

# N2E: A General Framework to Reduce Node-Differential Privacy to Edge-Differential Privacy for Graph Analytics

YIHUA HU, Nanyang Technological University, Singapore

HAO DING, Nanyang Technological University, Singapore

WEI DONG\*, Nanyang Technological University, Singapore

Differential privacy (DP) has been widely adopted to protect sensitive information in graph analytics. While edge-DP, which protects privacy at the edge level, has been extensively studied, node-DP, offering stronger protection for entire nodes and their incident edges, remains largely underexplored due to its technical challenges. A natural way to bridge this gap is to develop a general framework for reducing node-DP graph analytical tasks to edge-DP ones, enabling the reuse of existing edge-DP mechanisms. A straightforward solution based on group privacy divides the privacy budget by a given degree upper bound, but this leads to poor utility when the bound is set conservatively large to accommodate worst-case inputs. To address this, we propose node-to-edge (N2E), a general framework that reduces any node-DP graph analytical task to an edge-DP one, with the error dependency on the graph's true maximum degree. N2E introduces two novel techniques: a distance-preserving clipping mechanism that bounds edge distance between neighboring graphs after clipping, and the first node-DP mechanism for maximum degree approximation, enabling tight, privacy-preserving clipping thresholds. By instantiating N2E with existing edge-DP mechanisms, we obtain the first node-DP solutions for tasks such as maximum degree estimation. For edge counting, our method theoretically matches the error of the state-of-the-art, which is provably optimal, and significantly outperforms existing approaches for degree distribution estimation. Experimental results demonstrate that our framework achieves up to a 2.5× reduction in error for edge counting and up to an 80× reduction for degree distribution estimation.

CCS Concepts: • **Information systems** → **Data management systems**; • **Security and privacy** → **Database and storage security**.

Additional Key Words and Phrases: Differential privacy, SJA query processing

## ACM Reference Format:

Yihua Hu, Hao Ding, and Wei Dong. 2025. N2E: A General Framework to Reduce Node-Differential Privacy to Edge-Differential Privacy for Graph Analytics. *Proc. ACM Manag. Data* 3, 6 (SIGMOD), Article 343 (December 2025), 26 pages. <https://doi.org/10.1145/3769808>

## 1 Introduction

In the era of big data, graph analytics has become essential across various domains such as social networks, healthcare, and finance, yielding significant societal and economic benefits. For instance, social platforms such as Facebook and LinkedIn leverage graph analytics to support features like friend and professional connection recommendations. However, these graphs often contain sensitive information, and publishing analytical results can pose serious privacy risks, such as revealing

\*Wei Dong is the corresponding author.

Authors' Contact Information: Yihua Hu, Nanyang Technological University, Singapore, Singapore, yihua001@e.ntu.edu.sg; Hao Ding, Nanyang Technological University, Singapore, Singapore, hao028@e.ntu.edu.sg; Wei Dong, Nanyang Technological University, Singapore, Singapore, wei\_dong@ntu.edu.sg.



This work is licensed under a Creative Commons Attribution 4.0 International License.

© 2025 Copyright held by the owner/author(s).

ACM 2836-6573/2025/12-ART343

<https://doi.org/10.1145/3769808>

specific edges or exposing node attributes [38, 45, 49]. To address these concerns, *differential privacy* (DP) [17] has emerged as a rigorous standard for privacy-preserving graph analytics. Under DP, two graphs are considered neighbors if they differ by a single individual, represented either as a node or an edge. DP ensures that the analytical results between two neighboring graphs are  $(\epsilon, \delta)$ -indistinguishable, preventing the inference of any individual's information. The privacy budget  $\epsilon$  controls privacy leakage, with smaller values providing stronger privacy guarantees, while  $\delta$  controls the failure probability, quantifying the likelihood that the privacy guarantee does not hold.

A key distinction in applying DP to graph analytics lies in the choice of protecting either edges or nodes. Specifically, *edge-DP* [25, 32, 43] protects individual edges, while *node-DP* [5, 7, 33] protects entire nodes with its incident edges. Accordingly, two graphs are neighbors under edge-DP if they differ by a single edge, whereas under node-DP, they differ by a node and all its edges. Although node-DP offers stronger privacy guarantees, it is more challenging to achieve since a node has a greater contribution to the analytical result than a single edge. This leads to a significant research gap between node-DP and edge-DP. Many graph analytical tasks, such as maximum degree estimation [10, 25, 32], and shortest path computation [21, 44] have been extensively studied under edge-DP but lack corresponding solutions under node-DP. Yet, node-DP is more meaningful in practice, as it better aligns with real-world privacy concerns, where users (nodes) and their associated relationships (edges) should be protected as a whole rather than in isolation.

