge differential privacy*, i.e., neighboring graphs that differ by a single edge in Definition 1. Nissim, Raskhodnikova, and Smith [25] demonstrated the first edge-differentially private graph algorithms. They showed how to estimate the cost of a minimum spanning tree and the number of triangles in a graph by calibrating noise to a local variant of sensitivity called *smooth sensitivity*. Subsequent works in designing edge differentially-private algorithms for computing graph statistics include [21, 19, 23, 36]. Gupta, Ligett, McSherry, Roth and Talwar [18] gave the first edge differentially-private algorithms for classical graph optimization problems, such as vertex cover, and minimum s-t cut, by making clever use of the exponential mechanism in existing non-private algorithms that solve the same problem.

An even more desirable notion of privacy in graphs is the notion of *node differential privacy* i.e., neighboring graphs that differ by a single node and edges incident to it in Definition 1. The concept of node differentially-private algorithms for 1-dimensional functions (functions that output a single real value) on graphs was first rigorously studied independently by Kasiviswanathan, Nissim, Raskhodnikova and Smith [22], as well as, Blocki, Blum, Datta, and Sheffet [4], and Chen and Zhou [9]. Their techniques were later extended to higher-dimensional functions on graphs [28, 6]. Subsequent works have focused on developing node differentially-private algorithms for a family of network models: stochastic block models and graphons [7, 30]. A more recent line of work has focused on the continual release of graph statistics such as degree-distributions and subgraph counts in an online setting [33, 15]. Gehrke, Lui, and Pass [16] introduce a more robust notion of differential privacy called Zero-Knowledge Differential Privacy (ZKDP), which tackles the problem of auxiliary information in social networks. This work uses existing results from sublinear-time algorithms as a building block to achieve ZKDP for several graph problems. However, it is important to note that the final ZKDP mechanisms are not computable in sublinear-time.

The literature on designing differentially-private algorithms for estimating graph parameters in sublinear time is far less developed. The only paper we are aware of is due to Sivasubramaniam, Li and He [32], who give the first sublinear-time differentially-private algorithm for approximating the average degree of a graph. Our work addresses this gap

[1] Graphs $G_1 = (V, E_1)$, $G_2 = (V, E_2)$ are *node-neighboring*, denoted by $G_1 \sim_v G_2$, if there exists a vertex $v \in V$ such that $E_1(V \setminus \{v\}) = E_2(V \setminus \{v\})$. Graphs G_1 and G_2 are *edge-neighboring* i.e., $G_1 \sim_e G_2$ if there exists an edge e such that $E_1 \setminus \{e\} = E_2 \setminus \{e\}$.

by initiating a systematic study of differentially-private sublinear-time algorithms for the problems of estimating the following graph parameters: (1) the average-degree of a graph, (2) the size of a maximum matching, and (3) the size of a minimum vertex cover. As an overarching technique, we formally introduce the notion of *Coupled Global Sensitivity* and use it to analyze the privacy of our randomized approximation algorithms.

1.1 Our Results

1.1.1 Privately Approximating the Average Degree

We obtain a differentially-private sublinear-time algorithm for estimating the average degree $\bar{d}_G = \frac{\sum_{v \in V} \deg(v)}{|V|}$, of a graph $G = (V, E)$, with respect to edge-differential privacy, which achieves a multiplicative approximation of $(1+\rho)$, for any constant $\rho > 0$. Specifically, our algorithm outputs a value $\tilde{d}$ such that w.h.p. we have $(1-\rho)\bar{d}_G \leqslant \tilde{d} \leqslant (1+\rho)\bar{d}_G$, for graphs with $\bar{d}_G = \Omega(1)$. Throughout the paper we denote $|V| = n$.

We work in the *neighbor-query* model, in which we are given oracle access to a simple graph $G = (V, E)$, where the algorithm can obtain the identity of the i-th neighbor of a vertex $v \in V$ in constant time. If $i > \deg(v)$ for a particular vertex v, then $\perp$ is returned. The algorithm may also perform *degree queries*, namely for any $v \in V$ it can obtain $\deg(v)$ in constant time.

▶ **Theorem 2.** *There is an ε-edge differentially-private $(1+\rho)$-approximation algorithm for estimating the average degree $\bar{d}_G \geqslant 1$ of a graph G on n vertices that runs in time*[2] *$O(\sqrt{n} \cdot poly(\log(n)/\rho) \cdot poly(1/\varepsilon))$ where $\varepsilon^{-1} = o(\log^{1/4}(n))$.*

The problem of estimating the average degree of a graph was first studied by Feige [14], who gave a sublinear time $(2+\rho)$-approximation (multiplicative) for any constant $\rho > 0$, making $O(\sqrt{n/d_0})$ many degree queries, where d_0 is a lower bound on the number of queries. He also notes that $\Omega(\sqrt{n/d_0})$ queries are necessary for a $2 - o(1)$-approximation, and hence, for the interesting cases when we may assume $d_0 \geqslant 1$, $\Omega(n)$ degree queries are necessary[3]. Goldreich and Ron [17] subsequently gave a $(1+\rho)$-approximation using both degree and neighbor queries, running in time $\tilde{O}((n/\sqrt{m}) \cdot \mathrm{poly}(1/\rho))$. This bound is also tight, since every constant-factor approximation algorithm must make $\Omega(n/\sqrt{\rho m})$ degree and neighbor queries [17]. A simpler analysis achieving the same bounds was given by Seshadri [31]. Further, Dasgupta, Kumar and Sarlós [11] studied this problem in the model where access to the graph is via samples, in the context of massive networks where the number of nodes may not be known. They obtain a $(1+\rho)$-approximation that uses roughly $O(\log d_U \cdot \log\log d_U)$ samples where d_U is an upper bound on the maximum degree of the graph.

In recent work, Sivasubramaniam, Li and He [32] gave a sublinear-time differentially-private algorithm for approximating the average degree of a graph using Feige's [14] algorithm. Their algorithm achieves a $(2+\rho+o(1))$-approximation for every constant $\rho > 0$. They achieve this by calculating a tight bound for the global sensitivity of the final estimate of Feige's algorithm and adding Laplace noise with respect to this quantity appropriately. By contrast, we achieve a $(1+\rho)$-approximation for any constant $\rho > 0$ – assuming that the privacy parameter is $\varepsilon^{-1} = o(\log^{1/4} n)$.

[2] from here on, we use running time and number of queries interchangeably.

[3] Observe that for $\bar{d}_G = o(1)$ a multiplicative approximation algorithm that can distinguish between two graphs on n vertices, one with 0 edges, and another with, say 1 edge, must sample $\Omega(n)$ vertices, and hence cannot be running in sublinear time.

1.1.2 Privately Approximating the Size of a Maximum Matching and Minimum Vertex Cover

Given an undirected graph, a set of vertex-disjoint edges is called a *matching*. A matching M is *maximal* if M is not properly contained in another matching. A matching M is *maximum* if for any other matching M', $|M| \geqslant |M'|$. A *vertex cover* of a graph is a set of vertices that includes at least one endpoint of every edge of the graph. A *minimum* vertex cover is a vertex cover of the smallest possible size. For a minimization problem, we say that a value $\hat{y}$ is an (α, β)-approximation to y if $y \leqslant \hat{y} \leqslant \alpha y + \beta$. For a maximization problem, we say that a value $\hat{y}$ is an (α, β)-approximation to y if $\frac{y}{\alpha} - \beta \leqslant \hat{y} \leqslant y$. An algorithm $\mathcal{A}$ is an (α, β)-approximation for a value $V(x)$ if it computes an (α, β)-approximation to $V(x)$ with probability at least $2/3$ for any proper input x.

For a graph $G = (V, E)$, we work in the *bounded degree* model, where one can query an i-th neighbor ($i \in [d]$) of a vertex in constant time; denote this query as $\mathrm{Nbr}(v, i)$. Here d is the maximum degree of the graph. If $i > \deg(v)$ for a particular vertex v, then $\mathrm{Nbr}(v, i) = \perp$. We also assume query access to the degree of a vertex, i.e., one can query $\deg(v)$ for any $v \in V$ in constant time.

▶ **Theorem 3.** *There is an ε-(node and edge) differentially-private algorithm for the maximum matching problem that reports a $(2, \rho n)$-approximation with probability $1-(2/n^4+1/n^{192\varepsilon/\rho})$, and runs in expected time $\tilde{O}\left((\bar{d}+1)/\rho^2\right)$, where $\bar{d}$ is the average degree of the input graph.*

▶ **Theorem 4.** *There is an ε-(node and edge) differentially-private algorithm for the minimum vertex cover problem that reports a $(2, \rho n)$-approximation with probability $1-(2/n^4+1/n^{96\varepsilon/\rho})$, and runs in expected time $\tilde{O}\left((\bar{d}+1)/\rho^2\right)$, where $\bar{d}$ is the average degree of the input graph.*

Typically, the privacy parameter ε is a constant, and so is the approximation parameter ρ, in which case the success probability in the theorems above is $1 - 1/(\mathrm{poly}(n))$.

