Differentially Private Continual Release of Graph Statistics

Shuang Song*, Susan Little*, Sanjay Mehta*, Staal Vinterbo† and Kamalika Chaudhuri*

shs037@eng.ucsd.edu, slittle@ucsd.edu, srmehta@ucsd.edu, staal.vinterbo@ntnu.no, kamalika@eng.ucsd.edu

*University of California, San Diego
†Norwegian University of Science and Technology

Abstract—Motivated by understanding the dynamics of sensitive social networks over time, we consider the problem of continual release of statistics in a network that arrives online, while preserving privacy of its participants. For our privacy notion, we use differential privacy – the gold standard in privacy for statistical data analysis.

The main challenge in this problem is maintaining a good privacy-utility tradeoff; naive solutions that compose across time, as well as solutions suited to tabular data either lead to poor utility or do not directly apply. In this work, we show that if there is a publicly known upper bound on the maximum degree of any node in the entire network sequence, then we can release many common graph statistics such as degree distributions and subgraph counts continually with a better privacy-accuracy tradeoff.

Index Terms—privacy, differential privacy, graph statistics

I. INTRODUCTION

Dynamic social networks are ubiquitous models of social and economic phenomena, and analyzing them over a period of time can allow researchers to understand various aspects of human behavior. Many social networks, however, include sensitive and personal information about the people involved. Consequently, we need to design privacy-preserving algorithms that can summarize properties of dynamic social networks over time while still preserving the privacy of the participants.

As a concrete motivating example, consider data on HIV transmission collected from patients in a particular region over multiple years [1], [2], [3]. Advances in sequencing technology allow scientists to infer putative transmission links by measuring similarities between HIV sequences obtained from different patients. These links can then be resolved into transmission networks, reflecting the patterns of transmission in that population. Epidemiologists would like to study properties of these networks as they grow over time to understand how HIV propagates. Since there is considerable social stigma associated with HIV, these networks are highly sensitive information, and public release of their properties needs to ensure that privacy of the included individuals is not violated. Additionally, analyses of these networks need to happen intermittently – for example, once a year –

so that properties of the network as it evolves may be studied.

In this paper, we consider continual privacy-preserving release of graph statistics, such as degree distributions and subgraph counts, from sensitive networks where nodes and their associated edges appear over time in an online manner. For our privacy notion, we use differential privacy [4] – the gold standard in private data analysis. Differential privacy guarantees privacy by ensuring that the participation of a single person in the dataset does not change the probability of any outcome by much; this is enforced by adding enough noise to either the input data or to the output of a function computed on the data so as to obscure the private value of a single individual. Since in our applications, a node corresponds to a single person, we use node differential privacy [5], where the goal is to hide the participation of any single node.

There are two main challenges in continually releasing graph statistics with node differential privacy. The first is that node differential privacy itself is often very difficult to attain, and can only be attained in either bounded degree graphs or graphs that can be projected to be degree-bounded. The second challenge pertains to the online nature of the problem. Prior work has looked at continual release of statistics based on streaming tabular data [6], [7], [8], [9], [10], [11]; however, these works rely on the fact that in tabular data, at any time t, we only get information about the t-th individual, and not about individuals who already exist in the data. This property no longer holds in online graphs, as an incoming node may bring in new information about existing nodes in the form of connecting edges, and therefore these solutions do not directly apply.

In this work, we show that if there is a publicly known upper bound on the maximum degree of any node in the entire graph sequence, then, a *difference sequence* – namely, the sequence of differences in the statistics computed on subsequent graphs – has low sensitivity. The assumption of bounded maximum degree holds for many real networks, as many real-world graphs, such as social interaction networks, collaboration networks,

computer networks and disease transmission networks, that are scale-free with power-law degree distributions have low maximum degree. Given this assumption holds, we show in particular that the sensitivity of the entire difference sequence only depends on the publicly known upper bound, and not on the length of the sequence. This implies that we can release a private version of the difference sequence with relatively high accuracy, which can be used to continually release the target statistic with high privacy-accuracy tradeoff.

It is commonly believed that many real-world networks, such as social interaction networks, collaboration networks, computer networks and disease transmission networks are scale-free with degree distributions following a power law; such graphs have low maximum degrees.

We derive the sensitivity of the difference sequence for a number of common graph statistics, such as degree distribution, number of high degree nodes, as well as counts of fixed subgraphs. We then implement our algorithms and evaluate them on three real and two synthetic datasets against two natural baselines. Our experimental results show that the algorithm outperforms these baselines in terms of utility for these datasets over a range of privacy parameters.

A. Related Work

Since its inception [4], differential privacy has become the gold standard for private data analysis and has been used in a long line of work – see [12], [13] for surveys. Differential privacy guarantees privacy by ensuring that the participation of a single individual in a dataset does not significantly affect the probability of any outcome; this is enforced by adding enough noise to obscure the influence of a single person.

To apply differential privacy to graph data, it is therefore important to determine what a single person's data contributes to the graph. Prior work has looked at two forms of differential privacy in graphs – *edge differential privacy*, where an edge corresponds to a person's private value, and *node differential privacy*, where a single node corresponds to a person. In our motivating application, a patient corresponds to a node, and hence node differential privacy is our privacy notion of choice.

Prior work on edge differential privacy [14], [5] has looked at how to compute a number of statistics for *static graphs* while preserving privacy. For example, [15] computes subgraph counts, and [5] degree distributions with edge differential privacy. It is also known how to successfully calculate more complex graph parameters under this notion; for example, [16] fits exponential random graph models and [17], [18] computes spectral graph statistics such as pagerank.

In contrast, achieving node differential privacy is considerably more challenging. Changing a single node and its associated edges can alter even simple statistics of a static graph significantly; this means that any differentially private solution needs to add a considerable amount of noise to hide the effect of a single node, resulting in low utility. Prior work has addressed this challenge in two separate ways. The first is to assume that there is a publicly known upper bound on the maximum degree of any node in the graph [19], [20].

The second is to use a carefully-designed projection from the input graph to a bounded degree graph, where adding or removing a single node has less effect, and then release statistics of the projected graph with privacy. To ensure that the entire process is privacy-preserving, the projection itself is required to be *smooth* – in the sense that changing a single node should not change the statistics of the projected graph by much. This approach has been taken by [21], who releases degree distributions and subgraph counts for static graphs and [22], who releases subgraph counts and local profile queries. [23] releases degree distributions by using a flow-based projection algorithm. Finally, [24] proposes an improved projection method for releasing degree distributions, and is the state-of-the-art in this area. In this paper, we show that when the graph arrives *online*, existing projection-based approaches can yield poor utility, and therefore, we consider bounded degree graphs, where domain knowledge suggests an upper bound on the maximum degree.

Finally, while we are not aware of any work on differentially private statistics on streaming graph data, prior work has looked at releasing private statistics on streaming *tabular data* in an online manner [6], [7], [8], [9], [10], [11]. In these settings, however, complete information about a single person (or a group of people) arrives at each time step, which makes the problem of private release considerably easier than online graph data, where newly arriving nodes may include information in the form of edges to already existing nodes. Thus, these approaches do not directly translate to online graphs.

II. PRELIMINARIES

A. Graphs and Graph Sequences

Formally, we consider a graph $G = (V, E)$, where a node $v \in V$ represents a person and an edge $(u, v) \in E$ a relationship. G may be directed or undirected, depending on the application. We assume that each node $v \in V$ is associated with a time stamp, denoted by $v.\texttt{time}$, that records when v enters the graph.

In our setting, a graph arrives online as more and more of its vertices and some of their adjacent edges become visible. More specifically, at time t, a set of vertices ∂V_t arrives, along with a set of edges ∂E_t; each edge in ∂E_t

has at least one end-point in ∂V_t, and the other end-point may be a vertex that arrived earlier. These vertices and edges, along with vertices and edges that arrived earlier comprise a graph G_t. Given a function f that operates on graphs, our goal is to output (a private approximation to) $f(G_t)$ at each time step t.

More formally, the arrival process comprises a *graph sequence* $\mathcal{G} = (G_1, G_2, \ldots)$, which is defined as a sequence of graphs with $G_t = (V_t, E_t)$ such that $V_0 = \emptyset$, $\partial V_t = \{v : v.\texttt{time} = t\}$ is the set of all nodes with time stamp t and $V_t = V_{t-1} \cup \partial V_t$ for $t \geq 1$ is the set of all nodes with time stamps $\leq t$. Additionally, we let $E_0 = \emptyset$, $\partial E_t = \{(u, v) | u \in \partial V_t, v \in V_t \text{ or } u \in V_t, v \in \partial V_t\}$, and $E_t = E_{t-1} \cup \partial E_t$. Given a function f that operates on a graph, we define f applied to the graph sequence $f(\mathcal{G})$ as the sequence $(f(G_1), f(G_2), \ldots)$.

If the graph sequence is $\mathcal{G} = (G_1, G_2, \ldots, G_T)$, then the error of $\mathcal{A}(G)$ is defined as: $\sum_{t=1}^{T} |\mathcal{A}(G_t) - f(G_t)|$. Our goal is to design an algorithm $\mathcal{A}$ that has as low error as possible.

B. Differential Privacy

The gold standard for privacy in data-mining applications is differential privacy [4], which essentially ensures that the participation of a single person in a database does not change the probability of any outcome by much. The formal definition is as follows.

Definition II.1 (ϵ-Differential privacy). *A randomized algorithm $\mathcal{A}$ is said to guarantee ϵ-differential privacy if for any two databases S and S' that differ in the private value of a single individual, and for any $w \in Range(\mathcal{A})$, we have*

$$\mathbb{P}\left[\mathcal{A}(S) = w\right] \leq e^{\epsilon} \cdot \mathbb{P}\left[\mathcal{A}(S') = w\right],$$

where the probability is with respect to the randomness in $\mathcal{A}$. Here ϵ is a privacy parameter, often called the privacy budget.

    a) Global Sensitivity Mechanism.: A popular differential privacy mechanism is the Global Sensitivity Mechanism, introduced by [4]. Let f be a function that operates on a database S. The Global Sensitivity of f, denoted by GS (f), is the maximum value of the difference $\|f(S) - f(S')\|_1$ when S and S' are any two databases that differ by a the participation of a single person.

Given a function f, a database S and a privacy budget ϵ, the Global Sensitivity Mechanism computes an ϵ-differentially private approximation to $f(S)$ as follows: $\mathcal{A}_{GS}(S) = f(S) + \text{Lap}\left(\frac{\text{GS}(f)}{\epsilon}\right)$. It was shown in [4] that this method preserves ϵ-differential privacy.

    b) Node Differential Privacy.: To apply differential privacy to graphs, we need to determine what constitutes a single person's data in a graph. For the kind of graphs that we will study, a node v corresponds to a single person. This is known as *node differential privacy* [5], which ensures that the addition or removal of a single node along with its adjacent edges does not change the probability of any outcome by much.

C. Bounded Degree Graphs

A major challenge with ensuring node differential privacy is that the global sensitivity GS (f) may be very large even for simple graph functions f, which in turn requires the addition of a large amount of noise to ensure privacy. For example, if f is the number of nodes with degree ≥ 1, and we have an empty graph G on n nodes, then adding a single node connected to every other node can increase f by as much as n.

Prior work has addressed this challenge in two separate ways. The first is by considering *Bounded Degree Graphs* [21], [22], [19], where an a-priori bound on the degree of any node is known to the user and the algorithm designer. This is the solution that we will consider in this paper.

A second line of prior work [22], [21], [23], [24] presents *Graph Projections* algorithms that may be used to project graphs into lower degree graphs such that the resulting projections have low global sensitivity for some graph functions. In Section IV-A1, we show that natural extensions of some of these projections may be quite unstable when a graph appears online.

We first define bounded degree graphs. Let $\deg_G(v)$ denote the degree of node v in an undirected graph G, out-$\deg_G(v)$ and in-$\deg_G(v)$ denote the out-degree and in-degree of v in a directed graph G.

Definition II.2. *An undirected graph $G = (V, E)$ is D-bounded if $deg_G(v) \leq D$ for any $v \in V$. A graph sequence $\mathcal{G} = (G_1, G_2, \ldots)$ is D-bounded if for all t, G_t is D-bounded. In other words, the degree of all nodes remain bounded by D in the entire graph sequence.*

A directed graph $G = (V, E)$ is D_{out}-out-bounded if out-$deg_G(v) \leq D_{out}$ for any $v \in V$. A graph sequence $\mathcal{G} = (G_1, G_2, \ldots)$ is D_{out}-out-bounded if for all t, G_t is D_{out}-out-bounded. Similarly, G is D_{in}-in-bounded if in-$deg_G(v) \leq D_{in}$ for any $v \in V$. $\mathcal{G}$ is D_{in}-in-bounded if for all t, G_t is D_{in}-in-bounded.

We say a directed graph or a graph sequence is (D_{in}, D_{out})-bounded if it is both D_{in}-in-bounded and D_{out}-out-bounded.

In this work, we assume that the domain consists only of degree bounded graphs. This ensures that the global sensitivity of certain common graph functions, such as degree distribution and subgraph counts, is low, and allows us to obtain privacy with relatively low noise. Additionally, many common sensitive graphs, such as the HIV transmission graph and co-authorship networks,

typically have relatively low maximum degree, thus ensuring that the assumption holds for low or moderate values of D.

D. Graph Functions

This work will consider two types of functions on graph sequences. The first consists of functions of the degree distribution. The specific functions we will look at for undirected graphs are highDeg$_\tau (G)$, which counts the number of nodes in G with degree $\geq \tau$ and the degree histogram hist (G), which counts the number of nodes with degree d for any $d \in \mathbb{N}_+$. Similarly, for directed graph, we consider highOutDeg$_\tau (G)$, the number of nodes with out-degree $\geq \tau$, and the out-degree histogram histOut (G), which counts the number of nodes with out-degree d for any $d \in \mathbb{N}_+$.

The second class of functions will involve subgraph counts. Given a subgraph S, we will count the number of occurrences of this subgraph $S(G)$ in the entire graph G. For example, when S is a triangle, $S(G)$ will count the number of triangles in the graph. When G is directed, so will be the corresponding subgraphs.

E. Other Notations

In a directed graph, an edge is denoted by an ordered tuple, i.e., (u, v) represents a directed edge pointing from node u to v.

We use $(a_t)_{t=1}^T$ as an abbreviation for vector $(a_1, a_2, \ldots, a_T)$.

For any integer i, we use $[i]$ to denote the set $\{1, 2, \ldots, i\}$.

A degree histogram h is a mapping from degrees to counts, i.e., given $d \in \mathbb{N}_+$, $h(d)$ is the number of nodes with degree equal to d. We define the distance between two histograms h and h' as $\|h(d) - h'(d)\|_1 = \sum_{d \in \mathbb{N}_+} |h(d) - h'(d)|$. Given two sequences of histograms $(h_t)_{t=1}^T$ and $(h'_t)_{t=1}^T$, we define the generalized L_1 distance between them as $\sum_{t=1}^T \|h_t(d) - h'_t(d)\|_1$.

III. MAIN ALGORITHM

Recall that we are given as input a D-bounded (or $(D_{\text{in}}, D_{\text{out}})$-bounded) graph sequence $\mathcal{G} = (G_1, G_2, \ldots, G_T)$ that arrives online, a privacy budget ϵ and a function f. Our goal is to publish an ϵ-differentially private approximation to the sequence $f(\mathcal{G})$ in an online manner. Specifically, at time t, an incoming vertex set ∂V_t and edges ∂E_t adjacent to it and the existing vertices arrive, and our goal is to release a private approximation to $f(G_t)$ with low additive L_1-error.

a) Baseline Approaches.: A naive approach is to calculate $f(G_t)$ at each t and add noise proportional to its global sensitivity over ϵ. Since ∂E_t may contain information on individuals in G_{t-1} in the form of adjacent edges, this procedure will not provide ϵ-differential privacy.