Therefore, a well-motivated objective is to develop a general framework for reducing node-DP tasks to edge-DP. More precisely, given any graph task  $Q$  and an edge-DP mechanism  $\mathcal{M}_Q^{\text{edge}}$ , this framework produces a corresponding node-DP protocol  $\mathcal{M}_Q^{\text{node}}$ . Recall that the key challenge in achieving node-DP, as opposed to edge-DP, lies in the greater impact of a single node on the graph structure. However, this impact can be naturally bounded, as each node is incident to at most  $O(\widehat{N})$ <sup>1</sup> edges, where  $\widehat{N}$  is a predefined upper bound on the number of nodes  $N$  in the graph. Consequently, protecting an edge's privacy also protects part of the associated node's privacy. This principle is formally captured in the DP literature as *group privacy* [19] (see Lemma 3.4 for a formal definition). Intuitively, since two neighboring graphs under node-DP have an *edge distance*, which is defined as the number of differing edges, bounded by  $O(\widehat{N})$ , any  $(\epsilon/\widehat{N}, \delta/\widehat{N})$ -edge-DP protocol naturally follows  $(\epsilon, \delta)$ -node-DP.

However, this approach always breaks utility in practice. Since most edge-DP mechanisms exhibit errors that grow linearly with  $1/\epsilon$ , the group privacy solution amplifies the error by a factor of  $\widehat{N}$ . In practice,  $\widehat{N}$  should be set conservatively large to ensure coverage of all possible graphs, leading to a large degradation in utility. Take edge counting as an example, applying group privacy with a state-of-the-art edge-DP protocol results in an error of  $O(\widehat{N}/\epsilon)$ . Instead, the state-of-the-art node-DP solution achieves an error of  $\tilde{O}(\text{deg}(G)/\epsilon)$  [11, 22], where  $\text{deg}(G)$  is the maximum degree of nodes in the graph  $G$ . In real-world scenarios,  $\text{deg}(G)$  is typically much smaller than  $\widehat{N}$ . For example, the average Facebook user has around 130 friends, far fewer than the total number of users of the platform.

The bad performance of the group privacy approach stems from its dependency on  $\widehat{N}$ , the worst-case upper bound on a node's contribution. A more preferable solution is to use the actual maximum node's contribution in the graph  $G$ , i.e.,  $\text{deg}(G)$ . However,  $\text{deg}(G)$  itself is highly sensitive, and using it as a parameter to scale randomness with group privacy would break privacy guarantees [11, 16, 28]. A common strategy to mitigate worst-case error dependency is the *clipping*

<sup>1</sup>We use  $O(\cdot)$  for standard asymptotic bounds and  $\tilde{O}(\cdot)$  to suppress logarithmic factors. All logarithms are written as  $\log$  and treated equivalently in  $O(\cdot)$ .

*mechanism* [2, 28]. By clipping the input graph with a degree upper bound  $\tau$ , the resulting node-DP error shifts its dependency from  $\widehat{N}$  to  $\tau$ . Moreover, when  $\tau$  is set close to  $\deg(G)$ , we achieve the error dependency that improves from  $\widehat{N}$  to  $\deg(G)$ . For example, in the node-DP edge counting problem, we can achieve the state-of-the-art error  $O(\deg(G)/\epsilon)$ . However, this approach introduces two key challenges.

**Challenge 1. Distance-preserving clipping mechanism.** The clipping mechanism needs to be distance-preserving, meaning that for a given degree upper bound  $\tau$ , the edge distance of two neighboring graphs after clipping remains upper bounded by a function of  $\tau$ . However, no existing clipping mechanism satisfies this requirement under node-DP. To illustrate this, consider the most straightforward clipping mechanism that deletes all nodes with degrees higher than  $\tau$ . Such a clipping mechanism fails to meet the distance-preserving requirement under node-DP. Consider a graph  $G$  where all  $N$  nodes have degree  $\tau$ , and construct a neighboring graph  $G'$  by adding a new node that connects to all  $N$  nodes in  $G$ . After applying the clipping mechanism,  $G$  remains unchanged, but  $G'$  becomes empty. This results in an edge distance of  $N \cdot \tau/2$ , which violates the distance-preserving requirement.