The question of approximating the size of a vertex cover in sublinear-time was first posed by Parnas and Ron [27], who obtained a $(2, \rho n)$-approximation in time $d^{O(\log d/\rho^3)}$, where d is the maximum degree of the graph. Nguyen and Onak [24] improved upon this result by giving a $(2, \rho n)$-approximation for the maximum matching problem, and consequently a $(2, \rho n)$-approximation for the vertex cover problem, in time $O(2^{O(d)/\rho^2})$. The result of [24] was later improved by Yoshida, Yamamoto and Ito [35], who gave an ingenious analysis of the original algorithm to achieve a running time of $O(d^4/\rho^2)$. Onak, Ron, Rosen and Rubinfeld [26] proposed a near-optimal time complexity of $\tilde{O}(\bar{d} \cdot \mathrm{poly}(1/\rho))$, where $\bar{d}$ is the average degree of a graph, but Chen, Kannan, and Khanna [10] identified a subtlety in their analysis, which proved to be crucial to their improved time complexity claim. Very recently, building on ideas from the analysis of [35], Behnezhad [3] gave a new analysis for achieving a $(2, \rho n)$-approximation to the size of maximum matching and minimum vertex cover in time $\tilde{O}((\bar{d}+1)/\rho^2)$. Behnezhad's result nearly matches the lower bound given by Parnas and Ron [27], who showed that $\Omega(\bar{d}+1)$ queries are necessary for obtaining a $(O(1), \rho n)$-estimate in the case of the maximum matching or minimum vertex cover problem.

Our final DP algorithm simply runs the non-private approximation algorithm [3] and then adds Laplace noise proportional to the Coupled Global Sensitivity (of the non-private algorithm). Thus, our time complexity is identical to the non-private approximation algorithm. We show that the added Laplace noise is small enough that it preserves the approximation guarantees of the non-private approximation algorithm.

1.2 Organization

We define and motivate the notion of Coupled Global Sensitivity as a privacy tool in Section 1.3. Then we give a high-level overview of the techniques used for our results in Section 1.4. The formal privacy and accuracy analysis of Theorem 2 are in Sections 2 and 4. The formal analysis for Theorems 3 and 4 are in the full version [5]. We conclude with some open problems in Section 5.

1.3 Coupled Global Sensitivity as a Tool in Privacy analysis

Background and Motivation. Given a query $f : \mathcal{D} \to \mathbb{R}^d$ a general mechanism to answer the query privately is to compute $f(D)$ and then add noise. The global sensitivity of a function was introduced in the celebrated paper by Dwork, McSherry, Nissim and Smith [12], who showed that it suffices to perturb the output of the function with noise proportional to the global sensitivity of the function in order to preserve differential privacy.

▶ **Definition 5** (Global sensitivity). *For a query $f : \mathcal{D} \to \mathbb{R}^d$, the global sensitivity of f (wrt the ℓ_1-metric) is given by*

$$GS_f = \max_{A,B \in \mathcal{D}: A \sim B} \|f(A) - f(B)\|_1 .$$

One can preserve differential privacy by computing $f(D)$ and adding Laplacian noise[4] scaled to the global sensitivity of f, where D is a database. However, in many contexts we may not be able to compute the function f exactly. For example, if the dataset D is very large and our algorithm needs to run in sublinear-time or if the function f is intractable e.g., $f(G)$ is the size of the minimum vertex cover. In cases where we cannot compute f exactly, an attractive alternative is to use a randomized algorithm, say $\mathcal{A}_f$, to approximate the value of f. Given an approximation algorithm $\mathcal{A}_f$ it is natural to ask whether or not we can add noise to $\mathcal{A}_f(D)$ to obtain a differentially private approximation of $f(D)$ and (if possible) how to scale the noise. We first observe that computing $\mathcal{A}_f(D)$ and adding noise scaled to the global sensitivity of f does not necessarily work. Intuitively, this is because the sensitivity of $\mathcal{A}_f$ can be vastly different from that of f. For example, suppose that $GS_f = 1$, $f(D) = n = f(D') + 1$ for neighboring datasets $D \sim D'$ and that our approximation algorithm guarantees that $0.999 \cdot f(D) \leqslant \mathcal{A}_f(D) \leqslant 1.001 \cdot f(D)$. It is possible that $A_f(D) = 1.001n$ and $A_f(D') = 0.999(n-1)$ so that $|A_f(D) - A_f(D')| \geqslant 0.002n$ which can be arbitrarily larger than GS_f as n increases.

Coupled Global Sensitivity. We propose the notion of *coupled global sensitivity* of randomized algorithms as a framework for providing general-purpose privacy mechanisms for approximation algorithms running on a database D. In this framework, our differentially-private algorithms can follow a unified strategy, in which in the first step a non-private randomized approximation algorithm $\mathcal{A}_f(D)$ is run on the dataset, and privacy is obtained by adding Laplace noise proportional with the coupled global sensitivity of $\mathcal{A}_f$[5]. The concept of coupled global sensitivity has been used implicitly in prior work on differential privacy e.g., see [1, 8]. Our work formalizes this notion as a general tool that can be used to design and analyze differentially private approximation algorithms.

[4] Here, the probability density function of the Laplace distribution $\mathrm{Lap}(\lambda)$ is $h(z) = \frac{1}{2\lambda} \exp\left(-\frac{|z|}{\lambda}\right)$.

[5] We note that this is the simplest application of CGS, and as we will see in the analysis of estimating the average degree, we can use CGS to add noise to intermediate quantities used by the randomized algorithm as well.

Notation. When $\mathcal{A}$ is a randomized algorithm we use the notation $x := \mathcal{A}(D; r)$ to denote the output when running $\mathcal{A}$ on input D with fixed random coins r. Similarly, $\mathcal{A}(D)$ can be viewed as a random variable taken over the selection of the random coins r.

▶ **Definition 6** (Coupling). *Let Z and Z' be two random variables defined over the probability spaces $\mathcal{Z}$ and $\mathcal{Z}'$, respectively. A coupling of Z and Z', is a joint variable (Z_c, Z'_c) taking values in the product space $(\mathcal{Z} \times \mathcal{Z}')$ such that Z_c has the same marginal distribution as Z and Z'_c has the same marginal distribution as Z'. The set of all couplings is denoted by* $\mathsf{Couple}(Z, Z')$.

▶ **Definition 7** (Coupled global sensitivity of a randomized algorithm). *Let $\mathcal{A} : \mathcal{D} \times \mathcal{R} \to \mathbb{R}^k$ be a randomized algorithm that outputs a real-valued vector. Then the* coupled global sensitivity *of $\mathcal{A}$ is defined as*

$$\mathsf{CGS}_{\mathcal{A}} := \max_{D_1 \sim D_2} \min_{C \in \mathsf{Couple}(\mathcal{A}(D_1), \mathcal{A}(D_2))} \max_{(z,z') \in C} \|z - z'\|_1$$

▶ Remark 8. We can try to relax the definition of Coupled Global Sensitivity as follows: $\mathsf{CGS}_{\mathcal{A},\delta}$ is the minimum value, say x such that for all neighboring inputs $D_1 \sim D_2$, there exists a coupling C such that $\Pr_{(z,z') \sim C}[|z - z'| > x] \leqslant \delta$. We need to be careful here as we need to ensure that the minimum value x is always well-defined. If we can ensure this, then we can also show that adding noise proportional to $\mathsf{CGS}_{\mathcal{A},\delta}$ preserves (ε, δ)-differential privacy.

▶ **Fact 9.** *Let $\mathcal{A} : \mathcal{D} \times \mathcal{R} \to \mathbb{R}^k$ be a randomized algorithm viewed as a function that takes as input a dataset $\mathcal{D}$ and a random string in the finite set $\mathcal{R}$, and outputs a real-valued vector. For a finite set $\mathcal{R}$, denote by $\mathsf{Sym}(\mathcal{R})$ the symmetric group of all permutations on the elements in $\mathcal{R}$. Then,*

$$\mathsf{CGS}_{\mathcal{A}} \leqslant \max_{D_1 \sim D_2} \min_{\sigma \in \mathsf{Sym}(\mathcal{R})} \max_{R \in \mathcal{R}} \|\mathcal{A}(D_1; R) - \mathcal{A}(D_2; \sigma(R))\|_1$$

The following theorem formalizes the fact that adding noise proportional to the coupled global sensitivity of a randomized algorithm preserves differential privacy (see full version [5] for a formal proof).

▶ **Theorem 10.** *Let $\mathcal{A} : \mathcal{D} \to \mathbb{R}^k$ be a randomized algorithm and define the Laplace mechanism $\mathcal{M}_L(D) = \mathcal{A}(D) + (Y_1, \ldots, Y_k)$, where Y_i are i.i.d. random variables drawn from $Lap(\mathsf{CGS}_{\mathcal{A}}/\varepsilon)$. The mechanism $\mathcal{M}_L$ preserves ε-differential privacy.*

How we use Coupled Global Sensitivity. In our algorithm for estimating the average degree we divide the algorithm into randomized sub-routines and show that the CGS of these sub-routines is small, therefore enabling us to add Laplacian noise proportional to the CGS and ensure the privacy of each sub-routine, and by composition, the privacy of the entire algorithm (See Theorem 13). Similarly, we show that the existing non-private sublinear-time algorithms for maximum matching and minimum vertex cover have small CGS, therefore enabling us to add Laplace noise proportional to the CGS to their outputs thus making them differentially-private (See full version [5]).