The correct way to do privacy accounting for this method is by sequential composition [4]. Suppose the graph sequence has total length T and we allocate privacy budget ϵ/T to each time step; then at time t, we calculate $f(G_t)$ and add noise proportional to its global sensitivity divided by ϵ/T. If the global sensitivity of $f(G_t)$ is $O(1)$, then, we add $O(T/\epsilon)$ noise to $f(G_t)$, which results in a $\Theta(T^2/\epsilon)$ expected L_1-error between $f(\mathcal{G})$ and the output of the algorithm.

A second approach is to calculate $f(G_t)$ and add noise proportional to the global sensitivity of the (entire) sequence $f(\mathcal{G})$ divided by ϵ. This preserves ϵ-differential privacy. However, the global sensitivity of the sequence $f(\mathcal{G})$ typically grows linearly with T, the length of the entire graph sequence, even if the graph sequence itself is degree-bounded. For example, if $f(G)$ is the number of nodes in G with degree $\geq \tau$, then, a single extra node with degree $\tau + 1$, added at time $t = 1$, can increase $f(G_t)$ by 1 for *every* t, resulting in a global sensitivity of $\Omega(T)$. Consequently, the expected L_1-error between the true value of $f(\mathcal{G})$ and the output of this approach is as again large as $\Theta(T^2/\epsilon)$.

b) Our Approach.: The main observation in this work is that for a number of popular functions, the *difference sequence* $\Delta = (f(G_1), f(G_2) - f(G_1), f(G_3) - f(G_2), \ldots)$ has considerably better properties. Observe that unlike certain functions on tabular data [6], [7], [8], [9], [10], [11], releasing $f(G_t) - f(G_{t-1})$ after adding noise proportional to its sensitivity over ϵ will still not be ϵ-differentially private – this is because ∂E_t can still include edges adjacent to people in G_{t-1}.

However, the difference sequence Δ does have considerably less global sensitivity than $f(\mathcal{G})$. In particular, we show that if the graph sequence $\mathcal{G}$ is D-bounded, then, the global sensitivity GS (Δ) of the entire difference sequence for a number of popular functions f depends only on D and not on the sequence length T. For example, in Section IV-A1, we show that when G is an undirected graph and f is the number of nodes with degree $\geq \tau$, the global sensitivity of the entire difference sequence is at most $2D + 1$.

This immediately suggests the following algorithm. At time t, calculate the difference $\Delta_t = f(G_t) - f(G_{t-1})$, and add Laplace noise proportional to its global sensitivity over ϵ to get a private perturbed version $\tilde{\Delta}_t$. Release the partial sum $\sum_{s=1}^t \tilde{\Delta}_s$, which is an approximation to $f(G_t)$. Since the expected value of $\tilde{\Delta}_t - \Delta_t$ is independent of T, the maximum standard deviation of

any partial sum is at most $O(\sqrt{T}/\epsilon)$, which results in an expected L_1-error of $O(T^{3/2}/\epsilon)$ – better than the $O(T^2/\epsilon)$-error achieved by the two baseline approaches.

Algorithm 1 SENSDIFF(Graph sequence $\mathcal{G}$, query f, privacy parameter ϵ)

> **for** $t = 1, \ldots, T$ **do**
>   Receive ∂V_t and ∂E_t, and construct G_t.
>   Calculate $\Delta_t = f(G_t) - f(G_{t-1})$.
>   Let $\mathrm{GS}_D(\Delta)$ be the global sensitivity of the difference sequence;
>   Calculate $\tilde{\Delta}_t = \Delta_t + \mathrm{Lap}\left(\frac{\mathrm{GS}_D(\Delta)}{\epsilon}\right)$, and the partial sum $\sum_{s=1}^{t} \tilde{\Delta}_s$.
> **end for**
> **return** $\left(\sum_{s=1}^{t} \tilde{\Delta}_s\right)_{t=1}^{T}$

The full algorithm, applied to a generic function f, is described in Algorithm 1. We call it SENSDIFF as it uses the global sensitivity of the difference sequence Δ. The rest of the paper is devoted to analyzing the global sensitivity of the difference sequence for a number of popular graph functions f. Our analysis exploits specific combinatorial properties of the graph functions in question, and is carried out for two popular classes of graph functions – functions of the degree distribution and subgraph counts.

IV. FUNCTIONS OF DEGREE DISTRIBUTIONS

We begin with functions of the degree distributions of the graph sequence, and consider both directed and undirected graphs. A summary of the results in this section is provided in Table I.

A. Undirected Graphs

For undirected graphs, we will consider two functions applied to graph sequences – first, the number of nodes with degree greater than or equal to a threshold τ, and second, the degree histogram.

1) Number of High Degree Nodes: Recall that $\mathrm{highDeg}_\tau(G)$ is the number of nodes in G with degree $\geq \tau$. We show below, that for D-bounded graphs, the difference sequence corresponding to $\mathrm{highDeg}_\tau(G)$ has global sensitivity at most $2D + 1$.

Lemma IV.1. *Let* $f(G) = highDeg_\tau(G)$. *For* D-*bounded graphs, the difference sequence corresponding to* f *has global sensitivity at most* $2D + 1$. *In fact, the global sensitivity is* $2D + 1$ *for any* $\tau < D$.

Notice that $\tau \leq D$ is needed for the statistic to be meaningful; if $\tau > D$, there is no high-degree node.

a) Projection Yields High Sensitivity in Graph Sequence: A common idea in static graph analysis with node differential privacy is to project the original graph into a bounded-degree graph. The sensitivity of some common statistics on this projected graph scales with the degree bound instead of the total number of nodes. The current state-of-the-art projection algorithm is proposed in [24]. Given a projection threshold $\tilde{D}$, a graph $G = (V, E)$ and an ordering of the nodes in V, the algorithm constructs a bounded-degree graph $G^{\tilde{D}}$ as follows. First, it adds all nodes in V to $G^{\tilde{D}}$; then it orders all edges in E according to the ordering of V, and for each edge (u, v), adds it to $G^{\tilde{D}}$ if and only if the addition does not make the degree of either u or v exceed $\tilde{D}$.

This algorithm can be easily adapted to the online graph setting. . However, it can be shown that the global sensitivity of the difference sequence is proportional to the total number of publications.

Lemma IV.2. *Let* $f(G) = highDeg_\tau\left(G^{\tilde{D}}\right)$. *For* D-*bounded graphs, the corresponding difference sequence that ends at time* T *has global sensitivity at least* $\tilde{D}T$ *for any* $\tilde{D} > \tau > 0$.

Notice that $\tilde{D} > \tau$ is needed for $\mathrm{highDeg}_\tau\left(G^{\tilde{D}}\right)$ to be meaningful; otherwise, we would have $\mathrm{highDeg}_\tau\left(G^{\tilde{D}}\right) = |V|$ for any G.

2) Degree Histogram: Degree histogram is another informative statistic of a graph. However, we can show that even for bounded graph, the sensitivity can scale quadratically with the degree bound. We use the generalized L_1 distance defined in Section II-E as the distance metric for the global sensitivity.

Lemma IV.3. *Let* $f(G) = hist(G)$. *For* D-*bounded graphs, the difference sequence corresponding to* f *has global sensitivity* $4D^2 + 2D + 1$.

B. Directed Graphs

For Directed graphs, we show similar results for the number of nodes with out-degree greater than or equal to a threshold τ and the out-degree histogram. Similar results can be obtained for in-degree.

1) Number of High Out-Degree Nodes: Recall that $\mathrm{highOutDeg}_\tau(G)$ denotes the number of nodes in G with degree $\geq \tau$. We show below that for D_{in}-in-bounded graphs, the difference sequence corresponding to $\mathrm{highOutDeg}_\tau(G)$ has global sensitivity $2D_{\mathrm{in}} + 1$.

Lemma IV.4. *Let* $f(G) = highOutDeg_\tau(G)$. *For* D_{in}-*in-bounded graphs, the difference sequence corresponding to* f *has global sensitivity* $2D_{in} + 1$.

2) Out-Degree Histogram: We show that for bounded directed graphs, the sensitivity of the histogram scales quadratically with the degree bounds as well.

TABLE I: Summary of degree distribution results.

	Undirected graph	Directed graph
(out-)degree histogram	$4D^2 + 2D + 1$ (for D-bounded)	$4D_{\text{out}}D_{\text{in}} + 2D_{\text{out}} + 1$ (for $(D_{\text{in}}, D_{\text{out}})$-bounded)
high-(out-)degree nodes	$2D + 1$ (for D-bounded)	$2D_{\text{in}} + 1$ (for D_{in}-in-bounded)

Lemma IV.5. *Let $f(G) = \text{histOut}(G)$. For (D_{in}, D_{out})-bounded graphs, the difference sequence corresponding to f has global sensitivity $4D_{out}D_{in} + 2D_{out} + 1$.*

V. FUNCTIONS OF SUBGRAPH COUNTS

In this section, we consider the count of some common directed and undirected subgraphs.

a) Popular subgraphs: In undirected graphs, we consider three subgraphs. 1) an edge, including two nodes and the edge between them, 2) a triangle, including three nodes with edges between any two of them and 3) a k-star, including one center node c, k boundary nodes $\{b_1, \ldots, b_k\}$ and edges $\{(c, b_i) : i \in [k]\}$. Table II summarizes these subgraphs.

In directed graphs, we consider five subgraphs. 1) an edge, including two nodes and an directed edge between them, 2) triangle I, including nodes $\{v_1, v_2, v_3\}$ and edges $\{(v_1, v_2), (v_2, v_3), (v_3, v_1)\}$, 3) triangle II, including nodes $\{v_1, v_2, v_3\}$ and edges $\{(v_1, v_2), (v_1, v_3), (v_2, v_3)\}$, 4) an out-$k$-star, including one center node c, k boundary nodes $\{b_1, \ldots, b_k\}$ and edges $\{(c, b_i) : i \in [k]\}$, 5) an in-k-star, including one center node c, k boundary nodes $\{b_1, \ldots, b_k\}$ and edges $\{(b_i, c) : i \in [k]\}$. Table III summarizes these subgraphs.

TABLE II: Subgraphs in undirected graphs.

Edge	Triangle	k-star
S_E^u	$S_\triangle^u$	$S_\star^u$

TABLE III: Subgraphs in directed graphs.

Edge	Triangle I	Triangle II	Out-k-star	In-k-star
S_E^d	$S_{\triangle 1}^d$	$S_{\triangle 2}^d$	$S_{\star o}^d$	$S_{\star i}^d$

A. Undirected Graphs

First, we present a general result that applies to any undirected subgraph S. Recall that $S(G)$ denotes the total number of copies of S in graph G.

Lemma V.1. *Given any undirected subgraph S, if $S(G)$ changes by at most S_+ with an additional node with degree D (and the corresponding edges), then the difference sequence corresponding to $S(\cdot)$ has global sensitivity S_+ for any D-bounded graph G.*

Now we show the values of S_+ for the subgraphs listed in Table II.

Lemma V.2. *Given the degree bound D, the value of S_+ for some common subgraphs are:*

1) *for S_E^u, $S_+ = D$;*
2) *for $S_\triangle^u$, $S_+ = \binom{D}{2}$;*
3) *for $S_\star^u$, $S_+ = D\binom{D-1}{k-1} + \binom{D}{k}$.*

B. Directed Graphs

Again, we first present a lemma that applies to any subgraph S, and then show the values of S_+, the maximum change in the subgraph count caused by an additional node, for the subgraphs in Table III.

Lemma V.3. *Given any directed subgraph S, if $S(G)$ changes by at most S_+ with an additional node with D_{in} in-degree and D_{out} out-degree (and the corresponding edges), then the difference sequence corresponding to $S(\cdot)$ has global sensitivity S_+ for any (D_{in}, D_{out})-bounded graph G.*

Lemma V.4. *Given degree bounds D_{in} and D_{out}, the value of S_+ for some common subgraphs are:*

1) *for S_E^d, $S_+ = D_{in} + D_{out}$;*
2) *for $S_{\triangle 1}^d$, $S_+ = D_{in}D_{out}$;*
3) *for $S_{\triangle 2}^d$, $S_+ = \binom{D_{in}+D_{out}}{2}$;*
4) *for $S_{\star o}^d$, $S_+ = D_{in}\binom{D_{out}-1}{k-1} + \binom{D_{out}}{k}$;*
5) *for $S_{\star i}^d$, $S_+ = D_{out}\binom{D_{in}-1}{k-1} + \binom{D_{in}}{k}$.*

VI. EXPERIMENTS

We next demonstrate the practical applicability of the proposed algorithm by comparing it with some natural baselines. In particular, we investigate the following questions:

1) What is the utility offered by SENSDIFF as a function of the privacy parameter ϵ and the number of releases?
2) How does its utility compare with existing baselines, such as composition across time steps, and composition coupled with graph projection?

These questions are considered in the context of five datasets – two synthetic and three real online graphs. We consider two versions of each dataset – directed and undirected, and two graph statistics – the number of high-degree nodes and the number of edges.

A. Methodology

1) Baseline Algorithms: We consider two natural baselines based on sequential differential privacy composition – COMPOSE-D-BOUNDED and COMPOSE-PROJECTION. COMPOSE-D-BOUNDED considers each graph G_t in the sequence separately, and adds noise

to $f(G_t)$ that is proportional to its D-bounded global sensitivity divided by ϵ/T. Here ϵ is the privacy parameter and T is the number of releases. COMPOSE-PROJECTIONuses a state-of-the-art projection algorithm – the one proposed in [24] – to project each G_t into $\tilde{G}_t$, and releases $f(\tilde{G}_t)$ after adding noise proportional to its global sensitivity divided by ϵ/T.

There are two other natural approaches. The first is to compute $f(\mathcal{G})$ and add noise proportional to its D-bounded global sensitivity divided by ϵ; the second is to use the projection algorithm to obtain a sequence of projected graphs $\tilde{\mathcal{G}} = (\tilde{G}_1, \ldots, \tilde{G}_T)$, compute $f(\tilde{\mathcal{G}})$ and add noise proportional to the its global sensitivity divided by ϵ. However, we can show that the utility of either of these approaches is guaranteed to be at most that of COMPOSE-D-BOUNDED and COMPOSE-PROJECTION; details are omitted due to space constraints.

2) Choice of Parameters: SENSDIFF and COMPOSE-D-BOUNDED both require an a-priori bound on the graph degree. We set this bound to be the actual maximum degree rounded up to the nearest 5-th integer.

COMPOSE-PROJECTION requires a projection threshold $\tilde{D}$ (or, $\tilde{D}_{\text{in}}$ and $\tilde{D}_{\text{out}}$ for directed graphs). This parameter is chosen by parameter tuning – we pick the parameter value out of a predetermined list that leads to the lowest error. Note that for the sake of fairness, we do not allocate any extra privacy budget to parameter tuning, which would be the case in reality; thus our estimate of the performance of COMPOSE-PROJECTION is *optimistic*.

The threshold τ in the high-degree or high-out-degree nodes experiments is set to be the 90-th percentile of the (non-private) degree distribution.

B. Datasets

We use five datasets – three real and two synthetic. In addition to the standard directed version of each dataset, we also consider an undirected version that is obtained by ignoring the edge directions. A brief summary of these datasets is presented in Table IV.

1) Real Data Sets:

a) HIV transmission graph: This is a graph of potential HIV transmissions where a node represents a patient and an edge connects two patients whose viral sequences have high similarity. An edge thus represents a plausible transmission; the graph also has spurious edges that may correspond to a patient transmitting the disease to multiple others within a short period of time. About two-thirds of the patients have an estimated date of infection (EDI) ranging from 1996 to 2016, which is taken as the time stamp of the corresponding node. For the remaining nodes, EDI could not be estimated as the patient was admitted long after infection; we set the corresponding time stamps as 1995.

b) Patents citation graph: The patents citation graph [25] contains all US patents granted between 1963 and 1999, and all citations made by patents from 1975 to 1999. A patent u's citing patent v naturally yields a directed edge from v to u. We pick all patents under subcategory *Computer Hardware & Software* (indexed 22) to form both a directed and an undirected graph, and publish statistics from years 1985 to 1999.

c) Paper citation graph: This is a graph of articles and their citations from ACL derived by [26]; each article has a recorded publication date from 1975 to 2013. We select the positive citations, where an edge from v to u implies that u endorsed the article v, and publish statistics from 1990 to 2013.