**Challenge 2. Node-DP maximum degree estimation.** To achieve an error dependency on  $\deg(G)$ , we should estimate a  $\tau$  as close to  $\deg(G)$  as possible. However, maximum degree estimation under node-DP remains an open problem. First, extending edge-DP maximum degree estimators to support node-DP is fundamentally infeasible. Under edge-DP, each edge affects the maximum degree by at most 1, so adding  $O(1/\epsilon)$  noise is sufficient to mask its contribution. In contrast, under node-DP, a single node can influence the maximum degree by up to  $\widehat{N}$ , requiring noise of scale  $O(\widehat{N}/\epsilon)$  to ensure privacy. However, this completely destroys utility, as the maximum degree is at most  $\widehat{N}$ . This challenge can be further reflected in the fact that many node-DP techniques [2, 7, 33, 42, 47] assume a predefined degree upper bound rather than proposing a method to estimate it.

**Problem Statement.** These challenges raise a question: *Is there a general framework to reduce node-DP tasks to edge-DP with an error upgraded at most by  $\widehat{O}(\deg(G))$ ?*

## 1.1 Our Results

This paper answers this question affirmatively. First, to address **Challenge 1**, we propose a distance-preserving clipping mechanism under node-DP. Specifically, given a degree upper bound  $\tau$ , the edge distance of any two neighboring graphs after clipping is bounded by  $\tau + k$ , where  $k$  is the number of nodes in  $G$  with degree at least  $\tau$ . When  $\tau$  approaches  $\deg(G)$ ,  $k$  remains small, satisfying the distance-preserving requirement. Furthermore, the proposed clipping mechanism is computationally efficient, operating in linear time.

Next, to address **Challenge 2**, we propose our node-DP protocol for maximum degree approximation. For any input graph  $G$ , the mechanism outputs a  $\tau^*$ , which can be shown to be a good approximation of  $\deg(G)$ . On one hand,  $\tau^*$  is not much smaller than  $\deg(G)$  with a bounded rank error: there are at most  $\widehat{O}(1/\epsilon)$  nodes<sup>2</sup> in  $G$  with degree at least  $\tau^*$ . On the other hand,  $\tau^*$  is at most  $O(\deg(G)) + \widehat{O}(1/\epsilon)$ . Our node-DP maximum degree approximator relies on solving  $O(\log \deg(G))$  number of linear programs, resulting in a polynomial-time complexity.

By combining these two proposed mechanisms, we develop our framework *node-to-edge* (N2E), a framework that can reduce an arbitrary node-DP task to an edge-DP task. N2E works as follows. Given any graph analytical task  $Q$  and an edge-DP mechanism  $\mathcal{M}_Q^{\text{edge}}$ , for any graph  $G$ , we first compute  $\tau^*$  as the approximation of  $\deg(G)$ . Then, we clip the graph using our distance-preserving clipping mechanism with degree upper bound  $\tau^*$  to obtain the clipped graph  $\overline{G}$ . Finally, this

<sup>2</sup>All results in the introduction hold with constant probability.

Task	Our Results	SOTA Edge-DP	SOTA Node-DP
Edge Counting	$O(\deg(G) \log \log \deg(G) + \log \frac{1}{\delta})$	$O(1)$	$O(\deg(G) \log \log \deg(G))$ [12]
Maximum Degree Estimation	$O(\deg(G) \log \log \deg(G) + \log \deg(G) \log \frac{1}{\delta})$	$O(1)$	$O(\bar{N})$
Degree Distribution	$O(\deg(G)^{1.5} + \deg(G)^{0.5} \log \frac{1}{\delta})$	$O(\deg(G)^{0.5})$	$O(\bar{N}^{1.5})$ [7]

Table 1. Summary of node-DP error on common graph analytical tasks.<sup>3</sup>Here, SOTA stands for state-of-the-art. [7] also proposes a mechanism that relies on a heuristic solution to estimate  $\deg(G)$ , but it lacks formal utility guarantees. We include this heuristic as a baseline in our experiments. We use  $d_{\text{edge}}(G, G')$  to denote the edge distance between  $G$  and  $G'$ .  $\varepsilon$  is regarded as a constant and hidden in the  $O$ -notation.

framework invokes the edge-DP mechanism  $\mathcal{M}_Q^{\text{edge}}$  on the clipped graph, with privacy budgets  $\varepsilon' = \varepsilon/O(\tau^*)$  and  $\delta' = \delta/O(\tau^*)$ .