1.4 Technical Overview

1.4.1 Privately Estimating the Average Degree

At a high-level, our private algorithm for estimating the average degree follows the non-private variant of Goldreich and Ron [17]. However, there are several challenges that prevent us from simply being able to add Laplacian noise to the output. We overcome these challenges

by first obtaining a new non-private algorithm with the same approximation ratio as that of [17], and then further add appropriate amounts of noise in several steps of the algorithm to obtain both privacy and accuracy guarantees. We begin by describing the algorithm of [17].

The Goldreich-Ron algorithm [17]. The strategy of the original non-private algorithm in [17] is to sample a set S of vertices and partition them into buckets S_i based on their degrees. In particular, for each i we set $S_i = B_i \cap S$ where the set B_i contains all vertices of degrees ranging between $((1+\beta)^{i-1}, (1+\beta)^i]$, where $\beta = \rho/c$ for some constant $c > 1$. Intuitively, as long as $|S_i|$ is sufficiently large the quantity $|S_i|/|S|$ is a good approximation for $|B_i|/n$ with high probability. Let I denote the indices i for which $|S_i|$ is sufficiently large. We can partition edges from the graph into three sets (1) edges with both endpoints in $\bigcup_{i\in I} B_i$, (2) edges with exactly one endpoint in $\bigcup_{i\in I} B_i$, and (3) edges with no endpoints in $\bigcup_{i\in I} B_i$. When the threshold for "large buckets" is tuned appropriately one can show that (whp) type 3 edges can be ignored as there are at most $o(n)$ such edges.

We could use $(1/|S|)\sum_{i\in I}|S_i|(1+\beta)^{i-1}$ as an approximation for $\frac{1}{n}\sum_{i\in I}\sum_{v\in B_i}\deg(v)$. The previous sum counts type (1) edges twice, type (2) edges once and type (3) edges zero times. While it is okay to ignore type (3) edges there could be a lot of type (2) edges which are under-counted. To correct for type (2) edges we can instead try to produce an approximation for the sum $\frac{1}{n}\sum_{i\in I}\sum_{v\in B_i}(1+\alpha_v)\deg(v)$ where α_v denotes the fraction of type (2) edges incident to v. Intuitively, α_v is included to ensure that type (2) edges are also counted twice. For each sampled node $v \in S_i$ we can pick a random neighbor $r(v)$ of v and define $X(v) = 1$ if $r(v) \notin \bigcup_{i\in I} B_i$; otherwise $X(v) = 0$. Observe that in the expected value of the random variable is $\mathbb{E}[X(v)] = \alpha_v$. Since $|S_i|$ is reasonably large for each $i \in I$ and $\deg(u) \approx \deg(v)$ for each pair $u, v \in S_i$ we can approximate the fraction of type (2) edges incident to B_i as $W_i/|S_i|$ where $W_i = \sum_{v\in S_i} X(v)$. Finally, we can use $(1/|S|)\sum_{i\in I}|S_i|(1+W_i/|S_i|)(1+\beta)^{i-1}$ as our final approximation for the average degree.

Challenges to making the original algorithm private by adding noise naively. The first naive attempt to transform the algorithm of [17] into a differentially private approximation would be to add noise to the final output. However, the coupled global sensitivity of this algorithm is large enough that the resulting algorithm is no longer a $(1+\rho)$-approximation.

A second natural strategy to make the above algorithm differentially private is to add Laplace noise to the degree of each vertex and partition vertices in S based on their noisy degrees $\tilde{d}(v) = \deg(v) + Y_v$ where $Y_v \sim \mathrm{Lap}(6/\varepsilon)$. (Note: To ensure that the algorithm still runs in sublinear time we could utilize lazy sampling and only sample $Y_v \sim \mathrm{Lap}(6/\varepsilon)$ when needed). In particular, we can let $\tilde{S}_i = S \cap \tilde{B}_i$ where $\tilde{B}_i$ denotes the set of all nodes v with noisy degree $\tilde{d}(v)$ ranging between $((1+\beta)^{i-1}, (1+\beta)^i]$. Now we can compute $W_i = Z_i + \sum_{v\in\tilde{S}_i} X(v)$ where $Z_i \sim \mathrm{Lap}(6/\varepsilon)$ and return $(1/|S|)\sum_{i\in I}|\tilde{S}_i|(1+\frac{W_i}{|\tilde{S}_i|})(1+\beta)^{i-1}$. While the above approach would preserve differential privacy, the final output may not be accurate. The problem is that the noise Y_v may cause a node v to shift buckets. It is not a problem if $v \in B_i$ shifts to an adjacent bucket i.e., $v \in \tilde{B}_{i-1}$ or $v \in \tilde{B}_{i+1}$ since $(1-\beta)^{i-2}$ and $(1-\beta)^{i+1}$ are still reasonable approximations for the original degree $\deg(v) \in ((1+\beta)^{i-1}, (1+\beta)^i]$. Indeed, when $\deg(v)$ is sufficiently large we can argue that $(1-\beta)\deg(v) < \tilde{d}(v) < (1+\beta)\deg(v)$ with high probability. However, this guarantee does not apply when $\deg(v)$ is small. In this case the Laplace noise Y_v might dominate $\deg(v)$ yielding an inaccurate approximation. Sivasubramaniam et al. [32], made similar observations, and because of these technical barriers, their paper analyzes the simpler strategy for estimating the average degree, which yields a less accurate result. The crucial observation here is that we need to deal with vertices having small degrees in our accuracy analysis separately.

Modified non-DP algorithm achieving the same approximation ratio. To address the challenges discussed above we first propose a modification to the strategy given by [17]. While the modified algorithm is still non-private it still achieves a $(1+\rho)$-approximation for any $\rho > 0$ and is amenable to differentially private adaptations. Our algorithm now samples vertices S *without replacement* and puts them into buckets $S_i = B_i \cap S$ according to their degrees. The key modification is that we merge all of the buckets with smaller degrees i.e., $i \leqslant K$[6] into one. We redefine B_1 to denote this merged bucket and $S_1 = S \cap B_1$ and we redefine I to be the set of all indices $i > K$ such that $|S_i|$ is sufficiently large. If B_1 is not too large then all of the edges incident to B_1 can simply be ignored as the total number of these edges will be small. Otherwise, we can account for edges that are incident to B_1 by adding $\frac{1}{|S|}\sum_{v\in S_1}(1+X(v))\deg(v)$ to our final output. Since we merged all of the buckets with smaller degrees we no longer have the guarantee that $\deg(u) \approx \deg(v)$ for all $u, v \in S_1$. However, since $\deg(v)$ is reasonably small for each $v \in S_1$ the variance is still manageable. Intuitively, the sum $\frac{1}{|S|}\sum_{v\in S_1}(1+X(v))\deg(v)$ approximates $\frac{1}{n}\sum_{v\in B_1}(1+\alpha_v)\deg(v)$ where α_v now denotes the fraction of edges incident to v whose second endpoint lies outside the set $B_1 \cup \bigcup_{i\in I} B_i$.

The differentially-private modified algorithm. We now introduce our sublinear-time differentially-private algorithm to approximate the average degree in Algorithm 4. Algorithm 4 relies on three subroutines given by Algorithms 1, 2, and 3. Splitting the algorithm into separate modules simplifies the privacy analysis as we can show that each subroutine is $\varepsilon/3$-differentially private – it follows that the entire algorithm is ε-differentially private. In Algorithm 1 we add Laplace noise to the degrees of *all* vertices in the graph and then return a sample of vertices, say S (sampled uniformly without replacement) along with their noisy degrees. For simplicity we describe Algorithm 1 in a way that the running time is linear in the size of the input. We do this to make our privacy analysis simpler. However, we can implement Algorithm 1 with lazy sampling of Laplace noise Y_u when required i.e., if node u is in our sample S or if $u = r(v)$ was the randomly selected neighbor of some node $v \in S$.

Algorithm 1 **NoisyDegree**.

NoisyDegree takes G as input and returns a set of sampled vertices along with the noisy degrees of every vertex in G.