2) Synthetic Data Sets: In addition to the real data sets, we consider two synthetic graphs that are generated from two separate disease transmission models.

a) Synthetic disease transmission graph I: This is a synthetic graph of disease transmissions based on the Barabási–Albert preferential attachment model. In the Barabási–Albert model, there are m_0 initial nodes, followed by a number of nodes that arrive sequentially. A node on arrival connects to k existing nodes, with a higher chance of connecting to nodes with higher degree.

We make three modifications to this model so that the generated graph is a more realistic disease transmission network. First, we assume there is a total of Y years with n nodes added per year; each node has a time stamp – its year of arrival – and the initial nodes have time stamp 0. All edges are directed from nodes with lower time stamps to those with higher time stamps. Second, to model the large number of isolated nodes that exist in real disease transmission networks, we ensure that each new node is isolated with probability proportional to a parameter P_{isolated}. Third, in practice, an infected individual is usually less likely to spread infection as time passes due to treatment or death. We build this property into the model by adding an extra decaying factor to the connection probability, i.e., the probability of a new node's connecting to an existing node v is proportional to $\deg(v) \times (\texttt{current\_time} - v.\texttt{time} + 1)^{-c}$ (or out-deg$(v) \times (\texttt{current\_time} - v.\texttt{time} + 1)^{-c}$ in directed graph) where c is the decay parameter. For our experiments, we generate a graph with parameters $P_{\text{isolated}} = 0.5$, $k = 1$, $m_0 = 500$, $n = 70$, $Y = 20$ and $c = 1$.

b) Synthetic disease transmission graph II: This is a synthetic disease transmission graph drawn from the popular SIR model [27] of infection overlaid on an underlying Barabási–Albert social network. In the SIR model, a node or individual as three statuses – susceptible (S), infectious (I) and recovered (R). The infectious individuals transmit the disease to susceptible individuals through social links with a transmission probability P_t.

TABLE IV: Summary of common properties of the graph datasets. Max degree refers to the undirected version of the graph while max in-degree and max out-degree refer to the directed version.

	# nodes	# edges	timespan (year)	max degree	max in-degree	max out-degree
HIV transmission	1660	456	21	13	12	8
Patent citation	91614	475427	15	247	211	246
Paper citation	9038	5249	24	36	19	36
Synthetic I	1990	665	20	5	1	4
Synthetic II	1088	588	20	6	1	5

With probability P_r, an infectious individual can recover; once recovered, an individual will not get infected again.

We generate an undirected social network $G_{\text{interact}} = (V, E)$ from the Barabási–Albert model, where a node represents a person and an edge a social interaction. We then simulate the transmission process as follows. Initially, every node is susceptible (S) except for n_0 randomly picked infectious (I) nodes. At time step t, an infectious node changes status to recovered (R) with probability P_r; then, each node u that is still infectious infects each one of its neighbors in G_{interact} with probability $P_i/\deg_{G_{\text{interact}}}(u)$. This gives us a transmission graph where a directed edge is a disease transmission. The time stamp of each node is the time when it is infected. We note that any node in the social interaction graph that has never been infected will not appear in the transmission graph. For our experiments, we use the parameters $P_r = 0.1$, $P_i = 0.18$ and a underlying interaction graph G_{interact} of size 10000 with attachment parameter $k = 2$.

C. Results

We compare SENSDIFF with the baseline algorithms COMPOSE-D-BOUNDED and COMPOSE-PROJECTION under varying ϵ – the privacy parameter and varying T – the number of releases. We measure utility by the relative L_1 error $\sum_{t=1}^{T} |\mathcal{A}(G_t) - f(G_t)|/f(G_t)$, where $f(G_t)$ is the non-private statistic and $\mathcal{A}(G_t)$ is the value estimated by an algorithm.

Figure 1 shows the privacy-utility tradeoffs of the three algorithms for the number of high-degree nodes (1st and 2nd columns) and number of edges (3rd and 4th columns) across all datasets (different rows). There is a clear trend of decreasing error as ϵ increases for all algorithms; however, SENSDIFF yields smaller error compared to the other two baselines in all the cases. We point out that these results are overly optimistic for COMPOSE-PROJECTION – as we do not spend any privacy budget tuning the projection thresholds.

Figure 2 presents the utility of all algorithms across different datasets (different rows) and statistics (different columns) as a function of the number of releases T. We fix ϵ to be 5. We see that the relative error increases as T increases for all algorithms, across all datasets and statistics. Similar to the previous experiment, SENSDIFF

achieves lower errors in most cases for the number of high-degree nodes and significantly lower errors in all cases for the number of edges. In addition, we observe that the error of SENSDIFF increases at a smaller rate with increasing number of releases compared to the two baselines. This implies that SENSDIFF achieves better utility for large T, and confirms the theoretical arguments in Section III.

Reconsidering the two questions proposed in the beginning of the section, we conclude that SENSDIFF offers better utility under a wide range of ϵ and T compared to both baselines. COMPOSE-D-BOUNDED yields the worst utility across all datasets for both statistics, which is to be expected as it does not take advantage of either projections or additional properties of the graph sequence. SENSDIFF outperforms COMPOSE-PROJECTION in most cases, and has a more significant advantage for large T.

VII. CONCLUSION

In summary, we present a general algorithm for continually releasing statistics of a graph sequence. Our proposed algorithm exploits the difference sequence of the statistics, which has lower sensitivity compared to the original sequence, to achieve improved utility. We derive the global sensitivity of the difference sequences for common statistics including degree statistics and subgraph counts for bounded-degree graphs. Evaluations on real and synthetic graphs demonstrate the practical applicability of the proposed algorithm by showing that it outperforms two natural baselines over a wide range of parameters. In particular, the proposed algorithm is much less sensitive with respect to the number of releases compared with the baselines.

Our work is thus a step towards privacy-preserving analysis of online graphs, and has the potential to lead to insights in epidemiology of stigma-inducing diseases such as HIV, while still preserving the patient privacy.
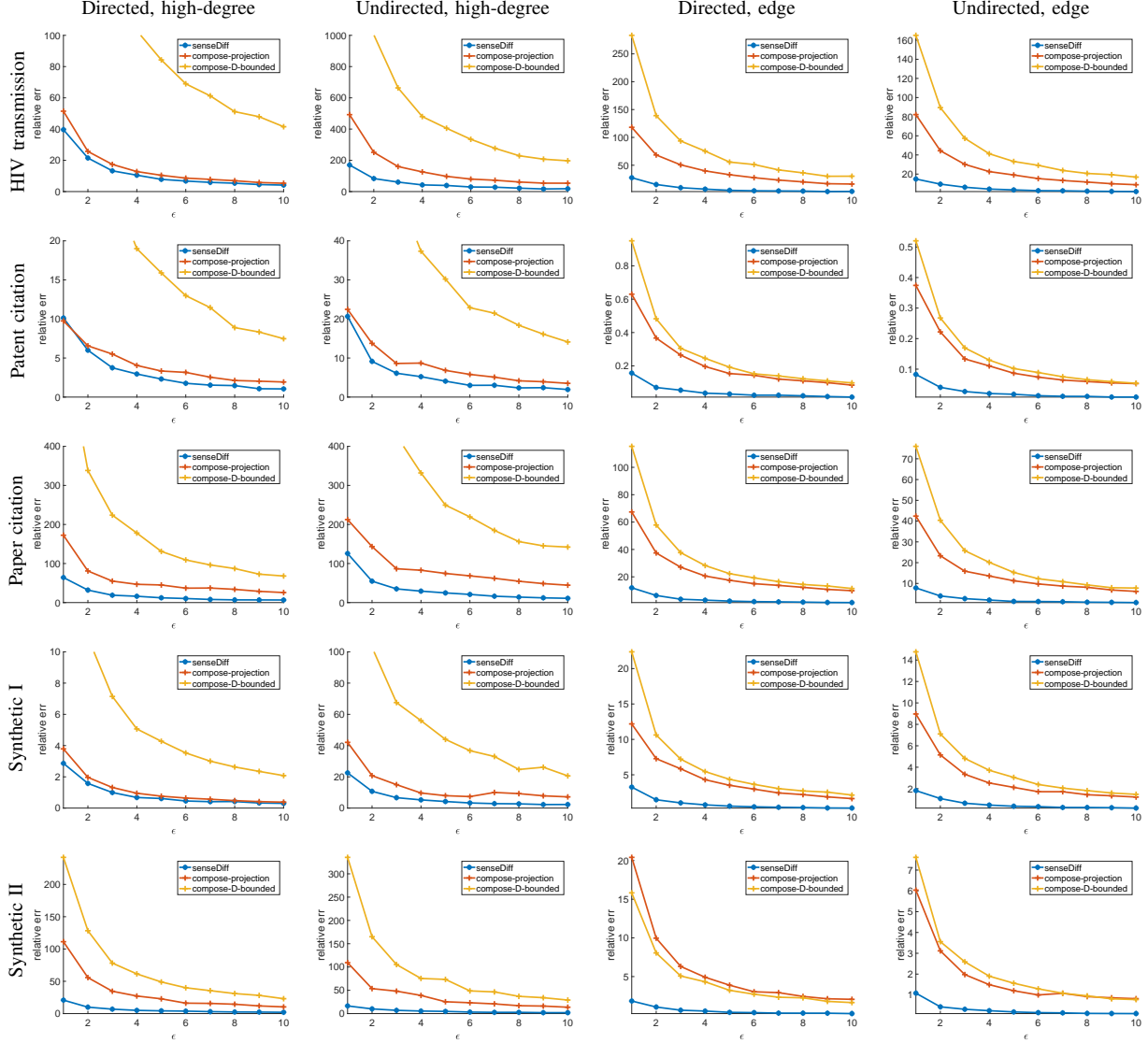
Fig. 1: L_1 error vs. privacy parameter ϵ. Publish every 1 year. Averaged over 100 runs.

REFERENCES

[1] S. J. Little, S. L. K. Pond, C. M. Anderson, J. A. Young, J. O. Wertheim, S. R. Mehta, S. May, and D. M. Smith, "Using hiv networks to inform real time prevention interventions," *PLoS one*, vol. 9, no. 6, p. e98443, 2014.

[2] X. Wang, Y. Wu, L. Mao, W. Xia, W. Zhang, L. Dai, S. R. Mehta, J. O. Wertheim, X. Dong, T. Zhang *et al.*, "Targeting hiv prevention based on molecular epidemiology among deeply sampled subnetworks of men who have sex with men," *Clinical Infectious Diseases*, vol. 61, no. 9, pp. 1462–1468, 2015.

[3] J. O. Wertheim, S. L. K. Pond, L. A. Forgione, S. R. Mehta, B. Murrell, S. Shah, D. M. Smith, K. Scheffler, and L. V. Torian, "Social and genetic networks of hiv-1 transmission in new york city," *PLoS pathogens*, vol. 13, no. 1, p. e1006000, 2017.

[4] C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating noise to sensitivity in private data analysis," in *TCC*, 2006.

[5] M. Hay, C. Li, G. Miklau, and D. Jensen, "Accurate estimation of the degree distribution of private networks," in *ICDM*. IEEE, 2009, pp. 169–178.

[6] J. Bolot, N. Fawaz, S. Muthukrishnan, A. Nikolov, and N. Taft, "Private decayed predicate sums on streams," in *ICDT*. ACM, 2013, pp. 284–295.

[7] J. Cao, Q. Xiao, G. Ghinita, N. Li, E. Bertino, and K.-L. Tan, "Efficient and accurate strategies for differentially-private sliding window queries," in *EDBT*. ACM, 2013, pp. 191–202.

[8] T.-H. H. Chan, E. Shi, and D. Song, "Private and continual release of statistics," *TISSEC*, vol. 14, no. 3, p. 26, 2011.

[9] G. Kellaris, S. Papadopoulos, X. Xiao, and D. Papadias, "Differentially private event sequences over infinite streams," *VLDB*, vol. 7, no. 12, pp. 1155–1166, 2014.

[10] Y. Chen, A. Machanavajjhala, M. Hay, and G. Miklau, "Pegasus: Data-adaptive differentially private stream processing," in *CCS*. ACM, 2017, pp. 1375–1388.

[11] C. Dwork, M. Naor, T. Pitassi, and G. N. Rothblum, "Differential privacy under continual observation," in *STOC*. ACM, 2010, pp. 715–724.

[12] A. Sarwate and K. Chaudhuri, "Signal processing and machine learning with differential privacy: Algorithms and challenges for continuous data," *Signal Processing Magazine, IEEE*, vol. 30,
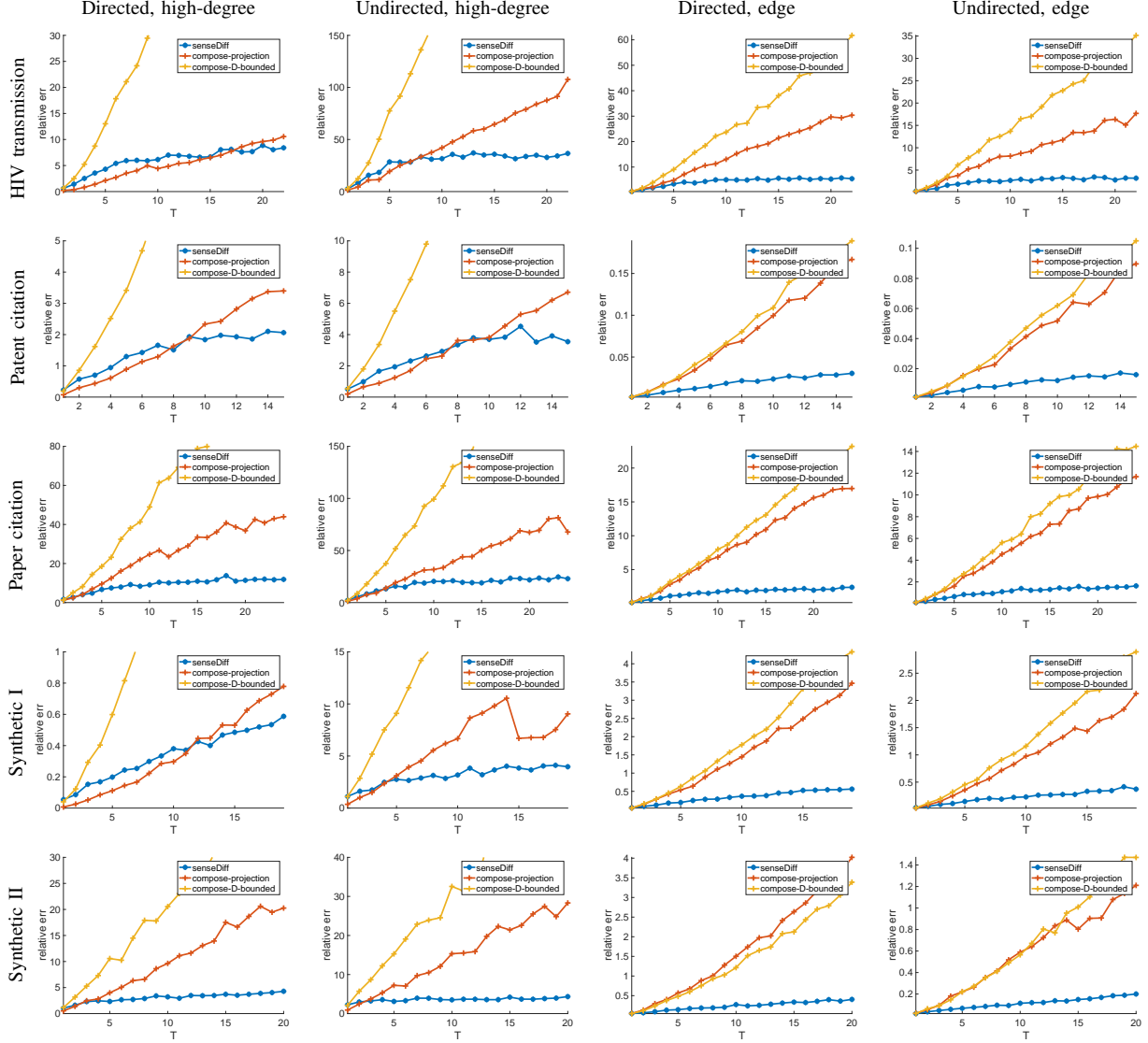
Fig. 2: L_1 error vs. number of publications T. $\epsilon = 5$. Averaged over 100 runs.

no. 5, pp. 86–94, Sept 2013.