N2E achieves an error of:

$$|Q(G) - Q(\bar{G})| + \text{Err}_Q^{\text{edge}}(\bar{G}, \varepsilon', \delta'),$$

where  $\bar{G}$  is obtained by clipping edges of  $O(\log \log \deg(G)/\varepsilon)$  nodes, and  $\text{Err}_Q^{\text{edge}}$  denotes the error of the edge-DP mechanism on the task  $Q$ . Such an error consists of two components. The first component is the clipping bias, which arises from the clipping process. This component can be further shown to be optimal (see Section 4.4 for details). The second component results from the edge-DP mechanism. Since  $\bar{G}$  is derived by clipping only a small number of nodes from  $G$ , most edge-DP protocols exhibit similar errors on both  $G$  and  $\bar{G}$  under the same parameters. Moreover, since  $\varepsilon$  is usually regarded as a constant in the DP literature, the second component aligns with our target error bound:  $\text{Err}_Q^{\text{edge}}(G, \varepsilon/\tilde{O}(\deg(G)), \delta/\tilde{O}(\deg(G)))$ . We will show our results upgrade the edge-DP error by  $\tilde{O}(\deg(G))$  for the most common graph analytical tasks in Table 1. For edge counting, our result matches the state-of-the-art node-DP error, up to an additional additive term of  $O(\log(1/\delta))$ . For degree distribution, our result reduces the error from  $O(\bar{N}^{1.5})$  to  $O(\deg(G)^{1.5} + \tilde{O}(\deg(G)^{0.5}))$ . For maximum degree estimation, we propose the first node-DP solution: we return the maximum degree of  $G'$ , where  $G'$  deletes  $\tilde{O}(\deg(G))$  edges from  $G$ .

Experimental results<sup>4</sup> show that N2E achieves up to a  $2.5\times$  error reduction compared to state-of-the-art methods in edge counting, and up to a  $80\times$  reduction compared to the existing state-of-the-art method in degree distribution. Moreover, N2E delivers high utility as the first node-DP solution for maximum degree estimation, achieving a relative rank error of around 1% in most experiments.

## 2 Related Work

DP graph analytics has been extensively studied, with early works focusing primarily on graph pattern counting queries. Most of these studies adopt edge-DP and target specific graph patterns. [40] proposed *smooth sensitivity*, which enables private triangle counting. [32] extended this approach to  $k$ -star counting and introduced *higher-order local sensitivity* for  $k$ -triangle counting. [46] proposed the ladder function, applicable to  $k$ -triangle and  $k$ -clique counting. [31] introduced *elastic sensitivity*, the first edge-DP mechanism supporting arbitrary graph pattern counting. More recently,

<sup>3</sup>Our results satisfy  $(\varepsilon, \delta)$ -node-DP, while the node-DP SOTA methods satisfy pure  $\varepsilon$ -node-DP. We set  $\delta$  to a negligible value of  $2^{-30}$  to ensure minimal practical impact. Importantly, our N2E framework supports transforming arbitrary node-DP tasks into edge-DP ones, which the SOTA methods cannot achieve.

<sup>4</sup>Code is available at <https://github.com/Chronomia/N2E>.

[14] proposed *residual sensitivity*, which also supports arbitrary patterns and has been shown to achieve near-optimal error [15]. Besides, some works study edge-DP graph pattern counting in dynamic settings [10, 24] or decentralized settings [20, 26, 29, 30], which is different from our setting. Moving towards node-DP, [33] and [2] proposed solutions for arbitrary graph pattern counting, albeit under strong assumptions on node degrees. [5] introduced the recursive mechanism, which achieves high utility but suffers from high computational cost. More recently, [11, 12] proposed efficient mechanisms that achieve both high utility and computational efficiency under node-DP. [13] studied how to answering multiple queries. Besides graph pattern counting, another widely studied problem is degree distribution. In this context, several works have investigated edge-DP [1, 6, 41, 48], while [7] studies the problem under node-DP. In these two problems, as demonstrated in our experiments, integrating our framework with edge-DP solutions leads to an improvement in error.