1. Uniformly and independently select $\Theta(\sqrt{n}\cdot\mathrm{poly}(\log(n)/\rho)\cdot\mathrm{poly}(1/\varepsilon))$ vertices (without replacement) from V and let S denote the set of selected vertices.
2. For every $v \in V(G)$,

$$\tilde{d}(v) = \deg(v) + Y_v\ ,$$

   where $Y_v \sim \mathrm{Lap}(6/\varepsilon)$.
3. Return $\{\tilde{d}(v)\}_{v\in V(G)}, S$

Given the output of Algorithm 1 we can partition the sample S into buckets $\tilde{S}_i = S \cap \tilde{B}_i$ using their noisy degree. Here, we define $\tilde{B}_i = \left\{v : \tilde{d}(v) \in \left((1+\beta)^{i-1}, (1+\beta)^i\right)\right\}$ and we also define a merged bucket $S_1 = S \cap \left\{v : \tilde{d}(v) \leqslant (1+\beta)^{K-1}\right\}$ containing all sampled nodes

[6] where we fix $K := \left(2 + \log_{1+\beta}\left(\frac{2|S|\sqrt{\rho}}{\beta \log_{1+\beta}(n)\sqrt{n\sqrt{\log n}}}\right)\right)$ in the sequel

with noisy degree at most $(1+\beta)^{K-1}$. Here, K is a degree threshold parameter that we can tune. Now given a size threshold parameter T we can define $I = \left\{i \geqslant K \ : \ \left|\tilde{S}_i\right| \geqslant 1.2T \cdot |S|\right\}$ to be the set of big buckets. We remark that as a special case we define $|S_1|$ to be "small" if $|S_1| < 1.2T \cdot \sqrt{|S|} \cdot |S|$ instead of $|S_1| < 1.2T \cdot |S|$. As an intuitive justification we note that (whp) for each node v with noisy degree $\tilde{d}(v) \leqslant (1+\beta)^{K-1}$ the actual degree $\deg(v)$ will not be too much larger than $(1+\beta)^{K-1}$. In this case we have $\sum_{v:\tilde{d}(v)\leqslant(1+\beta)^{K-1}} \deg(v) \leqslant |S_1| \max_{v:\tilde{d}(v)\leqslant(1+\beta)^{K-1}} \deg(v) = o(n)$ so that we can safely ignore the edges incident to S_1.

Intuitively, for each large bucket $i \in I$, Algorithm 2 computes $\tilde{\alpha}_i = W_i/|\tilde{S}_i|$ our approximation of the fraction of type (2) edges incident to $\tilde{B}_i$. If S_1 is large then type (2) edges are (re)defined to be the edges with exactly one endpoint in $\left\{v : \tilde{d}(v) \leqslant (1+\beta)^{K-1}\right\} \cup \bigcup_{i\in I} \tilde{B}_i$. To preserve differential privacy we add laplace noise to W_i i.e., $W_i = Z_i + \sum_{v\in\tilde{S}_i} X(v)$ where $Z_i \sim \mathsf{Lap}(6/\varepsilon)$. We remark that (whp) we will have $Z_i = o(|\tilde{S}_i|)$ for each large bucket $i \in I$. Thus, the addition of laplace noise will have a minimal impact on the accuracy of the final result.

Algorithm 2 NoisyBigSmallEdgeCount. Here $M_{\rho,n}$ is a degree threshold parameter and T is a size threshold parameter. Note that the relationship between the parameters K (used informally as a degree threshold parameter in the overview) and $M_{\rho,n}$ is $K = 2 + \log_{(1+\beta)}\lceil 6M_{\rho,n}/\beta\rceil$.

NoisyBigSmallEdgeCount takes as input $G, I, \{\tilde{S}_i\}_{i=1}^t, S_1, \{\tilde{d}(v)\}_{v\in V(G)}, M_{\rho,n}, T$ and returns an approximation of the fraction of edges that are between big buckets and small buckets.

1. For every $i \in I$, ▷ count the edges between buckets in I and small buckets
   a. For all $v \in \tilde{S}_i$,
      i. Pick a random neighbor of v, say $r(v)$.
      ii. If $|S_1| < 1.2T\cdot\sqrt{|S|}\cdot|S|$, i.e., if S_1 is a small bucket. Then if $\tilde{d}(r(v)) \in ((1+\beta)^{i-1}, (1+\beta)^i]$ for some $i \notin I$, then $X(v) = 1$, otherwise $X(v) = 0$.
      iii. Otherwise, S_1 is not small. Therefore, if $\tilde{d}(r(v)) \in ((1+\beta)^{i-1}, (1+\beta)^i]$ for some $i \notin I$ and $i > \log_{1+\beta}\lceil\left(\frac{6M_{\rho,n}}{\beta}\right)\rceil + 2$, then $X(v) = 1$, otherwise $X(v) = 0$.
   b. Define $W_i := \sum_{v\in\tilde{S}_i} X(v) + Z_i$ where $Z_i \sim \mathrm{Lap}(6/\varepsilon)$ and $\tilde{\alpha}_i := \frac{W_i}{|\tilde{S}_i|}$.
2. return $\{W_i\}_{i\in I}, \{\tilde{\alpha}_i\}_{i\in I}$

If the merged bucket S_1 is small then we can ignore edges incident to S_1 and Algorithm 3 will simply output $\frac{1}{|S|}\sum_{i\in I} |\tilde{S}_i| \cdot (1+\tilde{\alpha}_i) \cdot (1+\beta)^i$. In this case the output can be computed entirely from the differentially private outputs that have already been computed by Algorithms 1 and 2 without even looking at the graph G. Intuitively, for any large bucket $i \in I$ and $v \in \tilde{S}_i$ we expect that (whp) $|Y_v| = |\tilde{d}(v) - \deg(v)|$ is small enough to ensure that $(1+\beta)^{i-2} \leqslant \deg(v) \leqslant (1+\beta)^{i+1}$. Thus, $(1+\beta)^i$ is still a reasonable approximation for $\deg(v)$.

If the merged bucket S_1 is sufficiently large, then we need to account for the edges within S_1 itself as well as the fraction of edges between S_1 and small buckets. We introduce a new estimator to approximate the fraction of edges between S_1 and small buckets given by $Z + \sum_{v\in S_1}(1+X(v))\cdot\deg'(v)$ where $Z \sim \mathrm{Lap}\left(36M_{\rho,n}\left(3+\beta+\frac{1}{\beta}\right)\right)$ and $\deg'(v) = \min\{\deg(v), 6M_{\rho,n}\left(3+\beta+\frac{1}{\beta}\right)\}$ (See Algorithm 3) – the relationship between the parameters K and $M_{\rho,n}$ is $K = 2 + \log_{(1+\beta)}\lceil 6M_{\rho,n}/\beta\rceil$. The Laplace Noise term is added to preserve differential privacy. We define the clamped degrees $\deg'(v)$ to ensure that the coupled global sensitivity of the randomized subroutine computing $\sum_{v\in S_1}(1+X(v))\cdot\deg'(v)$ is upper

bounded by $12M_{\rho,n}\left(3+\beta+\frac{1}{\beta}\right)$. This way we can control the laplace noise parameters to ensure that $Z = o(|S_1|)$ with high probability so that the noise term Z does not adversely impact accuracy. Intuitively, we expect that $Y_v \leqslant M_{\rho,n}$ for all nodes v with high probability. In this case for any node $v \in S_1$ we will have $\deg'(v) = \deg(v) \leqslant 6M_{\rho,n}\left(3+\beta+\frac{1}{\beta}\right)$.

Algorithm 3 **NoisyAvgDegree**.

NoisyAvgDegree takes $\{\tilde{S}_i\}_{i=1}^t, \{\tilde{d}(v)\}_{v\in V(G)}, \{\tilde{\alpha}_i\}_{i\in I}, I, M_{\rho,n}, T$ as input and returns the noisy estimator for average degree of the graph.

1. If $|S_1| < 1.2T\cdot\sqrt{|S|}\cdot|S|$ then output $\frac{1}{|S|}\sum_{i\in I}|\tilde{S}_i|\cdot(1+\tilde{\alpha}_i)\cdot(1+\beta)^i$.
2. Else, for every $v \in S_1$,
   a. Pick a random neighbor of v, say $r(v)$.
   b. If $\tilde{d}(r(v)) \in ((1+\beta)^{i-1}, (1+\beta)^i]$ for some $i \notin I$ and $i > \log_{1+\beta}\lceil\left(\frac{6M_{\rho,n}}{\beta}\right)\rceil + 2$, then $X(v) = 1$, otherwise $X(v) = 0$.
   c. Output

$$\frac{1}{|S|}\left(\sum_{i\in I}|\tilde{S}_i|\cdot(1+\tilde{\alpha}_i)\cdot(1+\beta)^i + Z + \sum_{v\in S_1}(1+X(v))\cdot\deg'(v)\right),$$

where

$$Z \sim \mathrm{Lap}\left(36M_{\rho,n}\left(3+\beta+\frac{1}{\beta}\right)\right)$$

and

$$\deg'(v) = \min\left\{\deg(v), 6M_{\rho,n}\left(3+\beta+\frac{1}{\beta}\right)\right\}$$

Algorithm 4 **Main DP Algorithm**.

Main DP Algorithm that takes graph G as input and outputs an approximation of its average degree.