[13] C. Dwork and A. Roth, "The algorithmic foundations of differential privacy," *TCS*, vol. 9, no. 3-4, pp. 211–407, 2013.

[14] K. Nissim, S. Raskhodnikova, and A. Smith, "Smooth sensitivity and sampling in private data analysis," in *STOC*, 2007.

[15] V. Karwa, S. Raskhodnikova, A. Smith, and G. Yaroslavtsev, "Private analysis of graph structure," *VLDB*, vol. 4, no. 11, pp. 1146–1157, 2011.

[16] W. Lu and G. Miklau, "Exponential random graph estimation under differential privacy," in *KDD*. ACM, 2014, pp. 921–930.

[17] Y. Wang, X. Wu, and L. Wu, "Differential privacy preserving spectral graph analysis," in *PAKDD*, 2013, pp. 329–340.

[18] F. Ahmed, R. Jin, and A. X. Liu, "A random matrix approach to differential privacy and structure preserved social network graph publishing," *arXiv preprint arXiv:1307.0475*, 2013.

[19] C. Borgs, J. Chayes, and A. Smith, "Private graphon estimation for sparse graphs," in *NIPS*, 2015, pp. 1369–1377.

[20] J. Gehrke, E. Lui, and R. Pass, "Towards privacy for social networks: A zero-knowledge based definition of privacy," in *TCC*. Springer, 2011, pp. 432–449.

[21] S. P. Kasiviswanathan, K. Nissim, S. Raskhodnikova, and A. Smith, "Analyzing graphs with differential privacy," in *TCC*. Springer, 2013, pp. 457–476.

[22] J. Blocki, A. Blum, A. Datta, and O. Sheffet, "Differentially private data analysis of social networks via restricted sensitivity," in *ITCS*. ACM, 2013, pp. 87–96.

[23] S. Raskhodnikova and A. Smith, "Efficient lipschitz extensions for high-dimensional graph statistics and node private degree distributions," *arXiv preprint arXiv:1504.07912*, 2015.

[24] W.-Y. Day, N. Li, and M. Lyu, "Publishing graph degree distribution with node differential privacy," in *SIGMOD*, 2016, pp. 123–138.

[25] B. H. Hall, A. B. Jaffe, and M. Trajtenberg, "The nber patent citation data file: Lessons, insights and methodological tools," National Bureau of Economic Research, Tech. Rep., 2001.

[26] S. Kumar, "Structure and dynamics of signed citation networks," in *WWW*, 2016, pp. 63–64.

[27] W. Kermack and A. McKendrick, "A contribution to the mathematical theory of epidemics," *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*,

APPENDIX A
NOTATIONS

We first introduce some notations which are going to be used throughout the Appendix.

Given an undirected graph G, let $\text{edge}_G(v)$ denote the set of edges that are adjacent to node v. Given a directed graph G, let $\text{in-edge}_G(v)$ and $\text{out-edge}_G(v)$ denote the set of in-edges and the set of out-edges adjacent to v respectively. We may omit τ and write $\text{highDeg}(G)$ or $\text{highOutDeg}(G)$ if it is clear from the context.

APPENDIX B
PROOFS OF SECTION IV

A. Undirected Graph

Proof. (of Lemma IV.1) Consider $G = (V, E)$ with $V = \cup_{t=1}^{\infty} \partial V_t$ and $G = (V', E')$ with $V' = \cup_{t=1}^{\infty} \partial V_t'$ such that $\partial V_i' = \partial V_i \cup \{v^*\}$ and $\partial V_j = \partial V_j'$ for $j \neq i$. Let $(\Delta_t)_{t=1}^{\infty}$ and $(\Delta_t')_{t=1}^{\infty}$ be the difference sequences corresponding to G and G'. Observe that $\Delta_t = \text{highDeg}_\tau(G_t) - \text{highDeg}_\tau(G_{t-1})$ is the number of nodes in G whose degrees *cross* the threshold τ at time t, and our goal is to bound $\sum_t |\Delta_t - \Delta_t'|$.

First, observe that as new edges become visible, for any node v, and for $j > i$, we have $\deg_{G_i}(v) \leq \deg_{G_j}(v)$. Thus the degree of v can cross the threshold τ in at most one time step. If this happens at time step t, then v's crossing over increases only the entry Δ_t by 1, and changes no other $\Delta_{t'}$ for $t \neq t'$.

Now consider the effect of adding v^* to the graph. Two things can happen as a result of this addition. First, v^* itself may have degree higher than τ, but since its degree can cross τ at only one time step t, this will increase at most one Δ_t by at most one.

Second, some of the nodes that are connected to v^* can cross the degree threshold τ early as a result on this extra connection. Let v_ℓ be a node connected to v^*. If the degree of v_ℓ crossed the threshold τ at time t in G and now crosses it at time $t' < t$ in G', then this can change only the two entries Δ_t and $\Delta_{t'}$, each by at most 1. Since there are at most D such nodes, and any node that does not connect to v^* crosses τ at the same time in G and G', the total change $\sum_t |\Delta_t - \Delta_t'|$ in the difference sequences is at most $2D + 1$.

Next, we show two neighboring graphs G and G' where the L_1-distance between the two difference sequences is exactly $2D + 1$ for $\tau < D$. For any D and τ, we can construct the following two graphs. Let $\partial V_1 = \{u_1, \ldots, u_{\tau-1}\} \cup \{v_1, \ldots, v_D\}$, $\partial V_2 = \{u_\tau\}$ and $\partial V_i = \emptyset$ for any $i \geq 3$. Let $\partial V_1' = \partial V_1 \cup \{v^*\}$ and $\partial V_i' = \partial V_i$ for any $i \geq 2$. Let $E = \{(v_i, u_j) : i \in [D], j \in [\tau]\}$ and $E' = E \cup \{(v_i, v^*) : i \in [D]\}$. It is obvious that $\deg(u_i) = D$, $\deg(v^*) = D$, $\deg_{G'}(v_j) = \tau + 1 \leq D$ for all $i \in [D], j \in [\tau]$, which means both G and G' are D-bounded. The difference sequence of G is $(\tau - 1, D + 1, 0, \ldots)$ and that of G' is $(D + \tau, 1, 0, \ldots)$, with L_1 distance $2D + 1$.

We can conclude that the D-bounded global sensitivity of the difference sequence is always upper bounded by $2D + 1$, and is exactly $2D + 1$ for any $\tau < D$. $\square$

Proof. (of Lemma IV.2) Consider the following graph G. Let there be T copies of a $(\tilde{D} - 1)$-star, denoted by $\{S_1, \ldots, S_T\}$. For each $t \in [T]$, let the center node of star S_t be C_t. For every $t \in [T]$, let there be an edge (C_i, C_{i+1}). And for each $t \in [T]$, let the time stamps of all nodes in S_t be t.

Let G' be a graph which is the same as G except for an additional node v^* and an additional edge (v^*, C_1). Let v^* have time stamp 0.

Now we consider the projected graphs with projection threshold $\tilde{D}$ for G and G'.

The edges in $G^{\tilde{D}}$ includes all the edges in S_i for all i, and (C_1, C_2), (C_3, C_4), (C_5, C_6), ... We thus have $(\Delta_t)_{t=1}^{T} = (0, 2, 0, 2, \ldots)$.

The edges in $G'^{\tilde{D}}$ includes all the edges in S_i for all i, and (v^*, C_1), (C_2, C_3), (C_4, C_5), ... We thus have $(\Delta_t)_{t=1}^{T} = (1, 0, 2, 0, 2 \ldots)$.

Therefore, $\sum_{t=1}^{T} |\Delta_t - \Delta_t'| = 2(T - 1) + 1 = 2T - 1$ for any T. Notice that if there are $\tilde{D}$ copies of the star sequences, and v^* is connected to the center of the first stars of all the sequences, then $\sum_{t=1}^{T} |\Delta_t - \Delta_t'|$ would become $(2T - 1)\tilde{D}$. $\square$

Proof. (of Lemma IV.3) Consider $G = (V, E)$ with $V = \cup_{t=1}^{T} \partial V_t$ and $G' = (V', E')$ with $V' = \cup_{t=1}^{T} \partial V_t'$ with $\partial V_i' = \partial V_i \cup \{v^*\}$ and $\partial V_j = \partial V_j'$ for $j \neq i$.

Moreover, observe that for any t, the t-th entry in the difference sequence, $\Delta_t = (n_{t,0}, n_{t,1}, \ldots, n_{t,D})$, where $n_{t,k}$ is the number of nodes which have degree k for the *first time* in G_t, minus the number of nodes whose degree changes from k to $k + 1$ at time t.

If (Δ_t) and (Δ_t') are the difference sequences corresponding to G and G' respectively, then our goal is to bound $\sum_t \|\Delta_t - \Delta_t'\|_1$.

First, observe that as new edges become visible, for any node v, and for $j > i$, we have $\deg_{G_i}(v) \leq \deg_{G_j}(v)$. Consider any node $v \in V$ with $\deg_G(v) = d$. Since $\deg_{G_i}(v)$ is increasing with respect to i, it can change at most d times after v enters the graph. Changing from k to k' in time t would decrease $n_{t,k}$ by 1 and increase $n_{t,k'}$ by 1, for a total change of 2, and the entrance of v to the graph with degree k_0 at time t would increase n_{t,k_0} by 1.

Now consider the effect of adding v^* on the difference sequence. Since the $\deg_{G'}(v^*) \leq D$, with the analysis above, it at most can increase $\sum_t \|\Delta_t - \Delta_t'\|_1$ by $2D + 1$

– at most one for v^*'s entrance to the graph, and at most $2D$ for degree increases.

Additionally, since $\deg_{G'}(v^*) \leq D$, there are at most D nodes connected to v^*; let us call them $u_1, \ldots, u_D$. Consider a particular u_k, which now has an extra edge to v^*. Observe that adding v^* cannot cause u_k to enter the graph at a different time; however, it can cause u_k to change its degree from j to $j+1$ for any $j \in \{0, \ldots, D-1\}$ at an earlier time step, which may cause $\sum_t |\Delta_t - \Delta_t'|_1$ to increase by at most $4D$. Since there are D such nodes, the total change due to the addition of v^* is at most $4D^2 + 2D + 1$. The lemma follows. $\square$

B. Directed Graph

Proof. (of Lemma IV.4) Consider $G = (V, E)$ with $V = \cup_{t=1}^T \partial V_t$ and $G = (V', E')$ with $V' = \cup_{t=1}^T \partial V_t'$ with $\partial V_i' = \partial V_i \cup \{v^*\}$ and $\partial V_j = \partial V_j'$ for $j \neq i$. Let (Δ_t) and (Δ_t') be the difference sequences corresponding to the graphs G and G' and $f = \mathrm{highOutDeg}_\tau(\cdot)$. Observe that $\Delta_t = \mathrm{highOutDeg}_\tau(G_t) - \mathrm{highOutDeg}_\tau(G_{t-1})$ is the number of nodes in G whose out-degrees cross the threshold τ at time step t, and our goal is to bound $\sum_t |\Delta_t - \Delta_t'|$.

First, observe that as new edges become visible, for any node v, and for $j > i$, we have $\mathrm{out\text{-}deg}_{G_i}(v) \leq \mathrm{out\text{-}deg}_{G_j}(v)$. Thus the out-degree of v can cross the threshold τ in at most one time step. If this happens at time step t, then v's crossing over increases only the entry Δ_t by 1, and changes no other entries.

Now consider the effect of adding v^* to the graph. Two things can happen as a result of this addition. First, v^* itself may have out-degree higher than τ, but since its degree can cross τ at only one time step t, this will increase at most one $|\Delta_t - \Delta_t'|$ by at most one.

Second, some of the nodes that are connected to v^* can cross the degree threshold τ early as a result on this extra connection. Let v_ℓ be a node that points to v^*. If the out-degree of v_ℓ crossed the threshold τ at time t in G and now crosses it at time $t' < t$ in G', then this can change only the two entries Δ_t and $\Delta_{t'}$, each by at most 1. Since there are at most D_{in} such nodes, and any node that does not point to v^* crosses τ at the same time in G and G', the total change $\sum_t |\Delta_t - \Delta_t'|$ in the difference sequences is at most $2D_{\mathrm{in}} + 1$.

Now we show the lower bound by constructing G and G' with L_1 distance $2D_{\mathrm{in}} + 1$ between their difference sequences. For any D_{in} and τ, we can construct the following two graphs. Let $\partial V_1 = \{u_1, \ldots, u_{\tau-1}\} \cup \{v_1, \ldots, v_{D_{\mathrm{in}}}\}$, $\partial V_2 = \{u_\tau\}$, $\partial V_3 = \{w_1, \ldots, w_\tau\}$ and $\partial V_i = \emptyset$ for any $i \geq 4$. Let $\partial V_1' = \partial V_1 \cup \{v^*\}$ and $\partial V_i' = \partial V_i$ for any $i \geq 2$. Let $E = \{(v_i, u_j) : i \in [D_{\mathrm{in}}], j \in [\tau]\}$ and $E' = E \cup \{(v_i, v^*) : i \in [D_{\mathrm{in}}]\} \cup \{(v^*, w_k) : k \in [\tau]\}$. It is obvious that $\mathrm{in\text{-}deg}(u_i) = D_{\mathrm{in}}$, $\mathrm{in\text{-}deg}(v^*) = D_{\mathrm{in}}$, $\mathrm{in\text{-}deg}(v_j) = 0$,

$\mathrm{in\text{-}deg}_{G'}(w_k) = 1$ for all $i \in [D_{\mathrm{in}}], j, k \in [\tau]$, which means both G and G' are D_{in}-in-bounded. The difference sequence of G is $(0, D_{\mathrm{in}}, 0, \ldots)$ and that of G' is $(D_{\mathrm{in}}, 0, 1, 0, \ldots)$, with L_1 distance $2D_{\mathrm{in}} + 1$.

We can conclude that the D_{in}-in-bounded global sensitivity of the difference sequence is $2D_{\mathrm{in}} + 1$. $\square$

Proof. (of Lemma IV.5) Consider $G = (V, E)$ with $V = \cup_{t=1}^\infty \partial V_t$ and $G' = (V', E')$ with $V' = \cup_{t=1}^\infty \partial V_t'$ with $\partial V_i' = \partial V_i \cup \{v^*\}$ and $\partial V_j = \partial V_j'$ for $j \neq i$.

Moreover, observe that for any t, the t-th entry in the difference sequence, $\Delta_t = (n_{t,0}, n_{t,1}, \ldots, n_{t,D})$, where $n_{t,k}$ is the number of nodes which have out-degree k for the *first time* subtract the number of nodes whose out-degree changes from k to $k+1$ in G_t. If (Δ_t) and (Δ_t') are the difference sequences corresponding to G and G' respectively, then our goal is to bound $\sum_t \|\Delta_t - \Delta_t'\|_1$.

First, observe that as new edges become visible, for any node v, and for $j > i$, we have $\mathrm{out\text{-}deg}_{G_i}(v) \leq \mathrm{out\text{-}deg}_{G_j}(v)$. Consider any node $v \in V$ with $\mathrm{out\text{-}deg}_G(v) = d$. Since $\mathrm{out\text{-}deg}_{G_i}(v)$ is increasing with respect to i, it can change at most d times after v enters the graph. Changing from k to k' in time t would decrease $n_{t,k}$ by 1 and increase $n_{t,k'}$ by 1, and the entering of v to the graph with out-degree k_0 at time t would increase n_{t,k_0} by 1.

Now consider the effect of adding v^* on the difference sequence. Since the $\mathrm{out\text{-}deg}_{G'}(v^*) \leq D_{\mathrm{out}}$, with the analysis above, it at most can increase $\sum_t \|\Delta_t - \Delta_t'\|_1$ by $2D_{\mathrm{out}} + 1$ – at most one for v^*'s entrance to the graph, and at most $2D_{\mathrm{out}}$ for out-degree increases.