For other graph analytical tasks, research has primarily focused on the edge-DP setting. For maximum degree estimation, only edge-DP solutions have been proposed so far. The Laplace mechanism achieves an additive error, while [22] introduced the *shifted inverse mechanism*, which achieves a bounded rank error but requires exponential runtime. Besides, densest subgraph estimation [9, 23, 39], and shortest path computation [4, 8, 21, 44], have been studied extensively under edge-DP but still lack corresponding solutions under node-DP. For synthetic graph generation, a recent survey [35] reviewed 16 works, of which 14 focus on edge-DP and only 2 address node-DP.

### 3 Preliminaries

#### 3.1 Notation

We denote a graph as  $G = (V, E)$ , where  $V$  is the set of nodes and  $E$  is the set of edges. In this paper, we focus on undirected graphs. Each edge is denoted as a pair  $(u, v)$ , where  $u$  and  $v$  are the endpoint nodes of the edge. Let  $N = |V|$  and  $M = |E|$  denote the number of nodes and edges in a graph  $G$ , respectively.  $\widehat{N}$  is a predefined upper bound of  $N$ . For a node  $v \in V$  in graph  $G$ ,  $E(v)$  represents the set of edges incident to  $v$ , and  $\deg_G(v)$  represents the degree of  $v$ . The maximum degree in  $G$  is given by  $\deg(G) = \max_{v \in V} \deg_G(v)$ . We denote  $\deg^k(G)$  as the  $k$ -th largest degree of  $G$ , and  $N_\tau(G)$  as the number of nodes in  $G$  with degree  $\geq \tau$ . The graph dataset  $\mathcal{G}$  is a finite collection of undirected graphs.

For two graphs  $G$  and  $G'$ , the edge distance  $d_{\text{edge}}(G, G')$  is the minimum number of edge insertions or deletions required to transform  $G$  into  $G'$ . We write  $G \sim_{(u,v)} G'$  to denote  $d_{\text{edge}}(G, G') = 1$ , where  $(u, v)$  is the edge that differs between  $G$  and  $G'$ . Similarly, we define the node distance  $d_{\text{node}}(G, G')$  and write  $G \sim_v G'$  if  $G$  and  $G'$  differ by exactly one node. We write  $G \subseteq G'$  if  $G'$  contains  $G$ , i.e.,  $G$  can be obtained by deleting a subset of nodes from  $G'$  along with all edges connected to the nodes in the subset.

#### 3.2 Differential Privacy

*Definition 3.1 (Differential Privacy).* A randomized mechanism  $\mathcal{M} : \mathcal{G} \rightarrow \mathcal{R}$  satisfies  $(\epsilon, \delta)$ -differential privacy if, for any two neighboring graphs  $G, G' \in \mathcal{G}$ , and for any subset of outputs  $\mathcal{S}$  in the output space of mechanism  $\mathcal{M}$ , the following holds:

$$\Pr[\mathcal{M}(G) \in \mathcal{S}] \leq e^\epsilon \Pr[\mathcal{M}(G') \in \mathcal{S}] + \delta.$$

$\epsilon$  is usually a constant between 0.1 to 10 while  $\delta \ll 1/N$ . If  $\delta = 0$ , the mechanism satisfies  $(\epsilon, 0)$ -DP or simply  $\epsilon$ -DP.

In the context of graph data, two graphs are considered neighboring if one can be obtained by deleting a single edge from the other, denoted as  $G \sim_{\text{edge}} G'$ , or by deleting a single node along

with its incident edges, denoted as  $G \sim_{\text{node}} G'$ . These two notions, referred to as edge-DP and node-DP [25], provide privacy guarantees at different levels of granularity. With such definitions, under node-DP, the number of nodes  $N$  is private.