1. $\{\tilde{d}(v)\}_{v\in V(G)}, S :=$ **NoisyDegree**(G) ▷ see Algorithm 1
2. For $i = 1, 2\ldots, t$, let $\tilde{S}_i = \{v \in S : \tilde{d}(v) \in ((1+\beta)^{i-1}, (1+\beta)^i]\}$ where $t := \lceil\log_{(1+\beta)}(n)\rceil$.
3. Define $M_{\rho,n} := \frac{1}{3}\cdot\sqrt{\frac{\rho}{n\sqrt{\log(n)}}}\cdot\frac{|S|}{t}$, $S_1 := \cup_{i\leqslant\log_{1+\beta}\left(\frac{6M_{\rho,n}}{\beta}\right)+2}\tilde{S}_i$, and, $I = \{i > \log_{1+\beta}\left(\frac{6M_{\rho,n}}{\beta}\right)+2 : |\tilde{S}_i| \geqslant 1.2T\cdot|S|\}$ where $T := \frac{1}{2}\sqrt{\frac{\rho}{n}}\cdot\frac{\varepsilon}{(1+\varepsilon)}\cdot\frac{1}{t}$.
4. $\{W_i\}_{i\in I}, \{\tilde{\alpha}_i\}_{i\in I} :=$ **NoisyBigSmallEdgeCount**$(G, I, \{\tilde{S}_i\}_{i=1}^t)$. ▷ see Algorithm 2
5. **NoisyAvgDegree**$(G, S, \{\tilde{S}_i\}_{i=1}^t, \{\tilde{\alpha}_i\}_{i\in I}, I, M_{\rho,n}, T)$. ▷ see Algorithm 3

The full analysis of Theorem 2 can be found in Sections 2 and 4.

▶ Remark 11. A simpler algorithm for estimating the average degree was given by Seshadri [31]. The main intuition behind this algorithm is that out of m edges of a graph, there are not "too many" edges that contribute a high degree. Thus the algorithm samples vertices and a random neighbor of each sampled vertex, but it only counts edges (scaled by a factor of 2 times the degree of the sampled vertex) for which the degree of the random neighbor is higher than that of the degree of the sampled vertex.

The Coupled Global Sensitivity of the final estimate returned by this algorithm is high (proportional to the degree of the sampled vertex and its random neighbor); thus adding Laplace noise directly to the estimate would result in a very inaccurate algorithm. It is unclear how to mitigate this issue and make this algorithm differentially-private with a reasonable accuracy guarantee.

1.4.2 Privately Estimating Maximum Matching and Vertex Cover Size

At a high-level our private algorithms for estimating the maximum matching and vertex cover add laplace noise (to the outputs) proportional to the coupled global sensitivity of the randomized non-private algorithms for the corresponding problems. The challenge lies in proving the coupled global sensitivity of these non-private algorithms is small.

We first describe and analyze the coupled global sensitivity of the classical polynomial-time greedy matching algorithm. This is helpful in our analysis of the non-private sublinear-time algorithm for maximum matching in the sequel.

We then describe and give a proof sketch of the coupled global sensitivity of the non-private sublinear-time matching algorithm [3]. The formal proofs for privately estimating the maximum matching and minimum vertex cover size are in the full version [5]. Recall that d is the maximum degree and $\bar{d}$ is the average degree of the graph .

The Polynomial-time Greedy Matching Algorithm $\mathcal{A}_{MM}$. This algorithm takes as input a graph $G = (V, E)$ and a random permutation π on the set of pairs $(x, y) \in V \times V$, with $x \neq y$, and processes each pair of vertices (x, y) in the increasing order of ranks given by π, and greedily adds edges to a maximal matching whose size is finally output[7]. Since the size of the maximal matching produced is known to be at least $\frac{1}{2}$ of the size of a maximum matching, this gives a non-private 2-approximation of the size of a maximum matching in G.

CGS of the Greedy Algorithm $\mathcal{A}_{MM}$. We show that the CGS of the greedy algorithm (with respect to node-neighboring graphs) is at most 1. Note that once the ranking on the edges is fixed the maximal matching obtained by $\mathcal{A}_{MM}$ is also fixed. Let σ_I be the identity permutation over the ranking of edges, i.e., we have $\sigma_I(\pi) = \pi$. We use Fact 9 to observe that,

$$\begin{aligned}
\mathsf{CGS}_{\mathcal{A}_{MM}} &\leqslant \max_{G_1 \sim G_2} \min_{\sigma} \max_{\pi} |\mathcal{A}_{MM}(G_1; \pi) - \mathcal{A}_{MM}(G_2; \sigma(\pi))| \\
&\leqslant \max_{\substack{G_1 \sim G_2 \\ \pi}} |\mathcal{A}_{MM}(G_1; \pi) - \mathcal{A}_{MM}(G_2; \sigma_I(\pi))| \\
&= \max_{\substack{G_1 \sim G_2 \\ \pi}} |\mathcal{A}_{MM}(G_1; \pi) - \mathcal{A}_{MM}(G_2; \pi).|
\end{aligned}$$

Therefore it is sufficient to analyze the relative size of the matching obtained on node-neighboring graphs G_1, G_2 that are processed by the greedy algorithm in the order given by the same π.

Let $G_1 \sim G_2$ where v^* is such that $E(V_1 \setminus \{v^*\}) = E(V_2 \setminus \{v^*\})$. Denote the greedy matchings obtained from $\mathcal{A}_{MM}(G_1, \pi)$ as M_1 and from $\mathcal{A}_{MM}(G_2, \pi)$ as M_2. Suppose edge e^* is incident to v^* such that $e^* \in E_2$, and $e^* \notin E_1$. We will show that $||M_1| - |M_2|| \leqslant 1$, which implies that $\max_{\substack{G_1 \sim G_2 \\ \pi}} |\mathcal{A}_{MM}(G_1; \pi) - \mathcal{A}_{MM}(G_2; \pi)| \leqslant 1$, thus proving that $\mathsf{CGS}_{\mathcal{A}_{MM}} \leqslant 1$.

[7] We note that the non-private algorithms [24, 35, 26] only consider the ranking π over m edges of the graph, whereas we consider the ranking over all $\binom{n}{2}$ pairs of vertices. This is because we want to define a "global" ranking so that we can define the same ranking consistently over neighboring graphs that may have different edges.

We first claim that if $e^* \notin M_1 \cup M_2$ then $|M_1| = |M_2|$. Since the greedy algorithm considers edges in the same order, the exact same edges must have been placed in M_1 as in M_2 before $e*$ is processed. Since $e^* = (v^*, u)$ is not chosen in M_2 it must have been the case that by this time u was matched in M_2, and thus the same matched edge must occur in M_1. From here on the algorithm again must make the same choices for the edges to be placed in M_1 and M_2.

Next, we claim that if $e^* \in M_1 \cup M_2$ then $M := M_1 \oplus M_2$[8] is one connected component containing e^*. Consequently, $||M_1| - |M_2|| \leqslant 1$. Since $e^* \in M_2$ and e^* cannot be in M_1, it is clear that $e^* \in M$. Suppose for the sake of contradiction, M consists of two connected components C_1, C_2 and WLOG $e^* \in C_1$. Consider edges in C_2. By Berge's Lemma [34], C_2 is either an alternating path or an alternating even cycle, with alternating edges from M_1 and M_2. Also, the edges in C_2 exist in both G_1 and G_2 with the same ranking. Observe that since C_2 is separate from C_1 containing e^*, if we replace edges in C_2 belonging to M_2 in the original graph G_2 by edges in C_2 belonging to M_1, this is still a valid maximal matching for the graph G_2. In fact, the greedy algorithm considers edges in C_2 in the same order for both graphs G_1, G_2, so the edges in M_1 and M_2 should be the same, in other words, C_2 cannot be a part of $M = M_1 \oplus M_2$, and hence M must have only one connected component, which contains e^*. Now, since M is either an alternating path or even cycle, $||M_1| - |M_2|| \leqslant 1$.

The Local Maximum Matching Algorithm $\mathcal{A}_{\mathbf{sub-MM}}$. We describe the local algorithm implemented by [3] in Algorithm 5. We modify the original algorithm to sample vertices *without replacement*. The algorithm then calls the vertex cover oracle (denoted as $\mathcal{O}^{\pi}_{VC}$) on each sampled vertex which subsequently calls the maximal matching oracle (denoted as $\mathcal{O}^{\pi}_{MO}$) on the incident edges to determine whether the sampled vertex is in the matching fixed by the ranking of edges π. Finally, the algorithm returns an estimate of the maximum matching size based on the number of sampled vertices in the matching. We note that in [3] the same sampling algorithm simultaneously outputs an approximation to maximum matching size and minimum vertex cover size. We choose to write the sampling procedure for estimating the maximum matching size and minimum vertex cover size separately so that it is easier to understand the Coupled Global Sensitivity for outputting the two different estimators.

Algorithm 5 Local Maximum Matching algorithm $\mathcal{A}_{sub-MM}$ using Oracle access.

Input. Input Graph $G = (V, E)$.
1. Uniformly sample $s = 16 \cdot 24(\ln n)/\rho^2$ vertices from V without replacement.
2. For $i = 1 \dots s$, if $\mathcal{O}^{\pi}_{VC}(v_i) =$ True then let $X_i = 1$, otherwise let $X_i = 0$.
3. return $\tilde{M} = \frac{n}{2s}(\sum_{i \in [s]} X_i) - \frac{\rho n}{2}$.

▶ Remark 12. [3] gives an efficient simulation of the matching and vertex cover oracles which exposes edges incident to a vertex in batches only when they are needed. We assume the efficient simulation of these oracles in our algorithms.