Additionally, since $\mathrm{in\text{-}deg}_{G'}(v^*) \leq D_{\mathrm{in}}$, there are at most D_{in} nodes that point to v^*; let us call them $u_1, \ldots, u_D$. Consider a particular u_k, which now has an extra edge to v^*. Observe that adding v^* cannot cause u_k to enter the graph at a different time; however, it can cause u_k to change its out-degree from j to $j+1$ for any $j \in \{0, \ldots, D-1\}$ at an earlier time step, which may cause $\sum_t |\Delta_t - \Delta_t'|_1$ to increase by $4D_{\mathrm{out}}$. Since there are at most D_{in} such nodes, the total change due to the addition of v^* is at most $4D_{\mathrm{out}}D_{\mathrm{in}} + 2D_{\mathrm{out}} + 1$. The lemma follows. $\square$

APPENDIX C
PROOFS OF SECTION V

First, let $S_G(v)$ denote the number of copies of S in G that include v as one of its nodes.

A. Undirected Graph

Proof. (of Lemma V.1) The proof of Lemma V.3 applies. $\square$

Proof. (of Lemma V.2) Consider $G = (V, E)$ and $G' = (V', E')$ where $V' = V \cup \{v^*\}$ and $E' = E \cup \Delta E$

with ΔE consists of all edges adjacent to v^*. By the requirement, $|\Delta E| \leq D$.

For any subgraph S, let $\mathcal{S}$ denote the copies of S in G and $\mathcal{S}'$ denote that in G'. Notice that $\mathcal{S} \subseteq \mathcal{S}'$, since G is a subgraph of G' and thus any subgraph that appears in G would appear in G'. Moreover, any subgraph $\bar{S} \in \mathcal{S}'\backslash\mathcal{S}$ contain v^*, since otherwise it would appear in $\mathcal{S}$.

We now consider each of the subgraphs.

1) Every edge (v^*, v) can form one copy of S. There are in total at most D such edges.
2) Every pair of edges $(v^*, v_1) \in \Delta E$ and $(v^*, v_2) \in \Delta E$ can form one copy of S if there $(v_1, v_2) \in E$. There are in total at most $\binom{D}{2}$ such pairs, and thus $S_+ = \binom{D}{2}$.
3) If $k > D$, then there cannot be any copy of S in the graph, and thus $S_+ = 0$. Now we consider $k \leq D$. There are two types of stars in $\mathcal{S}'\backslash\mathcal{S}$, one with v^* as the center of the star, the other with v^* as a non-central node. Every k edges adjacent to v^* can form one star with v^* as its center; there are in total at most $\binom{D}{k}$ such stars. Every edge (v, v^*) of v^* can participate in at most $\binom{D-1}{k-1}$ stars. This is because in G, there are in total $\binom{\deg(v)}{k}$ stars that are centered at v and in G', the value becomes $\binom{\deg(v)+1}{k}$. Because of the degree bound, we have $\binom{\deg(v)+1}{k} - \binom{\deg(v)}{k} = \binom{\deg(v)}{k-1} \leq \binom{D-1}{k-1}$, where the last step follows because $\deg(v) + 1 \leq D$. There are at most D in-edges of v^*, which in total can form at most $D\binom{D-1}{k-1}$ stars. Therefore, $S_+ = \binom{D}{k} + D\binom{D-1}{k-1}$ for $k \leq D$.

$\square$

B. Directed Graph

Proof. (of Lemma V.3) Consider $G = (V, E)$ with $V = \cup_{t=1}^{\infty}\partial V_t$ and $G = (V', E')$ with $V' = \cup_{t=1}^{\infty}\partial V_t'$ with $\partial V_i' = \partial V_i \cup \{v^*\}$ and $\partial V_j = \partial V_j'$ for $j \neq i$.

Let (Δ_t) denote the difference sequence for subgraph S in G and (Δ_t') for that in G'. Recall that as defined in Section II-E, for a subgraph pattern S, $S(G)$ is the total number of S in graph G.

First, observe that for any graph G, as more nodes (and corresponding edges) become visible, the total number of S in G can only increase, i.e., $S(G_i) \leq S(G_j)$ for any $i < j$. So all elements Δ_t (and Δ_t) in the difference sequence are non-negative. Moreover, the sum of this sequence equals to $S(G)$ (and $S(G')$ for G).

Consider the additional node v^*. By requirement, if there are at most D_{in} in-edges and at most D_{out} out-edges to v^*, $S(G') - S(G)$ is at most S_+.

Second, let us compare Δ_t and Δ_t'. For any t, let $\mathcal{S}_t$ denote the copies of S that appears at time t for the first time in G, and $\mathcal{S}_t'$ denote that for G'. Any $S_t \in \mathcal{S}_t$ consists of some nodes in ∂V_t and some nodes in V_{t-1}.

Consider the appearance of S_t in G'. $\partial V_t \subseteq \partial V_t'$ and $V_{t-1} \subseteq V_{t-1}'$ implies that S_t must appear in G' at some $t' \leq t$; while $\partial V_t \cap V_{t-1}' = \emptyset$ implies that S_t appears at exactly t. Therefore, $\mathcal{S}_t \subseteq \mathcal{S}_t'$ and $\Delta_t' \geq \Delta_t$ for any t.

Therefore, $\sum_t |\Delta_t - \Delta_t'| = \sum_t (\Delta_t' - \Delta_t) = \sum_t \Delta_t' - \sum_t \Delta_t = S(G') - S(G) \leq S_+$, and the lemma follows.

$\square$

Proof. (of Lemma V.4) Consider $G = (V, E)$ and $G' = (V', E')$ where $V' = V \cup \{v^*\}$ and $E' = E \cup E_1 \cup E_2$ with E_1 consists of all in-edges of v^*, E_2 consists of all out-edges of v^*. By the requirement, $|E_1| \leq D_{\text{in}}$ and $|E_2| \leq D_{\text{out}}$.

For any subgraph S, let $\mathcal{S}$ denote the copies of S in G and $\mathcal{S}'$ denote that in G'. Notice that $\mathcal{S} \subseteq \mathcal{S}'$, since G is a subgraph of G' and thus any subgraph that appears in G would appear in G'. Moreover, any subgraph $\bar{S} \in \mathcal{S}'\backslash\mathcal{S}$ contain v^*, since otherwise it would appear in $\mathcal{S}$.

We now consider each of the subgraphs.

1) Every edge (v^*, v) or (v^*, v) can form one copy of S. There are in total at most $D_{\text{in}} + D_{\text{out}}$ such edges.
2) Consider any $\bar{S} \in \mathcal{S}'\backslash\mathcal{S}$. According to the definition of S, $\bar{S}$ must contain one in-edge and one out-edge of v^*. Notice that every pair of edges $(v_1, v^*) \in E_1$ and $(v^*, v_2) \in E_2$ can form one copy of S if there $(v_2, v_1) \in E$. There are in total at most $D_{\text{in}}D_{\text{out}}$ such pairs, and thus $S_+ = D_{\text{in}}D_{\text{out}}$.
3) Every pair of edges $(v_1, v^*) \in E_1$ and $(v^*, v_2) \in E_2$ can form one copy of S if there $(v_1, v_2) \in E$; every pair of edges $(v_1, v^*) \in E_1$ and $(v_1', v^*) \in E_1$ can form one copy of S if there $(v_1, v_1') \in E$ or $(v_1', v_1) \in E$, every pair of edges $(v^*, v_2) \in E_2$ and $(v^*, v_2') \in E_2$ can form one copy of S if there $(v_2, v_2') \in E$ or $(v_2', v_2) \in E$. There are in total at most $\binom{D_{\text{in}}+D_{\text{out}}}{2}$ such pairs, and thus $S_+ = \binom{D_{\text{in}}+D_{\text{out}}}{2}$.
4) If $k > D_{\text{out}}$, then there cannot be any copy of S in the graph, and thus $S_+ = 0$. Now we consider $k \leq D_{\text{out}}$. There are two types in $\mathcal{S}'\backslash\mathcal{S}$, one with v^* as the center of the star, the other with v^* as a non-central node. Every k out-edges of v^* can form one star with v^* as its center; there are in total at most $\binom{D_{\text{out}}}{k}$ such stars. Every in-edge (v, v^*) of v^* can form at most $\binom{D_{\text{out}}-1}{k-1}$ stars. This is because in G, there are in total $\binom{\text{out-deg}(v)}{k}$ stars that are centered at v and in G', the value becomes $\binom{\text{out-deg}(v)+1}{k}$. Because of the degree bound, we have $\binom{\text{out-deg}(v)+1}{k} - \binom{\text{out-deg}(v)}{k} = \binom{\text{out-deg}(v)}{k-1} \leq \binom{D_{\text{out}}-1}{k-1}$, where the last step follows because $\text{out-deg}(v) + 1 \leq D_{\text{out}}$. There are at most D_{in} in-edges of v^*, which in total can form at most $D_{\text{in}}\binom{D_{\text{out}}-1}{k-1}$ stars. Therefore, $S_+ = \binom{D_{\text{out}}}{k} + D_{\text{in}}\binom{D_{\text{out}}-1}{k-1}$ for $k \leq D_{\text{out}}$.
5) If we reverse the graph by converting every edge (u, v) to (v, u), then the problem reduces to the

above problem. The same result applies with D_{in} and D_{out} flipped, since the reversed graph is $(D_{\text{out}}, D_{\text{in}})$-bounded.

$\square$

APPENDIX D
BASELINE ALGORITHMS AND ANALYSES

In this section, we describes and analyze some baseline algorithms that are used in our experiments to publish the number of high-degree nodes. Since some of them use an extension of the projection algorithm in [24], we would formally describe the extensions of the projection algorithm and then describe our baseline algorithms with their sensitivity analysis.

A. Extension of the Projection Algorithm in [24] to Directed and Online Graphs

[24] provides a projection algorithm for undirected graph. In this section, we show that it can be adapted to directed graph and graph sequence. We describe the adapted algorithm in this section, and provide analysis in the next section for using it to publish the number of high-degree nodes.

The idea of the algorithm in [24] is to consider each edge one by one, and add it to the projected graph only if both ends of the edge have degree lower than the projection threshold $\tilde{D}$. The algorithm therefore needs an ordering of all edges, and the privacy proof in [24] requires the ordering to be consistent under G, G', i.e., two edges, if appears in both G and G', would have same relative order.

The algorithm can be naturally adapted to the directed graph setting, by having two projection thresholds, $\tilde{D}_{\text{in}}$ for in-degree and $\tilde{D}_{\text{out}}$ for out-degree. It can also be naturally adapted to the online graph setting, where we have a sequence of graphs $(G_1, G_2, \ldots,)$, and would like to publish a sequence of projected graphs $(G_1^{\tilde{D}}, G_2^{\tilde{D}}, \ldots)$. Here we can consider all edges in ∂E_1 one by one and publish $G_1^{\tilde{D}}$, and then consider edges in ∂E_2 and publish $G_2^{\tilde{D}}$ etc. More formally, we require the ordering used in the algorithm to be consistent with their order of appearance in the graph sequence.

For a graph G, let Λ be the ordering, represented by a sequence of all edges, such that e_1's appearing before e_2 means that e_1 is considered before e_2 by the algorithm. For a graph sequence, let there be an ordering sequence $(\Lambda_1, \ldots, \Lambda_T)$, where each Λ_i is an ordered sequence of all edges in ∂E_i. All edges in Λ_i are considered before edges in Λ_j for $i < j$, and edge e_1 is considered before e_2 if e_1 appears before e_2 in some Λ_i. Corresponding to Λ or $(\Lambda_1, \ldots, \Lambda_T)$, we can define a function λ that maps an edge to a unique value, such that $\lambda(e_1) < \lambda(e_2)$ means e_1 appears before e_2 in Λ.

Now, given the time stamps of all nodes, we can define the time stamp of an edge $e = (v_1, v_2)$ as $e.\texttt{time} = \max\{v_1.\texttt{time}, v_2.\texttt{time}\}$. For a graph sequence, we want $(\Lambda_1, \ldots, \Lambda_T)$ to be consistent with the time stamps of the edges in G, i.e., for two edges e, e' with $e.\texttt{time} < e'.\texttt{time}$, either $e \in \Lambda_i$, $e' \in \Lambda_j$ with $i < j$, or e and e' both appear in some Λ_i and e appears before e' in Λ_i. Obviously, such ordering (sequence) is consistent under G and G'.

Now we formally state the algorithms. Algorithm 2 is for projection of one undirected or directed graph, and Algorithm 3 is for an undirected or directed graph sequence. We note that Algorithm 2 with undirected graph is the original projection algorithm proposed in [24].

Algorithm 2 Projection algorithm for one undirected / directed graph (Graph $G = (V, E)$, edge ordering Λ, projection parameter $\tilde{D}$ (for undirected graph) or $\tilde{D}_{\text{in}}$, $\tilde{D}_{\text{out}}$ (for directed graph))

$\tilde{V} = V$, $\tilde{E} = \emptyset$.
for $(u, v) \in \Lambda$ **do**
    if $\deg_{(\tilde{V}, \tilde{E})}(u) < \tilde{D}$ **and** $\deg_{(\tilde{V}, \tilde{E})}(v) < \tilde{D}$
(or for directed graph, out-$\deg_{(\tilde{V}, \tilde{E})}(u) < \tilde{D}_{\text{out}}$ **and** in-$\deg_{(\tilde{V}, \tilde{E})}(v) < \tilde{D}_{\text{in}}$) **then**
        $\tilde{E} = \tilde{E} \cup \{(u, v)\}$.
    end if
end for
return $\tilde{G} = (\tilde{V}, \tilde{E})$

Algorithm 3 Projection algorithm for undirected / directed graph sequence (Graph $G = (V, E)$ and sequence $\mathcal{G} = (G_1, \ldots, G_m)$, edge ordering sequence $(\Lambda_1, \ldots, \Lambda_m)$, projection parameter $\tilde{D}$ (for undirected graph) or $\tilde{D}_{\text{in}}$, $\tilde{D}_{\text{out}}$ (for directed graph))

$\tilde{E} = \emptyset$.
for $i = 1, \ldots, m$ **do**
    $\tilde{V} = V_i$
    for $(u, v) \in \Lambda_i$ **do**
        if $\deg_{(\tilde{V}, \tilde{E})}(u) < \tilde{D}$ **and** $\deg_{(\tilde{V}, \tilde{E})}(v) < \tilde{D}$
(or for directed graph, out-$\deg_{(\tilde{V}, \tilde{E})}(u) < \tilde{D}_{\text{out}}$ **and** in-$\deg_{(\tilde{V}, \tilde{E})}(v) < \tilde{D}_{\text{in}}$) **then**
            $\tilde{E} = \tilde{E} \cup \{(u, v)\}$
        end if
    end for
    $\tilde{G}_i = (\tilde{V}, \tilde{E})$
end for
return $(\tilde{G}_1, \ldots, \tilde{G}_m)$

B. Baselines for Publishing the Number of High-Degree Nodes

In this section, we state the baseline algorithms for publishing the number of high-degree nodes of a graph sequence and show their sensitivity. All algorithms are based on the global sensitivity mechanism introduced in Section II-B, namely, to publish some $f(G)$, we compute the global sensitivity or the degree-bounded global sensitivity GS (f) of $f(G)$, and add noise proportional to GS $(f)/\epsilon$.

As has been mentioned in Section III, there are two baseline algorithms for high-degree nodes count. Combining with the algorithm proposed in [24], there are four baseline algorithms for both undirected and directed graph sequences.

For an undirected graph sequence $(G_1, \ldots, G_T)$,

1) COMPOSE-D-BOUNDED: For each $i \in [T]$, run the global sensitivity algorithm to publish $\mathrm{highDeg}_\tau (G_i)$ with privacy parameter ϵ/T and D-bounded global sensitivity $D+1$ (Lemma D.1).