Differential privacy shares some desired properties:

**LEMMA 3.2 (POST-PROCESSING).** *Let  $\mathcal{M} : \mathcal{G} \rightarrow \mathcal{R}$  be a randomized mechanism that satisfies  $(\epsilon, \delta)$ -differential privacy, and let  $\mathcal{M}' : \mathcal{R} \rightarrow \mathcal{Z}$  be any deterministic or randomized mechanism. Then,  $\mathcal{M}'(\mathcal{M}(G))$  satisfies  $(\epsilon, \delta)$ -differential privacy for any graph  $G \in \mathcal{G}$ .*

**LEMMA 3.3 (SEQUENTIAL COMPOSITION).** *Let  $\mathcal{M}_1, \mathcal{M}_2, \dots, \mathcal{M}_k$  be a series of randomized mechanisms, where for each  $i \in \{1, 2, \dots, k\}$ ,  $\mathcal{M}_i : \mathcal{G} \rightarrow \mathcal{R}_i$  satisfies  $(\epsilon, \delta)$ -differential privacy. The composition mechanism defined as*

$$\mathcal{M}(G) = (\mathcal{M}_1(G), \dots, \mathcal{M}_k(G))$$

*satisfies  $(k\epsilon, k\delta)$ -differential privacy.*

**LEMMA 3.4 (GROUP PRIVACY).** *If a mechanism  $\mathcal{M}$  satisfies  $(\epsilon, \delta)$ -differential privacy, then for any two graphs  $G$  and  $G'$  with  $d(G, G') = \lambda$  (i.e., differing by  $\lambda$  edges or nodes), and for any subset of outputs  $\mathcal{S} \subseteq \mathcal{R}$ , the following holds:*

$$\Pr[\mathcal{M}(D) \in \mathcal{S}] \leq e^{\lambda\epsilon} \Pr[\mathcal{M}(D') \in \mathcal{S}] + \lambda\delta.$$

*The mechanism  $\mathcal{M}$  satisfies  $(\lambda\epsilon, \lambda\delta)$ -differential privacy for the group of  $\lambda$  edges or nodes.*

### 3.3 Common DP Mechanisms

One of the most commonly used DP mechanisms is the *Laplace mechanism*. Here, given a query  $Q : \mathcal{G} \rightarrow \mathcal{R}$ , its *global sensitivity* is  $\text{GS}_Q = \max_{d(G, G')=1} \|Q(G) - Q(G')\|_1$ . For edge-DP and node-DP, we define  $d(G, G')$  as  $d_{\text{edge}}$  and  $d_{\text{node}}$ , respectively.

**Definition 3.5 (Laplace Mechanism [17]).** Let  $Q : \mathcal{G} \rightarrow \mathcal{R}$  be a query function with global sensitivity  $\text{GS}_Q$ . The mechanism

$$\mathcal{M}_Q(G) = Q(G) + \text{Lap}\left(\frac{\text{GS}_Q}{\epsilon}\right)$$

preserves  $\epsilon$ -DP, where  $\text{Lap}(\text{GS}_Q/\epsilon)$  is a random variable drawn from the Laplace distribution with scale of  $\text{GS}_Q/\epsilon$ .

Besides the Laplace mechanism, another widely used DP mechanism is the *sparse vector technique* (SVT). Given a (potentially infinite) sequence of queries  $Q_\ell : \mathcal{G} \rightarrow \mathcal{R}$  ( $\ell = 1, 2, \dots$ ) with  $\text{GS}_{Q_\ell} = 1$ , SVT identifies the index of the first query whose result exceeds a threshold  $\tilde{T}$ . The detailed procedure is shown in Algorithm 1.

In the original version of SVT [18], the parameter  $c = 2$ , meaning that the noise added to each query result is twice the scale of the noise added to the threshold. Very recently, [36] points out that in specific cases, the noise added to the query result can be reduced by half. To formalize this, we first introduce the definition of sensitivity-monotonicity.

**Definition 3.6.** A function  $f(G)$  is *sensitivity-monotonic* if: Given any two neighboring graphs  $G \sim G'$ , if  $G \subseteq G'$ , then  $f(G) \geq f(G')$ .