CGS of the Local Matching Algorithm $\mathcal{A}_{\mathbf{sub-MM}}$. Our main techniques involve identifying the sources of randomness in the local algorithm itself and then coupling the random coins of the runs of the algorithm on neighboring graphs. We follow the local algorithm given by [3] which samples vertices for both matching and vertex cover size estimation. We show that the identity coupling is sufficient in this case.

[8] $M_1 \oplus M_2$ is the *symmetric difference* of sets and this is defined as the set of edges in either M_1 or M_2 but not in their intersection.

In a previous version of our paper (before we were aware of the results of [3]), we analyzed the Coupled Global Sensitivity of the local matching algorithm given by [24, 35] which samples a set of edges uniformly at random, and calls a matching oracle on each sampled edge. The matching oracle indicates whether the edge is in the greedy matching fixed by the ranking π or not. Analyzing the Coupled Global Sensitivity of this algorithm is more challenging, i.e., considering the identity permutation σ_I over the ranking of edges π and sampled edges does not work. This is because for node-neighboring graphs G_1, G_2, it could be the case that all the edges sampled from G_1 belong to the matching M_1 fixed by the ranking π, but the same edges sampled from G_2 may not be in the matching M_2 fixed by the ranking π. Thus, we need to carefully define a bijection that maps edges in the matching M_1 to edges in the matching M_2.

2 Privacy Analysis of Theorem 2

▶ **Theorem 13.** *The Algorithm 4 is ε-DP.*

Proof. We will approach the privacy analysis in a modular fashion, i.e., we will analyze each sub-routine separately and show that by composition, the entire algorithm is ε-differentially private.

In the sequel, when analyzing the coupled global sensitivity of intermediate randomized quantities, we use Fact 9.

▷ Claim 14. Algorithm NoisyDegree (see Algorithm 1) is $\varepsilon/3$-DP.

Proof. First, fix any sample S. Define the function $f_{noisy-deg} := \{\tilde{d}(v)\}_{v\in V(G)}$. Observe that the degree of a node can change by at most 1 from adding or deleting an edge, and therefore $f_{noisy-deg}$ changes by at most 2 by adding or deleting an edge, in other words, the $GS_{f_{noisy-deg}} = 2$ and we can add noise proportional to $2/\varepsilon$. ◁

▷ Claim 15. Algorithm NoisyBigSmallEdgeCount (Algorithm 2) is $\varepsilon/3$-DP.

Proof. We fix noisy degrees $\{\tilde{d}(v)\}_{v\in V(G)}$, consequently fixing the buckets $\tilde{S}_1, \ldots, \tilde{S}_t$ and set I. Define the function $f_{t,\tilde{d}} := \{f_{\tilde{S}_i,\tilde{d}}(G; r)\}_{i\in I}$, and the function $f_{\tilde{S}_i,\tilde{d}}(G; r) = \sum_{v\in\tilde{S}_i} H(r(v))$ where $H(w) = 1$ if and only if we have $\tilde{d}(w) \in ((1+\beta)^{i-1}, (1+\beta)^i]$ for some $i \notin I$ and $|S_1| < 1.2T \cdot \sqrt{|S|} \cdot |S|$ or if $\tilde{d}(w) \in ((1+\beta)^{i-1}, (1+\beta)^i]$ for some $i \notin I$ and $i > \log_{1+\beta}\lceil\left(\frac{6M_{\rho,n}}{\beta}\right)\rceil + 2$; here $r(\cdot)$ defines the random coins used to sample a neighbor of v. We analyze $CGS_{f_{\tilde{S}_i,\tilde{d}}}$, and argue that $CGS_{f_{t,\tilde{d}}} \leqslant CGS_{f_{\tilde{S}_i,\tilde{d}}}$.

First, we show that for all fixed S, $\{\tilde{d}(v)\}_{v\in S}$ and $i \in I$, the $CGS_{f_{S_i,\tilde{d}}}$ is at most 2. Consider G and G' such that edge $(u^*, v^*) \in G$, but does not exist in G'. Fix any coupling such that $r(w) = r'(w)$ for all $w \neq u^*, v^*$, where r, r' defines the random coins for sampling neighbors of w in G and G' respectively. Now we have $X(w) = H(r(w)) = H(r'(w)) = X'(w)$ for all $w \neq u^*, v^*$. Thus, $CGS_{f_{\tilde{S}_i,\tilde{d}}} = |f_{\tilde{S}_i,\tilde{d}}(G; r) - f_{\tilde{S}_i,\tilde{d}}(G'; r')| = |\sum_{v\in\tilde{S}_i} H(r(v)) - \sum_{v\in\tilde{S}_i} H(r'(v))| = |H(r(v^*)) + H(r(u^*)) - H(r'(v^*)) - H(r'(u^*))| \leqslant 2$. Now, since the differing endpoints u^*, v^* can only appear in at most one of the i-th iterations simultaneously, it is clear to see that $CGS_{f_{t,\tilde{d}}}$ is also at most 2. ◁

▷ Claim 16. Algorithm NoisyAvgDegree (Algorithm 3) is $\varepsilon/3$-DP.

Proof. We fix noisy degrees $\{\tilde{d}(v)\}_{v\in V(G)}$, and sample S consequently fixing the buckets $\tilde{S}_1, \ldots, \tilde{S}_t$ and set I, and we fix $\{\tilde{\alpha}_i\}_{i=1}^t$. Note that the first output in Line 1 given by $\frac{1}{|S|}\sum_{i\in I} |\tilde{S}_i| \cdot (1+\tilde{\alpha}_i) \cdot (1+\beta)^i$ is already private since the terms in the summation consist of

parameters that are either noisy or public or both. We need to show that the second output in Line 2c is private. In particular, define the function $f_{S_1,\tilde{d}}(G;r) := \sum_{v\in S_1}(1+H_1(r(v)))\cdot\deg'(v)$ where $\deg'(v) = \min\{\deg(v), 6M_{\rho,n}\left(3+\beta+\frac{1}{\beta}\right)\}$ and $H_1(w) = 1$ if and only if $\tilde{d}(w) \in ((1+\beta)^{i-1},(1+\beta)^i]$ for some $i \notin I$ and $i > \log_{1+\beta}\lceil\left(\frac{6M_{\rho,n}}{\beta}\right)\rceil+2$. We claim that for all fixed S and $\{\tilde{d}(v)\}_{v\in S}$, the $CGS_{f_{S_1,\tilde{d}}}$ is at most $12M_{\rho,n}\left(3+\beta+\frac{1}{\beta}\right)$. Consider G and G' such that edge $(u^*,v^*)\in G$, but does not exist in G'. Fix any coupling such that $r(w) = r'(w)$ for all $w \neq u^*, v^*$, where r, r' defines the random coins for sampling neighbors of w in G and G' respectively. Now we have $X(w) = H_1(r(w)) = H_1(r'(w)) = X'(w)$ for all $w \neq u^*, v^*$. Thus, $|f_{S_1,\tilde{d}}(G;r) - f_{S_1,\tilde{d}}(G';r')| = |\sum_{v\in\tilde{S}_1}(1+H_1(r(v)))\cdot\deg'(v) - \sum_{v\in\tilde{S}_1}(1+H_1(r'(v)))\cdot\deg'(v)| = |(1+H(r(v^*)))\cdot\deg'(v^*) + (1+H(r(u^*)))\cdot\deg'(u^*) - (1+H(r'(v^*)))\cdot\deg'(v^*) - (1+H(r'(u^*)))\deg'(u^*)| \leqslant 2\cdot 6M_{\rho,n}\left(3+\beta+\frac{1}{\beta}\right) = 12M_{\rho,n}\left(3+\beta+\frac{1}{\beta}\right)$. Note that we introduce $\deg'(v)$, to ensure that the sensitivity of $f_{S_1,\tilde{d}}$ remains small. ◁

By composition, we have that the main algorithm is ε-DP. ◀

3 Preliminaries

We state the following tail bound for a random variable drawn from the Laplace Distribution.

▶ **Fact 17.** *If* $Y \sim \mathrm{Lap}(b)$*, then*

$$\Pr[|Y| \geqslant \ell\cdot b] = \exp(-\ell)\ .$$

Next, we state a well-known fact which implies that the concentration results for sampling with replacement obtained using Chernoff bounds type methods (bounding moment generating function + Markov inequality) can be transferred to the case of sampling without replacement.

▶ **Fact 18** ([2, 20])**.** *Let* $\mathcal{X} = (x_1,\ldots,x_N)$ *be a finite population of* N *points and* $X_1,\ldots,X_n$ *be a random sample drawn without replacement from* $\mathcal{X}$*, and* $Y_1,\ldots,Y_n$ *be a random sample drawn with replacement from* $\mathcal{X}$*. If* $f:\mathbb{R}\to\mathbb{R}$ *is continuous and convex, then*

$$\mathbb{E}\left[f\left(\sum_{i=1}^{n}X_i\right)\right] \leqslant \mathbb{E}\left[f\left(\sum_{i=1}^{n}Y_i\right)\right]\ .$$

4 Accuracy Analysis of Theorem 2

4.1 Proof Sketch of Theorem 2

In this section, we give a sketch of the accuracy analysis. The more formal proofs can be found in the full version [5].