2) COMPOSE-PROJECTION: For each $i \in [T]$, run Algorithm 2 on G_i with projection parameter $\tilde{D}$ to get $G_i^{\tilde{D}}$. Run the global sensitivity algorithm to publish $\mathrm{highDeg}_\tau \left(G_i^{\tilde{D}}\right)$ with privacy parameter ϵ/T and global sensitivity $\tilde{D}+1$ (Lemma D.3).

3) SENSSEQ-D-BOUNDED: Run the global sensitivity algorithm to publish $(\mathrm{highDeg}_\tau (G_i))_{i=1}^T$ with privacy parameter ϵ and D-bounded global sensitivity $(D+1)T$ (Lemma D.5).

4) SENSSEQ-PROJECTION: Run Algorithm 3 on $(G_i)_{i=1}^T$ with projection parameter $\tilde{D}$ to get $\left(G_i^{\tilde{D}}\right)_{i=1}^T$. Run the global sensitivity algorithm to publish $\left(\mathrm{highDeg}_\tau \left(G_i^{\tilde{D}}\right)\right)_{i=1}^T$ with privacy parameter ϵ and computed global sensitivity (Lemma D.7).

For a directed graph sequence $(G_1, \ldots, G_T)$,

1) COMPOSE-D-BOUNDED: For each $i \in [T]$, run the global sensitivity algorithm to publish $\mathrm{highOutDeg}_\tau (G_i)$ with privacy parameter ϵ/T and D_{in}-in-bounded global sensitivity $D_{\mathrm{in}} + 1$ (Lemma D.2).

2) COMPOSE-PROJECTION: For each $i \in [T]$, run Algorithm 2 on G_i with projection parameter $\tilde{D}_{\mathrm{in}}$, $\tilde{D}_{\mathrm{out}}$ to get $G_i^{\tilde{D}_{\mathrm{in}},\tilde{D}_{\mathrm{out}}}$. Run the global sensitivity algorithm to publish $\mathrm{highOutDeg}_\tau \left(G_i^{\tilde{D}_{\mathrm{in}},\tilde{D}_{\mathrm{out}}}\right)$ with privacy parameter ϵ/T and global sensitivity $\max\{\tilde{D}_{\mathrm{in}}+1, \tilde{D}_{\mathrm{out}}-1\}$ (Lemma D.4).

3) SENSSEQ-D-BOUNDED: Run the global sensitivity algorithm to publish $(\mathrm{highOutDeg}_\tau (G_i))_{i=1}^T$ with privacy parameter ϵ and D_{in}-in-bounded global sensitivity $(D_{\mathrm{in}} + 1)T$ (Lemma D.6).

4) SENSSEQ-PROJECTION: Run Algorithm 3 on $(G_i)_{i=1}^T$ with projection parameter $\tilde{D}_{\mathrm{in}}$, $\tilde{D}_{\mathrm{out}}$ to get $\left(G_i^{\tilde{D}_{\mathrm{in}},\tilde{D}_{\mathrm{out}}}\right)_{i=1}^T$. Run the global sensitivity algorithm to publish $\left(\mathrm{highOutDeg}_\tau \left(G_i^{\tilde{D}_{\mathrm{in}},\tilde{D}_{\mathrm{out}}}\right)\right)_{i=1}^T$ with privacy parameter ϵ and computed global sensitivity (Lemma D.7).

The lemmas below show that for both undirected and directed graph sequences, (3) is worse than (1); and (4) is worse than (2). Therefore, we only need to run (1) and (2), i.e., COMPOSE-D-BOUNDED and COMPOSE-PROJECTION.

1) Analyses for COMPOSE-D-BOUNDED:

Lemma D.1. *Given an undirected graph $G = (V, E)$, the D-bounded global sensitivity of publishing $\mathrm{highDeg}_\tau (G)$ is $D + 1$ for any $\tau \leq D$.*

Proof. Consider neighboring graphs $G = (V, E)$ and $G' = (V', E')$ with $V' = V \cup \{v^*\}$ and $E' = E \cup E^*$ where E^* consists of all edges adjacent to v^*.

Observe that for any node $v \in V$, $\deg_G (v)$ and $\deg_{G'} (v)$ are equal if v is not connected to v^* in G'. There are in total $|E^*| \leq D$ nodes that are connected to v^*; this extra connection can potentially push their degrees over the threshold. Moreover, v^* itself can have degree over τ. Therefore, $\mathrm{highDeg}_\tau (G') - \mathrm{highDeg}_\tau (G) \leq D + 1$.

Now we show neighboring D-bounded graphs G and G' such that $|\mathrm{highDeg}_\tau (G') - \mathrm{highDeg}_\tau (G)| = D + 1$. Let G consist of D $(\tau - 1)$-stars, and G' be equal to G except for an added node v^* that is connected to the center of all the stars. Since $D \geq \tau$, the degrees of all nodes still at most D. Additionally, we have $\mathrm{highDeg}_\tau (G) = 0$ and $\mathrm{highDeg}_\tau (G') = D + 1$. The lemma follows. $\square$

Lemma D.2. *Given a directed graph $G = (V, E)$, the D_{in}-in-bounded global sensitivity of publishing $\mathrm{highOutDeg}_\tau (G)$ is $D_{in} + 1$ for any $\tau \leq D_{out}$.*

Proof. Consider neighboring graphs $G = (V, E)$ and $G' = (V', E')$ with $V' = V \cup \{v^*\}$ and $E' = E \cup E_i^* \cup E_o^*$ where E_i^* consists of all in-edges adjacent to v^* and E_o^* consists of all out-edges adjacent to v^*. For any node $v \in V$, $\mathrm{out}\text{-}\deg_G (v)$ and $\mathrm{out}\text{-}\deg_{G'} (v)$ are equal if $(v, v^*) \notin E_i^*$; and there are in total $|E_i^*|$ nodes that have inbound edges to v^*; this extra edge can push their outdegrees over the threshold. Additionally, v^* itself may have outdegree over the threshold τ. Therefore, $\mathrm{highOutDeg}_\tau (G') - \mathrm{highOutDeg}_\tau (G) \leq |E_i^*| + 1 \leq D_{\mathrm{in}} + 1$.

Now we show neighboring D-bounded graphs G and G' such that $|\mathrm{highOutDeg}_\tau (G') - \mathrm{highOutDeg}_\tau (G)| = D_{\mathrm{in}} + 1$. Let G consists of D_{in} $(\tau - 1)$-out-stars and τ isolated nodes. Let G' be equal to G except for a node

v^* that has inbound edges from the center of all the stars and outbound edges to all the isolated nodes. Observe that as $\tau \leq D_{\text{in}}, D_{\text{out}}$, the in-degrees of all nodes are no larger than D_{in} and out-degrees of all nodes are no larger than D_{out}, and we have $\text{highOutDeg}_\tau(G) = 0$ and $\text{highOutDeg}_\tau(G') = D_{\text{in}} + 1$. $\qquad\square$

2) Analyses for COMPOSE-PROJECTION*:*

Lemma D.3. *Using Algorithm 2 on an undirected graph G to get $G^{\tilde{D}}$ and publishing $\text{highDeg}_\tau\left(G^{\tilde{D}}\right)$ has global sensitivity $\tilde{D} + 1$ for any $0 < \tau \leq \tilde{D}$.*

Proof. For two graphs H_1 and H_2 with the same set of nodes, we say an edge has the same status in graph H_1 and H_2 if it is present in both graphs or absent in both.

Consider neighboring graphs $G = (V, E)$ and $G' = (V', E')$ with $V' = V \cup \{v^*\}$ and $E' = E \cup E^*$, where E^* consists of all edges adjacent to v^*. Let Λ and λ be the edge order and the corresponding ordering function. Let the projected graphs be $G^{\tilde{D}}$ and $G'^{\tilde{D}}$.

In G', v^* may have a large number of adjacent edges E^*, but it is adjacent to at most $\tilde{D}$ edges in $G'^{\tilde{D}}$. Let such edges be $\{e_1^*, \ldots, e_i^*\}$ with $\lambda(e_j^*) < \lambda(e_{j+1}^*)$ for any j. Obviously, $i \leq \tilde{D}$. Notice that the projected graph of $(V', E \cup E^*)$ and that of $(V', E \cup \{e_1^*, \ldots, e_i^*\})$ are the same, therefore we can assume $E^* = \{e_1^*, \ldots, e_i^*\}$, which would not change $G'^{\tilde{D}}$.

Let $G_0 = (V', E)$, $G_1 = (V', E \cup \{e_1^*\})$, $G_2 = (V', E \cup \{e_1^*, e_2^*\})$, $\ldots$, $G_i = (V', E \cup \{e_1^*, \ldots, e_i^*\})$, i.e., G_0 is different from G by only an isolated node v^*, and G_i is exactly the same as G'. Moreover, considering the projected graphs, $G_0^{\tilde{D}}$ is the same as $G^{\tilde{D}}$ except for an isolated node v^*, since the isolated v^* in G_0 does not influence the projection algorithm; obviously, $G_i^{\tilde{D}}$ is the same as $G'^{\tilde{D}}$.

Given any graph H, let $f(H) = \text{highDeg}_\tau(H) - \mathbb{1}(\deg_H(v^*) \geq \tau)$, i.e., the number of high degree node in H not counting v^*. We let $\mathbb{1}(\deg_H(v^*) \geq \tau)$ be 0 if v^* is not in H. Obviously, $f\left(G_0^{\tilde{D}}\right) = f\left(G^{\tilde{D}}\right)$ (since an isolated node v^* does not have high degree) and $f\left(G_i^{\tilde{D}}\right) = f\left(G'^{\tilde{D}}\right)$. So we have

$$\left|f\left(G'^{\tilde{D}}\right) - f\left(G^{\tilde{D}}\right)\right| = \left|f\left(G_i^{\tilde{D}}\right) - f\left(G_0^{\tilde{D}}\right)\right|$$
$$= |f\left(G_i^{\tilde{D}}\right) - f\left(G_{i-1}^{\tilde{D}}\right) + f\left(G_{i-1}^{\tilde{D}}\right) - \cdots$$
$$- f\left(G_1^{\tilde{D}}\right) + f\left(G_1^{\tilde{D}}\right) - f\left(G_0^{\tilde{D}}\right)|$$
$$\leq \sum_{j=1}^{i}\left|f\left(G_j^{\tilde{D}}\right) - f\left(G_{j-1}^{\tilde{D}}\right)\right|.$$

Now we aim at bounding $\left|f\left(G_j^{\tilde{D}}\right) - f\left(G_{j-1}^{\tilde{D}}\right)\right|$ for any $j \in [i]$. Notice that G_j and G_{j-1} only differ by one edge e_j^*. Let us consider the influence of e_j^*

in the projected graph. Suppose $e^* = (v^*, v_0)$. Then there exists a finite "alternating" edge sequences (v^*, v_1), (v_1, v_2), (v_2, v_3), $\ldots$, $(v_{\ell-1}, v_\ell)$ for some $\ell \geq 1$ which satisfies the following conditions:

- All edges in the sequence are in E' and $\lambda((v^*, v_1)) < \lambda((v_1, v_2)) < \cdots < \lambda((v_{\ell-1}, v_\ell))$
- For all $k \in [\ell - 1] \cap 2\mathbb{Z} + 1$ (edges at odd positions), $\deg_{G_{j-1}^{\tilde{D}}}(v_k) = \tilde{D}$ and $(v_k, v_{k+1}) = \text{argmax}_{e \in F}\lambda(e)$ for $F = \text{edge}_{G_{j-1}^{\tilde{D}}}(v_k)$ $((v_k, v_{k+1})$ has the lowest priority among all adjacent edges of v_k in $G_{j-1}^{\tilde{D}})$
- For all $k \in [\ell - 2] \cap 2\mathbb{Z}$ (edges at even positions), $\deg_{G_{j-1}^{\tilde{D}}}(v_k) < \deg_{G_{j-1}}(v_k)$ and $(v_k, v_{k+1}) = \text{argmin}_{e \in F}\lambda(e)$ for $F = \text{edge}_{G_{j-1}}(v_k) \setminus \text{edge}_{G_{j-1}^{\tilde{D}}}(v_k)$ $((v_k, v_{k+1})$ has the highest priority among all adjacent edges of v_k in G_{j-1} but not in $G_{j-1}^{\tilde{D}})$

Running the projection algorithm, we would have the following process. (v^*, v_1) is added to $G_j^{\tilde{D}}$. Since $\deg_{G_{j-1}^{\tilde{D}}}(v_1) = \tilde{D}$, (v^*, v_1) will prevent another edge, (v_1, v_2), from being added to $G_j^{\tilde{D}}$. This saves one quota in the budget of v_2. Since some edge in $\text{edge}_{G_{j-1}}(v_2)$ is not present in $G_{j-1}^{\tilde{D}}$, we will have the first of them, (v_2, v_3), added to $G_j^{\tilde{D}}$ …This process keeps going and stops at $(v_{\ell-1}, v_\ell)$. In a word, all odd positioned edges in the sequence are added to $G_j^{\tilde{D}}$ and all even positioned edges are not. Any edge that is not in the sequence has the same status in $G_{j-1}^{\tilde{D}}$ and $G_j^{\tilde{D}}$.

Considering the degrees of all nodes in $\{v_1, \ldots, v_\ell\}$, only v_ℓ has different degrees in $G_{j-1}^{\tilde{D}}$ and $G_j^{\tilde{D}}$, and the difference is either 1 or -1; all other nodes in $\{v_1, \ldots, v_\ell\}$ have the same degree in both graphs. Also, all nodes in $V \setminus \{v_1, \ldots, v_\ell\}$ has the same degree in both graphs as well.

Therefore, $\left|f\left(G_j^{\tilde{D}}\right) - f\left(G_{j-1}^{\tilde{D}}\right)\right| \leq 1$, and thus $\left|f\left(G'^{\tilde{D}}\right) - f\left(G^{\tilde{D}}\right)\right| \leq i \leq \tilde{D}$. Taking into consideration that v^* can have high degree, we have $\left|\text{highDeg}_\tau\left(G'^{\tilde{D}}\right) - \text{highDeg}_\tau\left(G^{\tilde{D}}\right)\right| \leq \tilde{D} + 1$.

We now show that for any $\tilde{D}$ and τ such that $0 < \tau \leq \tilde{D}$, there exists neighboring graphs G and G' such that $\left|\text{highDeg}_\tau\left(G'^{\tilde{D}}\right) - \text{highDeg}_\tau\left(G^{\tilde{D}}\right)\right| = \tilde{D} + 1$. Let G be a graph with $\tilde{D}$ $(\tau - 1)$-stars, and G' be the same as G except for an additional node v^* that is connected to all the centers of the stars. Since $\tau \leq \tilde{D}$, all nodes in G and G' have degree no more than $\tilde{D}$, and thus $G^{\tilde{D}} = G$ and $G'^{\tilde{D}} = G'$. We have $\text{highDeg}_\tau(G) = 0$ and $\text{highDeg}_\tau(G') = \tilde{D} + 1$.

Therefore, the global sensitivity is $\tilde{D} + 1$. $\qquad\square$

Lemma D.4. *Using Algorithm 2 on a directed graph G to get $G^{\tilde{D}_{in}, \tilde{D}_{out}}$ and publishing $\text{highDeg}_\tau\left(G^{\tilde{D}_{in}, \tilde{D}_{out}}\right)$*

has global sensitivity $\max\{\tilde{D}_{in} + 1, \tilde{D}_{out} - 1\}$ *for any* $0 < \tau \leq \tilde{D}_{out}$.

Proof. For two graphs H_1 and H_2 with the same set of nodes, we say an edge has the same status in graph H_1 and H_2 if it is present in both graphs or absent in both.

Consider neighboring graphs $G = (V, E)$ and $G' = (V', E')$, with $V' = V \cup \{v^*\}$ and $E' = E \cup E_i^* \cup E_o^*$, where E_i^* consists of all in-edges adjacent to v^* and $^*E_o$ consists of all out-edges adjacent to v^*. Let Λ and λ be the edge order and the corresponding ordering function. To simplify the notation, we use $\tilde{G}$ to denote $G^{\tilde{D}_{in}, \tilde{D}_{out}}$ for any G. Let the projected graphs be $\tilde{G}$ and $\tilde{G}'$.