Then, the following lemma holds for SVT:

**LEMMA 3.7 ([36]).** *If the sequence of queries  $\{Q_\ell\}$  used in SVT are sensitivity-monotonic, given any  $\epsilon > 0$ , by setting  $c = 1$ , SVT preserves  $\epsilon$ -DP. Moreover, given an error failure probability  $\beta$ , if there*

**Algorithm 1:** SVT

---

**Input:** Graph  $G$ , threshold  $T$ , a sequence of queries  $\{Q_\ell(G)\}$ , privacy budget  $\varepsilon$   
**Output:** The first index  $\ell$  where the query result exceeds  $\tilde{T}$   
 /\* Compute the noisy threshold \*/

```

1  $\tilde{T} \leftarrow T + \text{Lap}(2/\varepsilon)$ 
2 for  $\ell = 1, 2, \dots$  do
3    $\tilde{Q}_\ell(G) \leftarrow Q_\ell(G) + \text{Lap}(2c/\varepsilon)$ 
4   if  $\tilde{Q}_\ell(G) > \tilde{T}$  then
5     return  $\ell$ 
6   end
7 end

```

---

exists some  $Q_k$  such that  $Q_k(G) \geq T + 4 \ln(2/\beta)/\varepsilon$ , then with probability at least  $1 - \beta$ , SVT returns an index  $\ell$  such that

$$\ell \leq k \quad \text{and} \quad Q_\ell(G) \geq T - \frac{4}{\varepsilon} \ln \frac{2k}{\beta}.$$

### 3.4 Optimality in Error of Node-DP Mechanisms

It is well known that the Laplace mechanism achieves *worst-case optimal* error [11], meaning it attains the optimal error when considering only the worst-case scenario. However, in graph data analytics, particularly under node-DP, worst-case optimal error is meaningless unless a strict predefined upper bound on each node's contribution to the analytical result, i.e.,  $\text{GS}_Q$  is provided. Take edge counting under node-DP as an example, where  $\text{GS}_Q = \hat{N}$ . Recall that  $\hat{N}$  is a predefined upper bound for the number of nodes. It should be set conservatively large enough to accommodate all possible input graphs.

To address this issue, *down neighborhood optimality* has been introduced to quantify the error for a node-DP mechanism [11].

*Definition 3.8 (Down Neighborhood Optimality).* Given query  $Q$ , and let  $\mathbb{M}$  denote the set of all  $(\varepsilon, \delta)$ -node-DP mechanisms, a node-DP mechanism  $\mathcal{M}_Q \in \mathbb{M}$  is  $(\rho, c)$ -down neighborhood optimal if, for any graph  $G$ ,

$$\Pr \left[ |\mathcal{M}_Q(G) - Q(G)| \leq c \cdot \mathcal{L}(G, \rho) \right] \geq \frac{2}{3},$$

where  $c$  is the optimality ratio, and

$$\mathcal{L}(G, \rho) := \min_{\mathcal{M}' \in \mathbb{M}} \max_{G' \subseteq G, d_{\text{node}}(G, G') \leq \rho} \left\{ \xi : \Pr[|\mathcal{M}'(G') - Q(G')| \leq \xi] \geq \frac{2}{3} \right\}.$$

Roughly speaking, for any graph  $G$ , we consider its  $\rho$ -down neighborhood, i.e.,  $G' \subseteq G$ ,  $d_{\text{node}}(G, G') \leq \rho$ . A mechanism  $\mathcal{M}_Q$  is said to be  $\rho$ -down neighborhood optimal if for every graph  $G$ ,  $\mathcal{M}_Q$  performs as well as the optimal mechanism  $\mathcal{M}'_Q$  that is specifically designed for  $G$  and its  $\rho$ -down neighborhood. It is clear that smaller  $\rho$  and  $c$  imply stronger optimality. When  $\rho = 0$ , this reduces to *instance optimality* with  $\mathcal{L}(G, 0) \equiv 0$ , since for any  $G$ , we can always find a tailor-made mechanism  $\mathcal{M}'(\cdot) = Q(G)$  which achieves 0 error on  $G$ .

$\mathcal{L}(G, \rho)$  can further be shown to be lower bounded by  $(\rho - 1)$ -downward query difference:

LEMMA 3.9 ([22]). *For any query  $Q$ , any  $\rho \geq 1$ , any  $\varepsilon \leq \ln 2$ , and any graph  $G$ , we have*

$$\mathcal{L}(G, \rho) \geq \frac{1}{2\rho} \Delta Q^{(\rho-1)}(G),$$

where

$$\Delta Q^{(\rho-1)}(G) = \max_{G' \subseteq G, d_{\text{node}}(G, G') \leq \rho} |Q(G) - Q(G')|$$

is the  $(\rho - 1)$ -downward query difference.