▶ **Theorem 19.** *For every* $\rho < 1/4$*,* $\beta \leqslant \rho/8$*, and* $\varepsilon^{-1} = o(\log^{1/4}(n))$*, for sufficiently large* n*, the main algorithm (see Algorithm 4) outputs a value* $\tilde{d}$ *such that with probability at least* $1-o(1)$*, it holds that*

$$(1-\rho)\cdot\bar{d} \leqslant \tilde{d} \leqslant (1+\rho)\cdot\bar{d}$$

Proof. The main proof strategy conditions on S_1 being sufficiently large or not. First, consider Case 1 when $|S_1| < 1.2T\cdot\sqrt{|S|}\cdot|S|$ where T is a size threshold parameter. We first show that for $i\in I$ the noisy buckets $|\tilde{B}_i|/n$ are approximated well by $|\tilde{S}_i|/|S|$. Next we show

that the number of vertices in buckets that are significantly smaller than the size threshold are of size $O(\sqrt{n})$ (for buckets $U' := \{v \in \tilde{B}_i : (i \notin I) \wedge (i > \log_{1+\beta}\left(\frac{6M_{\rho,n}}{\beta}\right) + 2)\}$, and of size $\tilde{O}(n^{3/4})$ (for bucket $B_1 := \cup_{i<\log_{1+\beta}\left(\frac{6M_{\rho,n}}{\beta}\right)+2}\tilde{B}_i$. This leads to the corollary (see full version [5] for the formal statement) which bounds the number of edges between small buckets as roughly $\tilde{O}(\rho n + n^{3/4})$.

One of our main contributions is showing that the actual fraction of edges between sufficiently large buckets and small buckets, denoted by α_i, is approximated well by our noisy estimator $\tilde{\alpha}_i$.

▶ **Corollary 20.** *Assuming that* $\varepsilon^{-1} = o(\log^{1/4}(n))$*, for every* $i \in I$*, for sufficiently large* n*, we have that with probability at least* $1 - o(1)$*,*

1. $|\tilde{\alpha}_i - \alpha_i| \leqslant \frac{\rho}{4}\alpha_i$ *if* $\alpha_i \geqslant \rho/8$.
2. $\tilde{\alpha}_i \leqslant \rho/4$*, if* $\alpha_i \leqslant \rho/8$.

Finally, we need to show that for sufficiently large noisy buckets, the actual degrees of the vertices (sans noise) only shifts to an adjacent noisy bucket. This helps us bound the number of edges whose one endpoint resides in a sufficiently large noisy bucket. We have shown that with high probability, all approximations of edges between the different types of buckets is good, which leads to the main Lemma for Case 1.

Now consider Case 2 when $|S_1| > 1.2T \cdot \sqrt{|S|} \cdot |S|$. We show that the bucket $|B_1|/n$ is now approximated well by $|S_1|/|S|$. We introduce a different estimator for counting edges between B_1 and small buckets given by $Z + \sum_{v \in S_1}(1 + X(v)) \cdot \deg'(v)$, where $Z \sim \text{Lap}\left(36M_{\rho,n}\left(3 + \beta + \frac{1}{\beta}\right)\right)$ and $\deg'(v) = \min\{\deg(v), 6M_{\rho,n}\left(3 + \beta + \frac{1}{\beta}\right)\}$. First, we show that for every $v \in S_1$, with high probability $\deg'(v) = \deg(v)$. Our main contribution in this case is showing that our estimator (sans noise) approximates the fraction of the sum of the edges between B_1 and all vertices in the graph (denoted by E_1), and the edges between B_1 and vertices in small buckets in the graph (denoted by E_1') well (see lemma below).

▶ **Lemma 21.** *Let* $\bar{d}_1$ *be the average degree of bucket* B_1*. If* $|B_1| > 1.5T \cdot \sqrt{|S|} \cdot n$*,*

1. *If* $\bar{d}_1 \geqslant 1$*, then with probability at least* $1 - o(1)$*,*

$$\left(1 - \frac{\rho}{4}\right) \cdot \frac{|E_1| + |E_1'|}{n} < \frac{1}{|S|}\sum_{v \in S_1}(1 + X(v)) \cdot \deg(v) < \left(1 + \frac{\rho}{4}\right) \cdot \frac{|E_1| + |E_1'|}{n}$$

2. *If* $\bar{d}_1 < 1$*, and* $\bar{d} \geqslant 1$*, then with probability at least* $1 - o(1)$*,*

$$\frac{|E_1| + |E_1'|}{n} - \rho/4 < \frac{1}{|S|}\sum_{v \in S_1}(1 + X(v)) \cdot \deg(v) < \frac{|E_1| + |E_1'|}{n} + \rho/4$$

To complete this part of the proof, we show that the noise added to the estimator (denoted by Z) is small and therefore, the noisy estimator also approximates the quantity $(|E_1| + |E_1'|)/n$ well.

The rest of the analysis is similar to Case 1 and we invoke the same lemmas to show that with high probability, the approximations of edges between the rest of the sufficiently large buckets, and between the small buckets, as well as between the sufficiently large buckets and small buckets is good, thus giving us the main Lemma for Case 2.

Combining these two main lemmas proves our main theorem statement. ◀

5 Conclusions and open questions

In this work we give a differentially-private sublinear-time $(1+\rho)$-approximation algorithm for estimating the average degree of the graph. We achieve a running time comparable to its non-private counterpart, which is also tight in terms of its asymptotic behaviour with respect to the number of vertices of the graph. We also give the first differentially-private approximation algorithms for the problems of estimating maximum matching size and vertex cover size of a graph.

To analyze the privacy of our algorithms, we proposed the notion of coupled global sensitivity, as a generalization of global sensitivity, which is applicable to randomized approximation algorithms. We show that coupled global sensitivity implies differential privacy, and use it to show that previous non-private algorithms from the literature, or variants, can be made private by finely tuning the amounts of noise added in various steps of the algorithms.

We propose several directions of investigation for developing the notion of coupled global sensivity further and open problems pertaining to differentially-private sublinear-time algorithms for graphs.

Other applications and limitations of CGS. In particular, what are the limitations of the CGS method? Can we characterize the set of algorithms with small CGS? Are there other natural problems for which we already have algorithms with small CGS, and hence that are easily amenable to privacy analogues? Are there algorithms for which we can prove large lower bounds on the CGS and yet they provide differential privacy?

Better approximations for maximum matching problems. In [24, 35], the authors also give a $(1, \rho n)$-approximation of maximum matching size with a query complexity that is exponential in d. Their analysis involves iterating over a sequence of oracles to augment paths of small length, in increasing order of lengths. The matching oracle considered in this work is used only in the first iteration. Analyzing the coupled global sensitivity of that algorithm appears to be much more involved, and we leave it as an open problem.

Better time complexity guarantees for $(2, \rho n)$-approximation matching and vertex cover algorithms. Note that our results in Theorems 3 and 4 achieve an expected running time. In contrast, the results in [3] achieve a high-probability bound on the time-complexity. This can be done by running multiple instances of the resulting approximation algorithm for enough time and returning the output of the instance that terminates first (the analysis involves a simple application of Markov inequality). Achieving this step in a way that preserves privacy would result in a degradation of the privacy parameter ε, due to composition. We leave it as an open question to provide a tighter privacy vs time-complexity analysis.

References

1 Daniel Alabi, Audra McMillan, Jayshree Sarathy, Adam D. Smith, and Salil P. Vadhan. Differentially private simple linear regression. *CoRR*, abs/2007.05157, 2020. `arXiv:2007.05157`.

2 Rémi Bardenet and Odalric-Ambrym Maillard. Concentration inequalities for sampling without replacement. *Bernoulli*, 21(3):1361–1385, 2015.

3 Soheil Behnezhad. Time-optimal sublinear algorithms for matching and vertex cover. *CoRR*, abs/2106.02942, 2021. `arXiv:2106.02942`.

4 Jeremiah Blocki, Avrim Blum, Anupam Datta, and Or Sheffet. Differentially private data analysis of social networks via restricted sensitivity. In Robert D. Kleinberg, editor, *Innovations in Theoretical Computer Science, ITCS '13, Berkeley, CA, USA, January 9-12, 2013*, pages 87–96. ACM, 2013. doi:10.1145/2422436.2422449.

5 Jeremiah Blocki, Elena Grigorescu, and Tamalika Mukherjee. Privately estimating graph parameters in sublinear time. *CoRR*, abs/2202.05776, 2022. arXiv:2202.05776.

6 Christian Borgs, Jennifer T. Chayes, and Adam D. Smith. Private graphon estimation for sparse graphs. In Corinna Cortes, Neil D. Lawrence, Daniel D. Lee, Masashi Sugiyama, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pages 1369–1377, 2015. URL: https://proceedings.neurips.cc/paper/2015/hash/7250eb93b3c18cc9daa29cf58af7a004-Abstract.html.