In G', v^* may have a large number of adjacent edges E^*, but it is adjacent to at most $\tilde{D}_{in}$ in-edges and at most $\tilde{D}_{out}$ out-edges in $\tilde{G}'$. Let such edges be $\{e_1^*, \dots, e_i^*\}$ with $\lambda(e_j^*) < \lambda(e_{j+1}^*)$ for any j. Obviously, this set contains at most $\tilde{D}_{in}$ in-edges and at most $\tilde{D}_{out}$ out-edges of v^*. Notice that the projected graph of $(V', E \cup E^*)$ and that of $(V', E \cup \{e_1^*, \dots, e_i^*\})$ are the same, therefore we can assume that E_i^* contains only the in-edges of v^* that are present in $\tilde{G}'$ and E_o^* contains only the out-edges of v^* that are present in $\tilde{G}'$ and have $E_i^* \cup E_o^* = \{e_1^*, \dots, e_i^*\}$. This assumption does not change $\tilde{G}'$.

Let $G_0 = (V', E)$, $G_1 = (V', E \cup \{e_1^*\})$, $G_2 = (V', E \cup \{e_1^*, e_2^*\})$, ..., $G_t = (V', E \cup E^*)$, i.e., G_0 is different from G by only an isolated node v^*, and G_i is exactly the same as G'. Moreover, considering the projected graphs, $\tilde{G}_0$ is the same as $\tilde{G}$ except for an isolated node v^*, since the isolated v^* in G_0 does not influence the projection algorithm; obviously, $\tilde{G}_i$ is the same as $\tilde{G}'$.

Given any graph H, let $f(H) = \text{highOutDeg}_\tau(H) - \mathbb{1}(\text{out-deg}_H(v^*) \geq \tau)$, i.e., the number of high out-degree nodes in H not counting v^*. We let $\mathbb{1}(\text{out-deg}_H(v^*) \geq \tau)$ be 0 if v^* is not in H. Obviously, $f(\tilde{G}_0) = f(\tilde{G})$ (since an isolated node v^* does not have high out-degree) and $f(\tilde{G}_t) = f(\tilde{G}')$. So we have

$$f(\tilde{G}') - f(\tilde{G}) = f(\tilde{G}_i) - f(\tilde{G}_0)$$
$$= f(\tilde{G}_i) - f(\tilde{G}_{i-1}) + f(\tilde{G}_{i-1}) - \dots$$
$$\quad - f(\tilde{G}_1) + f(\tilde{G}_1) - f(\tilde{G}_0)$$
$$= \left(f(\tilde{G}_i) - f(\tilde{G}_{i-1})\right) + \dots + \left(f(\tilde{G}_1) - f(\tilde{G}_0)\right).$$

Now we aim at calculating $f(\tilde{G}_j) - f(\tilde{G}_{j-1})$. Notice that G_j and G_{j-1} only differ by one edge e_j^*. Let us consider the influence of e_j^* in the projected graph. We need to consider two cases – e_j^* is an out-edge of v^* and e_j^* is an in-edge.

First, suppose $e_j^* = (v^*, v_1)$ is an out-edge of v^*.

There exists a finite "alternating" edges sequences $e_j^* = (v^*, v_1), e_2 = (v_2, v_1), e_3 = (v_2, v_3), e_4 = (v_4, v_3)$ ..., $e_\ell = (v_{\ell-1}, v_\ell)$ (or the last one might be $e_\ell = (v_\ell, v_{\ell-1})$) for some $\ell \geq 1$ which satisfies the following conditions:

- All edges in the sequence are in V' and $\lambda(e_j^*) < \lambda(e_2) < \dots < \lambda(e_\ell)$
- For all $k \in [\ell - 1] \cap 2\mathbb{Z} + 1$, in-deg$_{\tilde{G}_{j-1}}(v_k) = \tilde{D}_{in}$ and $e_{k+1} = (v_{k+1}, v_k) = \text{argmax}_{e \in E'} \lambda(e)$ for $E' = \text{in-edge}_{\tilde{G}_{j-1}}(v_k)$ (e_{k+1} has the lowest priority among all adjacent in-edges of v_k that are in $\tilde{G}_{j-1}$)
- For all $k \in [\ell - 2] \cap 2\mathbb{Z}$, out-deg$_{\tilde{G}_{j-1}}(v_k) <$ out-deg$_{G_{j-1}}(v_k)$ and $e_{k+1} = (v_k, v_{k+1}) = \text{argmin}_{e \in E'} \lambda(e)$ for $E' = \text{out-edge}_{G_{j-1}}(v_k) \setminus \text{out-edge}_{\tilde{G}_{j-1}}(v_k)$ (e_{k+1} has the highest priority among all adjacent out-edges of v_k that are not in $\tilde{G}_{j-1}$)

Running the projection algorithm, we would have the following process. (v^*, v_1) is added to $\tilde{G}_j$. Since in-deg$_{\tilde{G}_{j-1}}(v_1) = \tilde{D}_{in}$, (v^*, v_1) will prevent another edge, (v_2, v_1), from being added to $\tilde{G}_j$. This saves one quota in the out-degree budget of v_2. Since some edge in out-edge$_{G_{j-1}}(v_2)$ is not present in $\tilde{G}_{j-1}$, we will have the first of them, (v_3, v_2), added to $\tilde{G}_j$ ... This process keeps going and stops at e_ℓ. In a word, all odd positioned edges in the sequence are added to $\tilde{G}_j$ and all even positioned edges are not. Any edge that is not in the sequence has the same status in $\tilde{G}_{j-1}$ and $\tilde{G}_j$.

Considering the out-degrees of all nodes in $\{v_1, \dots, v_\ell\}$, if ℓ is even, only v_ℓ has different out-degrees in $\tilde{G}_{j-1}$ and $\tilde{G}_j$ and out-deg$_{\tilde{G}_{j-1}}(v_\ell) -$ out-deg$_{\tilde{G}_j}(v_\ell) = 1$ and all other nodes have the same degree in the two graphs; and if ℓ is odd, then all nodes have the same out-degrees in $\tilde{G}_{j-1}$ and $\tilde{G}_j$. All nodes in $V \setminus \{v_1, \dots, v_\ell\}$ have the same out-degree in $\tilde{G}_{j-1}$ and $\tilde{G}_j$.

Second, suppose $e_j^* = (v_1, v^*)$ is an in-edge of v^*.

There exists a finite "alternating" edges sequences $e_j^* = (v_1, v^*), e_2 = (v_1, v_2), e_3 = (v_3, v_2), e_4 = (v_3, v_4)$ ..., $e_\ell = (v_{\ell-1}, v_\ell)$ (or the last one might be $e_\ell = (v_\ell, v_{\ell-1})$) for some $\ell \geq 1$ which satisfies the following conditions:

- All edges in the sequence are in V' and $\lambda(e_j^*) < \lambda(e_2) < \dots < \lambda(e_\ell)$
- For all $k \in [\ell - 1] \cap 2\mathbb{Z} + 1$, out-deg$_{\tilde{G}_{j-1}}(v_k) = \tilde{D}_{out}$ and $e_{k+1} = (v_{k+1}, v_k) = \text{argmax}_{e \in E'} \lambda(e)$ for $E' = \text{out-edge}_{\tilde{G}_{j-1}}(v_k)$ (e_{k+1} has the lowest priority among all adjacent out-edges of v_k that are in $\tilde{G}_{j-1}$)
- For all $k \in [\ell - 2] \cap 2\mathbb{Z}$, in-deg$_{\tilde{G}_{j-1}}(v_k) <$ in-deg$_{G_{j-1}}(v_k)$ and $e_{k+1} = (v_k, v_{k+1}) = \text{argmin}_{e \in E'} \lambda(e)$ for $E' = \text{in-edge}_{G_{j-1}}(v_k) \setminus \text{in-edge}_{\tilde{G}_{j-1}}(v_k)$ (e_{k+1}

has the highest priority among all adjacent in-edges of v_k that are not in $\tilde{G_{j-1}}$)

In the projection algorithm, all odd positioned edges in the sequence are added to $\tilde{G}_j$ and all even positioned edges are not. Any edge that is not in the sequence has the same status in $\tilde{G_{j-1}}$ and $\tilde{G}_j$.

Considering the out-degrees of all nodes in $\{v_1, \ldots, v_\ell\}$, if ℓ is odd, only v_ℓ has different out-degrees in $\tilde{G_{j-1}}$ and $\tilde{G}_j$ and out-deg$_{\tilde{G}_j}(v_\ell) -$ out-deg$_{\tilde{G_{j-1}}}(v_\ell) = 1$, and all other nodes have the same degree in the two graphs; and if ℓ is even, then all nodes have the same out-degrees in $\tilde{G_{j-1}}$ and $\tilde{G}_j$. All nodes in $V \backslash \{v_1, \ldots, v_\ell\}$ has the same out-degree in $\tilde{G_{j-1}}$ and $\tilde{G}_j$.

Therefore, if e_j^* is an out-edge of v^*, then $f\left(\tilde{G}_j\right) - f\left(\tilde{G_{j-1}}\right)$ is either -1 or 0; if e_j^* an in-edge of v^*, then $f\left(\tilde{G}_j\right) - f\left(\tilde{G_{j-1}}\right)$ is either 1 or 0.
We have

$$\left(f\left(\tilde{G}_i\right) - f\left(\tilde{G_{i-1}}\right)\right) + \cdots + \left(f\left(\tilde{G}_1\right) - f\left(\tilde{G}_0\right)\right)$$
$$\in \left[-\text{out-deg}_{\tilde{G}'}(v^*), \text{in-deg}_{\tilde{G}'}(v^*)\right],$$

and since

$$\text{highOutDeg}_\tau\left(\tilde{G}'\right) - \text{highOutDeg}_\tau\left(\tilde{G}\right)$$
$$= f\left(\tilde{G}'\right) - f\left(\tilde{G}\right) + \mathbb{1}\left(\text{out-deg}_{\tilde{G}'}(v^*) \geq \tau\right),$$

we have

$$\text{highOutDeg}_\tau\left(\tilde{G}'\right) - \text{highOutDeg}_\tau\left(\tilde{G}\right)$$
$$\leq \text{in-deg}_{\tilde{G}'}(v^*) + \mathbb{1}\left(\text{out-deg}_{\tilde{G}'}(v^*) \geq \tau\right) \leq \tilde{D}_{\text{in}} + 1,$$
$$\text{highOutDeg}_\tau\left(\tilde{G}\right) - \text{highOutDeg}_\tau\left(\tilde{G}'\right)$$
$$\leq \text{out-deg}_{\tilde{G}'}(v^*) - \mathbb{1}\left(\text{out-deg}_{\tilde{G}'}(v^*) \geq \tau\right) \leq \tilde{D}_{\text{out}} - 1.$$

Therefore the global sensitivity is upper bounded by $\max\{\tilde{D}_{\text{in}} + 1, \tilde{D}_{\text{out}} - 1\}$.

We now show that for any $\tilde{D}_{\text{in}}, \tilde{D}_{\text{out}} > 0$, there exists neighboring graphs G and G' such that the difference between $\text{highOutDeg}_\tau\left(\tilde{G}'\right)$ and $\text{highOutDeg}_\tau\left(\tilde{G}\right)$ is $\tilde{D}_{\text{in}} + 1$ and another neighboring graphs G and G' such that the difference is $\tilde{D}_{\text{out}} - 1$.

Let G be a graph with $\tilde{D}_{\text{in}}$ $(\tau - 1)$-out-stars, and τ isolated nodes. Let G' be the same as G except for an additional node v^* that has in-edges from all the centers of the stars and out-edges to all the isolated nodes. All nodes in G and G' have out-degree no more than $\tilde{D}_{\text{out}}$ and in-degree no more than $\tilde{D}_{\text{in}}$, and thus $\tilde{G} = G$ and $\tilde{G}' = G'$. We have $\text{highOutDeg}_\tau(G) = 0$ and $\text{highOutDeg}_\tau(G') = \tilde{D}_{\text{in}} + 1$.

Let G be a graph with $\tilde{D}_{\text{out}}$ τ-out-stars, with the nodes of the i-th star labeled as $\{c^i, b_1^i, \ldots, b_\tau^i\}$, where c^i is the center node, and b_j^i denote the j-th outer

node of the i-th star. Additionally, for every $i \in [\tau]$, let there be a set of $D_{\text{in}} - 1$ nodes $\{a_1^i, \ldots, a_{\tilde{D}_{\text{in}}-1}^i\}$ that point to b_1^i. Suppose all edges in the stars have the lowest priority in Λ. Let G' be the same as G except for an additional node v^* with additional edges $\{(v^*, b_1^i) : i \in [\tau]\}$, i.e., v^* points to the first non-central node of all stars. Let this set of new edges take highest priority in the ordering Λ. In G, we have out-deg $\left(c^i\right) = \tau$, out-deg $\left(a_j^i\right) = 1$, out-deg $\left(b_j^i\right) = 0$ for all i, j; in-deg $\left(c^i\right) = 0$, in-deg $\left(a_j^i\right) = 0$ (for all j), in-deg $\left(b_1^i\right) = \tilde{D}_{\text{in}}$, in-deg $\left(b_j^i\right) = 1$ (for $j > 1$) for all i. So $\tilde{G} = G$, and $\text{highOutDeg}_\tau\left(\tilde{G}\right) = \tilde{D}_{\text{out}}$. On the other hand, running Algorithm 2 in G', we would first consider the edges adjacent to v^* and edges adjacent to a_j^i and add all of them to $\tilde{G}'$. Now, since for all i, b_1^i has in-degree equal to $\tilde{D}_{\text{in}}$, we can no longer add edge (c^i, b_1^i); we can still add (c^i, b_j^i) for all $j > 1$ and have our final $\tilde{G}'$. In $\tilde{G}'$, we have out-deg $\left(c^i\right) = \tau - 1$, out-deg $\left(a_j^i\right) = 1$, out-deg $\left(b_j^i\right) = 0$, out-deg $(v^*) = \tilde{D}_{\text{out}}$ for all i, j, and thus $\text{highOutDeg}_\tau\left(\tilde{G}'\right) = 1$. So $\text{highOutDeg}_\tau\left(\tilde{G}\right) - \text{highOutDeg}_\tau\left(\tilde{G}'\right) = \tilde{D}_{\text{out}} - 1$.

Therefore, the global sensitivity is $\max\{\tilde{D}_{\text{in}}+1, \tilde{D}_{\text{out}}-1\}$.

We note that it is a must to put bound on both in-degree and out-degree. Apparently, we must have a bound on the in-degree (otherwise v^* can have many in-edges and each can make the out-degree of one node cross τ). Yet if we only bound in-degree, then v^* can have many out-edges. Each out-edge (v^*, v_0) can saturate the in-degree of v_0 earlier, forcing it to discard another of its in-edge (v_1, v_0), causing the out-degree of v_1 to decrease by 1. So the sensitivity can be at least the out-degree of v^*. $\qquad\square$

3) Analyses for SENSSEQ-D-BOUNDED*:*

Lemma D.5. *Given an undirected graph G, the D-bounded global sensitivity for publishing $\left(\text{highDeg}_\tau(G_i)\right)\right)_{i=1}^T$ is $(D+1)T$.*

Proof. Consider $G = (V, E)$ and $G = (V', E')$ such that $V = \cup_{t=1}^\infty \partial V_t$, $V' = \cup_{t=1}^\infty \partial V_t'$, $\partial V_i' = \partial V_i \cup \{v^*\}$ and $\partial V_j = \partial V_j'$ for $j \neq i$. For any $v \in V$ that is not connected with v^*, $\text{deg}_{G_i}(v)$ and $\text{deg}_{G_i'}(v)$ are the same for any i. There are at most D nodes that are connected with v^*; let $V^* = \{v \in V : (v, v^*) \in E'\}$ be the set of

all such nodes. We have

$$\sum_{i=1}^{T}|\text{highDeg}_\tau\,(G_i) - \text{highDeg}_\tau\,(G_i')|$$

$$=\sum_{i=1}^{T}\left|\sum_{v\in V}\mathbb{1}\left(\deg_{G_i}(v)\geq\tau\right) - \sum_{v\in V'}\mathbb{1}\left(\deg_{G_i'}(v)\geq\tau\right)\right|$$

$$=\sum_{i=1}^{T}|\sum_{v\in V}\left(\mathbb{1}\left(\deg_{G_i}(v)\geq\tau\right) - \mathbb{1}\left(\deg_{G_i'}(v)\geq\tau\right)\right)$$
$$-\mathbb{1}\left(\deg_{G_i'}(v^*)\geq\tau\right)|$$

$$=\sum_{i=1}^{T}|\sum_{v\in V^*}\left(\mathbb{1}\left(\deg_{G_i}(v)\geq\tau\right) - \mathbb{1}\left(\deg_{G_i'}(v)\geq\tau\right)\right)$$
$$-\mathbb{1}\left(\deg_{G_i'}(v^*)\geq\tau\right)|$$

$$\leq\sum_{i=1}^{T}(|V^*|+1)\leq(D+1)T,$$

where the third equality follows from the fact that $\deg_{G_i'}(v) = \deg_{G_i}(v)$ for $v\notin V^*$, and the last inequality follows from the fact that the size of V^*, which consists of all nodes adjacent to v^*, is at most D.

Now we show that there exists neighboring G and G' such that the D-bounded global sensitivity is $(D+1)T$. Lemma D.1 shows that there exists neighboring D-bounded graphs G and G', such that $|\text{highDeg}\,(G') - \text{highDeg}\,(G)| = D+1$. Let $G_1 = G$, $G_1' = G'$. We then have $\text{highDeg}\,(G_i) = \text{highDeg}\,(G_1) = \text{highDeg}\,(G)$ and the same for G'. Therefore, $\sum_{i=1}^{T}|\text{highDeg}\,(G_i) - \text{highDeg}\,(G_i')| = (D+1)T$. □

Lemma D.6. *Given a directed graph G, the D_{in}-bounded global sensitivity for publishing $(\text{highOutDeg}_\tau\,(G_i))_{i=1}^{T}$ is $(D_{in}+1)T$.*

Proof. Consider $G = (V,E)$ and $G = (V',E')$ such that $V = \cup_{t=1}^{\infty}\partial V_t$, $V' = \cup_{t=1}^{\infty}\partial V_t'$, $\partial V_i' = \partial V_i\cup\{v^*\}$ and $\partial V_j = \partial V_j'$ for $j\neq i$. For any $v\in V$ that does not point to v^*, out-$\deg_{G_i}(v)$ and out-$\deg_{G_i'}(v)$ are the same for any i. There are at most D_{in} nodes that point to v^*; let $V^* = \{v\in V : (v,v^*)\in E'\}$ denote all such

nodes. We have

$$\sum_{i=1}^{T}|\text{highOutDeg}_\tau\,(G_i) - \text{highOutDeg}_\tau\,(G_i')|$$

$$=\sum_{i=1}^{T}|\sum_{v\in V}\mathbb{1}\left(\text{out-}\deg_{G_i}(v)\geq\tau\right)$$
$$-\sum_{v\in V'}\mathbb{1}\left(\text{out-}\deg_{G_i'}(v)\geq\tau\right)|$$

$$=\sum_{i=1}^{T}|\sum_{v\in V}\left(\mathbb{1}\left(\text{out-}\deg_{G_i}(v)\geq\tau\right)\right.$$
$$\left.-\mathbb{1}\left(\text{out-}\deg_{G_i'}(v)\geq\tau\right)\right) - \mathbb{1}\left(\text{out-}\deg_{G_i'}(v^*)\geq\tau\right)|$$

$$=\sum_{i=1}^{T}|\sum_{v\in V^*}\left(\mathbb{1}\left(\text{out-}\deg_{G_i}(v)\geq\tau\right)\right.$$
$$\left.-\mathbb{1}\left(\text{out-}\deg_{G_i'}(v)\geq\tau\right)\right) - \mathbb{1}\left(\text{out-}\deg_{G_i'}(v^*)\geq\tau\right)|$$

$$\leq\sum_{i=1}^{T}(|V^*|+1)\leq(D_{in}+1)T,$$

where the third equality follows from the fact that out-$\deg_{G_i'}(v) = $ out-$\deg_{G_i}(v)$ for $v\notin V^*$, and the last inequality follows from the fact that the size of V^*, which consists of all nodes that point to v^*, is at most D_{in}.

Now we show that there exists neighboring G and G' such that the D_{in}-in-bounded global sensitivity is $(D_{in}+1)T$. Lemma D.2 shows that there exists neighboring D_{in}-bounded graphs G and G', such that $|\text{highOutDeg}\,(G') - \text{highOutDeg}\,(G)| = D_{in}+1$. Let $G_1 = G$, $G_1' = G'$. We then have $\text{highOutDeg}\,(G_i) = \text{highOutDeg}\,(G_1) = \text{highOutDeg}\,(G)$ and the same for G'. Therefore, $\sum_{i=1}^{T}|\text{highOutDeg}\,(G_i) - \text{highOutDeg}\,(G_i')| = (D_{in}+1)T$. □

4) Analyses for SENSSEQ-PROJECTION*:*

Lemma D.7. *Using Algorithm 3 on an undirected (or directed) graph sequence $(G_i)_{i=1}^{T}$ to get $\left(G_i^{\tilde{D}}\right)_{i=1}^{T}$ (or $\left(G_i^{\tilde{D}_{in},\tilde{D}_{out}}\right)_{i=1}^{T}$) and publishing $\left(\text{highDeg}_\tau\left(G_i^{\tilde{D}}\right)\right)_{i=1}^{T}$ (or $\left(\text{highOutDeg}_\tau\left(G_i^{\tilde{D}_{in},\tilde{D}_{out}}\right)\right)_{i=1}^{T}$) has global sensitivity at least $(D+1)T$ (or at least $\max\{\tilde{D}_{in}+1,\tilde{D}_{out}-1\}\cdot T$).*

Proof. Let $\tilde{G}$ denote $G^{\tilde{D}}$ or $G^{\tilde{D}_{in},\tilde{D}_{out}}$ for any G.

From Lemma D.3 (or Lemma D.4), there exists graphs $G = (V,E)$ and $G' = (V',E')$, such that $\left|\text{highDeg}_\tau\left(\tilde{G}'\right) - \text{highDeg}_\tau\left(\tilde{G}\right)\right|$ (or the formula with highOutDeg ()) equal to $D+1$ (or $\max\{\tilde{D}_{in}+1,\tilde{D}_{out}-1\}$). Let G and G' be such graphs. Let $G_1 = G$,

$G'_1 = G'$ and $\partial V_j = \partial V'_j = \emptyset$ for $j > 1$. We thus have $\tilde{G}_i = \tilde{G}_1 = \tilde{G}$ and $\tilde{G}'_i = \tilde{G}'_1 = \tilde{G}'$ for any i. Therefore, all elements in the published sequence are the same, with value $\text{highDeg}_\tau\left(\tilde{G}\right)$ and $\text{highDeg}_\tau\left(\tilde{G}'\right)$ (or $\text{highOutDeg}_\tau\left(\tilde{G}\right)$ and $\text{highOutDeg}_\tau\left(\tilde{G}'\right)$).

Therefore,

$$\sum_{i=1}^{T}\left|\text{highDeg}_\tau\left(\tilde{G}_i\right) - \text{highDeg}_\tau\left(\tilde{G}_i\right)\right| = (\tilde{D}+1)T$$

or

$$\sum_{i=1}^{T}\left|\text{highOutDeg}_\tau\left(\tilde{G}_i\right) - \text{highOutDeg}_\tau\left(\tilde{G}_i\right)\right|$$
$$= \max\{\tilde{D}_{\text{in}}+1, \tilde{D}_{\text{out}}-1\}\cdot T,$$

and we can conclude that the global sensitivity of publishing the whole sequence is at least $(\tilde{D}+1)T$ (or $\max\{\tilde{D}_{\text{in}}+1, \tilde{D}_{\text{out}}-1\}\cdot T$). $\qquad\square$

C. Baselines for Publishing the Number of Edges

Similar to the algorithms for publishing the number of high-degree node, we have four baselines for publishing the number of edges.

As has been mentioned in Section III, there are two baseline algorithms for high-degree nodes count. Combining with the algorithm proposed in [24], there are four baseline algorithms for both undirected and directed graph sequences.

For an undirected graph sequence $(G_1, \ldots, G_T)$,

1) COMPOSE-D-BOUNDED: For each $i \in [T]$, run the global sensitivity algorithm to publish $\text{numedge}(G_i)$ with privacy parameter ϵ/T and D-bounded global sensitivity D (Lemma D.8).
2) COMPOSE-PROJECTION: For each $i \in [T]$, run Algorithm 2 on G_i with projection parameter $\tilde{D}$ to get $G_i^{\tilde{D}}$. Run the global sensitivity algorithm to publish $\text{numedge}\left(G_i^{\tilde{D}}\right)$ with privacy parameter ϵ/T and global sensitivity $\tilde{D}$ (Lemma D.9).
3) SENSSEQ-D-BOUNDED: Run the global sensitivity algorithm to publish $(\text{numedge}(G_i))_{i=1}^{T}$ with privacy parameter ϵ and D-bounded global sensitivity.
4) SENSSEQ-PROJECTION: Run Algorithm 3 on $(G_i)_{i=1}^{T}$ with projection parameter $\tilde{D}$ to get $\left(G_i^{\tilde{D}}\right)_{i=1}^{T}$. Run the global sensitivity algorithm to publish $\left(\text{numedge}\left(G_i^{\tilde{D}}\right)\right)_{i=1}^{T}$ with privacy parameter ϵ and computed global sensitivity.

For a directed graph sequence $(G_1, \ldots, G_T)$,

1) COMPOSE-D-BOUNDED: For each $i \in [T]$, run the global sensitivity algorithm to publish $\text{numedge}(G_i)$ with privacy parameter ϵ/T and

$(D_{\text{in}}, D_{\text{out}})$-bounded global sensitivity $D_{\text{in}} + D_{\text{out}}$ (Lemma D.8).
2) COMPOSE-PROJECTION: For each $i \in [T]$, run Algorithm 2 on G_i with projection parameter $\tilde{D}_{\text{in}}$, $\tilde{D}_{\text{out}}$ to get $G_i^{\tilde{D}_{\text{in}}, \tilde{D}_{\text{out}}}$. Run the global sensitivity algorithm to publish $\text{numedge}\left(G_i^{\tilde{D}_{\text{in}}, \tilde{D}_{\text{out}}}\right)$ with privacy parameter ϵ/T and global sensitivity $\tilde{D}_{\text{in}} + \tilde{D}_{\text{out}}$ (Lemma D.9).
3) SENSSEQ-D-BOUNDED: Run the global sensitivity algorithm to publish $(\text{numedge}(G_i))_{i=1}^{T}$ with privacy parameter ϵ and D_{in}-in-bounded global sensitivity.
4) SENSSEQ-PROJECTION: Run Algorithm 3 on $(G_i)_{i=1}^{T}$ with projection parameter $\tilde{D}_{\text{in}}$, $\tilde{D}_{\text{out}}$ to get $\left(G_i^{\tilde{D}_{\text{in}}, \tilde{D}_{\text{out}}}\right)_{i=1}^{T}$. Run the global sensitivity algorithm to publish $\left(\text{numedge}\left(G_i^{\tilde{D}_{\text{in}}, \tilde{D}_{\text{out}}}\right)\right)_{i=1}^{T}$ with privacy parameter ϵ and computed global sensitivity.

Similar to the results for number of high-degree nodes, it is not hard to see that for both undirected and directed graphs, (3) is worse than (1); and (4) is worse than (2). Therefore, we only need to run (1) and (2), i.e., COMPOSE-D-BOUNDED and COMPOSE-PROJECTION.

1) Analyses for COMPOSE-D-BOUNDED*:*

Lemma D.8. *Given an undirected graph $G = (V, E)$, the D-bounded global sensitivity of publishing $\text{numedge}(G)$ is D.*
Given an directed graph $G = (V, E)$, the (D_{in}, D_{out})-bounded global sensitivity of publishing $\text{numedge}(G)$ is $D_{in} + D_{out}$.

Proof. For undirected graph, suppose we add an additional node v^* to $G = (V, E)$ with d adjacent edges. This does not affect any edge in E, but only adds $d \leq D$ edges to the graph. And thus $\text{numedge}(G)$ changes by at most D. When $d = D$, the change is exactly D.

Similar holds for directed graph. A node v^* with D_{in}-in-edges and D_{out}-out-edges can increase the total number of edges by $D_{\text{in}} + D_{\text{out}}$. $\qquad\square$

2) Analyses for COMPOSE-PROJECTION*:*

Lemma D.9. *Using Algorithm 2 on an undirected graph G to get $G^{\tilde{D}}$ and publishing $\text{numedge}\left(G^{\tilde{D}}\right)$ has global sensitivity $\tilde{D}$.*
Using Algorithm 2 on an directed graph G to get $G^{\tilde{D}_{in}, \tilde{D}_{out}}$ and publishing $\text{numedge}\left(G^{\tilde{D}_{in}, \tilde{D}_{out}}\right)$ has global sensitivity $\tilde{D}_{in} + \tilde{D}_{out}$.

Proof. We follow the same analysis as in Lemma D.3 and D.4.

For undirected graph G, for every additional edge (v^*, v_1) adjacent to v^*, we know from the proof of Lemma D.3 that it yields an "alternating sequence" of

edges such that all odd positioned edges are added while all even positioned edges are deleted from the projected graph without (v^*, v_1). So the number of edges can only increase by 1 or decrease by 1. Since there are at most $\tilde{D}$ edges adjacent to v^* that are added in the projected graph, the number of edges changes by at most $\tilde{D}$.

For directed graph, the same analysis holds. For every in-edge (v_1, v^*) or every out-edge (v^*, v_1), there is an "alternating sequence" of edges such that that all odd positioned edges are added while all even positioned edges are deleted from the projected graph. So the number of edges changes by at most $\tilde{D}_{\text{in}} + \tilde{D}_{\text{out}}$.

For both undirected and directed graph, it is easy to see that the number of edges change by exactly $\tilde{D}$ and $\tilde{D}_{\text{in}} + \tilde{D}_{\text{out}}$ if G does not contain any edge. $\qquad\square$

APPENDIX E
OTHER

Lemma E.1. *Suppose we use* SENSDIFF *to publish* $\{\sum_{s=1}^{t} \tilde{\Delta}_s\}_{t=1}^{T}$. *For any $t \leq T$, the standard deviation of $\sum_{s=1}^{t} \tilde{\Delta}_s - f(G_t)$ is of order $O\left(\sqrt{t}\right)$.*

Proof. $\sum_{s=1}^{t} \tilde{\Delta}_s - f(G_t) = \sum_{s=1}^{t} \text{Lap}\left(\frac{\text{GS}_D(\Delta)}{\epsilon}\right)$. The sum of these t i.i.d. Laplace random variables follows the same distribution as the difference between two Gamma random variables with scale parameters $\frac{\text{GS}_D(\Delta)}{\epsilon}$ and shape parameters t. Therefore, to analyze the variance of the difference, we only need to analyze the variance of $X - Y$, where $X, Y \sim \Gamma(t, a)$ with $a = \frac{\text{GS}_D(\Delta)}{\epsilon}$. Since X and Y are independent, we have $\text{Var}\left(X - Y\right) = \text{Var}\left(X\right) + \text{Var}\left(Y\right) = 2ta^2$, and thus the standard deviation of $\sum_{s=1}^{t} \tilde{\Delta}_s - f(G_t)$ is $\sqrt{2}\frac{\text{GS}_D(\Delta)}{\epsilon}\sqrt{t}$. $\qquad\square$

On the other hand, it is easy to see that COMPOSE-D-BOUNDED adds noise with standard deviation $\sqrt{2}\frac{\text{GS}_D(f)}{\epsilon/T} = \sqrt{2}\frac{\text{GS}_D(f)}{\epsilon}T$.