7 Christian Borgs, Jennifer T. Chayes, Adam D. Smith, and Ilias Zadik. Revealing network structure, confidentially: Improved rates for node-private graphon estimation. In Mikkel Thorup, editor, *59th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2018, Paris, France, October 7-9, 2018*, pages 533–543. IEEE Computer Society, 2018. doi:10.1109/FOCS.2018.00057.

8 Kamalika Chaudhuri and Staal A. Vinterbo. A stability-based validation procedure for differentially private machine learning. In Christopher J. C. Burges, Léon Bottou, Zoubin Ghahramani, and Kilian Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States*, pages 2652–2660, 2013. URL: https://proceedings.neurips.cc/paper/2013/hash/e6d8545daa42d5ced125a4bf747b3688-Abstract.html.

9 Shixi Chen and Shuigeng Zhou. Recursive mechanism: towards node differential privacy and unrestricted joins. In Kenneth A. Ross, Divesh Srivastava, and Dimitris Papadias, editors, *Proceedings of the ACM SIGMOD International Conference on Management of Data, SIGMOD 2013, New York, NY, USA, June 22-27, 2013*, pages 653–664. ACM, 2013. doi:10.1145/2463676.2465304.

10 Yu Chen, Sampath Kannan, and Sanjeev Khanna. Sublinear algorithms and lower bounds for metric TSP cost estimation. In Artur Czumaj, Anuj Dawar, and Emanuela Merelli, editors, *47th International Colloquium on Automata, Languages, and Programming, ICALP 2020, July 8-11, 2020, Saarbrücken, Germany (Virtual Conference)*, volume 168 of *LIPIcs*, pages 30:1–30:19. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020. doi:10.4230/LIPIcs.ICALP.2020.30.

11 Anirban Dasgupta, Ravi Kumar, and Tamás Sarlós. On estimating the average degree. In Chin-Wan Chung, Andrei Z. Broder, Kyuseok Shim, and Torsten Suel, editors, *23rd International World Wide Web Conference, WWW '14, Seoul, Republic of Korea, April 7-11, 2014*, pages 795–806. ACM, 2014. doi:10.1145/2566486.2568019.

12 Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. *Journal of Privacy and Confidentiality*, 7(3):17–51, May 2017.

13 Cynthia Dwork and Aaron Roth. The algorithmic foundations of differential privacy. *Found. Trends Theor. Comput. Sci.*, 9(3-4):211–407, 2014. doi:10.1561/0400000042.

14 Uriel Feige. On sums of independent random variables with unbounded variance and estimating the average degree in a graph. *SIAM J. Comput.*, 35(4):964–984, 2006. doi:10.1137/S0097539704447304.

15 Hendrik Fichtenberger, Monika Henzinger, and Wolfgang Ost. Differentially private algorithms for graphs under continual observation. In Petra Mutzel, Rasmus Pagh, and Grzegorz Herman, editors, *29th Annual European Symposium on Algorithms, ESA 2021, September 6-8, 2021, Lisbon, Portugal (Virtual Conference)*, volume 204 of *LIPIcs*, pages 42:1–42:16. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021. doi:10.4230/LIPIcs.ESA.2021.42.

16 Johannes Gehrke, Edward Lui, and Rafael Pass. Towards privacy for social networks: A zero-knowledge based definition of privacy. In Yuval Ishai, editor, *Theory of Cryptography - 8th Theory of Cryptography Conference, TCC 2011, Providence, RI, USA, March 28-30, 2011. Proceedings*, volume 6597 of *Lecture Notes in Computer Science*, pages 432–449. Springer, 2011. doi:10.1007/978-3-642-19571-6_26.

17 Oded Goldreich and Dana Ron. Approximating average parameters of graphs. *Random Struct. Algorithms*, 32(4):473–493, 2008. doi:10.1002/rsa.20203.

18 Anupam Gupta, Katrina Ligett, Frank McSherry, Aaron Roth, and Kunal Talwar. Differentially private combinatorial optimization. In Moses Charikar, editor, *Proceedings of the Twenty-First Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2010, Austin, Texas, USA, January 17-19, 2010*, pages 1106–1125. SIAM, 2010. doi:10.1137/1.9781611973075.90.

19 Michael Hay, Chao Li, Gerome Miklau, and David D. Jensen. Accurate estimation of the degree distribution of private networks. In Wei Wang, Hillol Kargupta, Sanjay Ranka, Philip S. Yu, and Xindong Wu, editors, *ICDM 2009, The Ninth IEEE International Conference on Data Mining, Miami, Florida, USA, 6-9 December 2009*, pages 169–178. IEEE Computer Society, 2009. doi:10.1109/ICDM.2009.11.

20 Wassily Hoeffding. Probability inequalities for sums of bounded random variables. In *The collected works of Wassily Hoeffding*, pages 409–426. Springer, 1994.

21 Vishesh Karwa, Sofya Raskhodnikova, Adam D. Smith, and Grigory Yaroslavtsev. Private analysis of graph structure. *ACM Trans. Database Syst.*, 39(3):22:1–22:33, 2014. doi:10.1145/2611523.

22 Shiva Prasad Kasiviswanathan, Kobbi Nissim, Sofya Raskhodnikova, and Adam D. Smith. Analyzing graphs with node differential privacy. In Amit Sahai, editor, *Theory of Cryptography - 10th Theory of Cryptography Conference, TCC 2013, Tokyo, Japan, March 3-6, 2013. Proceedings*, volume 7785 of *Lecture Notes in Computer Science*, pages 457–476. Springer, 2013. doi:10.1007/978-3-642-36594-2_26.

23 Wentian Lu and Gerome Miklau. Exponential random graph estimation under differential privacy. In Sofus A. Macskassy, Claudia Perlich, Jure Leskovec, Wei Wang, and Rayid Ghani, editors, *The 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '14, New York, NY, USA - August 24 - 27, 2014*, pages 921–930. ACM, 2014. doi:10.1145/2623330.2623683.

24 Huy N. Nguyen and Krzysztof Onak. Constant-time approximation algorithms via local improvements. In *49th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2008, October 25-28, 2008, Philadelphia, PA, USA*, pages 327–336. IEEE Computer Society, 2008. doi:10.1109/FOCS.2008.81.

25 Kobbi Nissim, Sofya Raskhodnikova, and Adam D. Smith. Smooth sensitivity and sampling in private data analysis. In David S. Johnson and Uriel Feige, editors, *Proceedings of the 39th Annual ACM Symposium on Theory of Computing, San Diego, California, USA, June 11-13, 2007*, pages 75–84. ACM, 2007. doi:10.1145/1250790.1250803.

26 Krzysztof Onak, Dana Ron, Michal Rosen, and Ronitt Rubinfeld. A near-optimal sublinear-time algorithm for approximating the minimum vertex cover size. In Yuval Rabani, editor, *Proceedings of the Twenty-Third Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2012, Kyoto, Japan, January 17-19, 2012*, pages 1123–1131. SIAM, 2012. doi:10.1137/1.9781611973099.88.

27 Michal Parnas and Dana Ron. Approximating the minimum vertex cover in sublinear time and a connection to distributed algorithms. *Theor. Comput. Sci.*, 381(1-3):183–196, 2007. doi:10.1016/j.tcs.2007.04.040.

28 Sofya Raskhodnikova and Adam D. Smith. Efficient lipschitz extensions for high-dimensional graph statistics and node private degree distributions. *CoRR*, abs/1504.07912, 2015. arXiv:1504.07912.

29 Dana Ron. Sublinear-time algorithms for approximating graph parameters. In *Computing and Software Science*, volume 10000 of *Lecture Notes in Computer Science*, pages 105–122. Springer, 2019.

30 Adam Sealfon and Jonathan R. Ullman. Efficiently estimating erdos-renyi graphs with node differential privacy. *J. Priv. Confidentiality*, 11(1), 2021. doi:10.29012/jpc.745.

31 C. Seshadhri. A simpler sublinear algorithm for approximating the triangle count. *CoRR*, abs/1505.01927, 2015. arXiv:1505.01927.

32 Harry Sivasubramaniam, Haonan Li, and Xi He. Differentially private sublinear average degree approximation.

33 Shuang Song, Susan Little, Sanjay Mehta, Staal A. Vinterbo, and Kamalika Chaudhuri. Differentially private continual release of graph statistics. *CoRR*, abs/1809.02575, 2018. arXiv:1809.02575.

34 Douglas Brent West et al. *Introduction to graph theory*, volume 2. Prentice hall Upper Saddle River, 2001.

35 Yuichi Yoshida, Masaki Yamamoto, and Hiro Ito. Improved constant-time approximation algorithms for maximum matchings and other optimization problems. *SIAM J. Comput.*, 41(4):1074–1093, 2012. doi:10.1137/110828691.

36 Jun Zhang, Graham Cormode, Cecilia M. Procopiuc, Divesh Srivastava, and Xiaokui Xiao. Private release of graph statistics using ladder functions. In Timos K. Sellis, Susan B. Davidson, and Zachary G. Ives, editors, *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data, Melbourne, Victoria, Australia, May 31 - June 4, 2015*, pages 731–745. ACM, 2015. doi:10.1145/2723372.2737785.