Therefore, if a node-DP mechanism achieves an error of  $c \cdot \Delta Q^{(\rho-1)}(G)$ , then it is  $(\rho, 2c\rho)$ -down neighborhood optimal. Taking edge counting under node-DP as example,  $\Delta Q^{(0)}(G) = \text{deg}(G)$ , which is much smaller than  $\widehat{N}$  for most graphs. By convention, a mechanism is said to be optimal if it is  $(\rho, 2c\rho)$ -down neighborhood optimal with  $\rho = \text{polylog}$  and  $c$  is a constant.

## 4 Methodology

In this section, we present our general framework, N2E, for reducing node-DP graph analytical tasks to edge-DP ones. We begin in Section 4.1 with a strawman solution based on group privacy. Such an idea suffers from utility issues due to its dependency on the node count upper bound  $\widehat{N}$ . However, it offers valuable insights for the design of our mechanism. To address these issues, we propose to use a clipping mechanism to bound individual node contributions based on a predefined clipping threshold. In Section 4.2, we introduce a distance-preserving clipping algorithm, followed by a node-DP maximum degree approximation algorithm in Section 4.3 to determine the clipping threshold. Finally, in Section 4.4, we combine these steps into a general framework that achieves instance-level error.

### 4.1 Strawman and Roadmap

A natural approach to reducing node-DP tasks to edge-DP is to leverage group privacy. Given a graph analytical task  $Q$ , an  $(\varepsilon, \delta)$ -edge-DP mechanism  $\mathcal{M}_Q^{\text{edge}}(G, \varepsilon, \delta)$ , an input graph  $G$ , and a node number upper bound  $\widehat{N}$  for  $G$ , group privacy (Lemma 3.4) guarantees that  $\mathcal{M}_Q^{\text{edge}}(G, \varepsilon/\widehat{N}, \delta/\widehat{N})$  satisfies  $(\varepsilon, \delta)$ -node-DP. However, this approach breaks the utility. Since most edge-DP mechanisms exhibit errors growing linearly with  $1/\varepsilon$ , dividing the privacy budget by  $\widehat{N}$  amplifies the error by a factor of  $\widehat{N}$ . As shown in the introduction, for edge counting, this approach yields an error of  $O(\widehat{N}/\varepsilon)$ . In contrast, the state-of-the-art node-DP mechanism for edge counting achieves an error of only  $\tilde{O}(\text{deg}(G)/\varepsilon)$ .

A common approach to address this issue is to employ the clipping mechanism. Here, we first clip  $G$  with a degree upper bound  $\tau$ , i.e., all nodes after the clipping should have a degree at most  $\tau$ . Besides, we require that after clipping, any two node-DP neighboring graphs should have an edge distance bounded by  $\tilde{O}(\tau)$ . Then, invoking the edge-DP mechanism on the clipped graph with privacy budgets divided by the edge distance we derived, we obtain a node-DP protocol. When  $\tau$  approaches  $\text{deg}(G)$ , we eventually reduce the error dependency from  $\widehat{N}$  to  $\text{deg}(G)$ .

Following the roadmap, we identify two main challenges, that have not been addressed under node-DP:

1. Design a distance-preserving clipping mechanism: Given any graph  $G$  and a degree upper bound  $\tau$ , after the clipping, two node-DP neighboring graphs have an edge distance bounded by  $g(G, \tau)$ . When  $\tau$  is close to  $\text{deg}(G)$ ,  $g(G, \tau)$  should be well bounded by  $\tilde{O}(\tau)$ .
2. Develop a node-DP mechanism to approximate the maximum degree.

### 4.2 Node-DP Distance-Preserving Clipping Mechanism

To address the first challenge in our roadmap, we first present a distance-preserving clipping mechanism under node-DP. So far, there are several distance-preserving clipping mechanisms working under edge-DP, but they can easily be shown to lose the distance-preserving property

under node-DP. The first example is a naive clipping algorithm that removes all edges from nodes with degree  $> \tau$ . Under edge-DP, this approach ensures that the edge distance between neighboring graphs after clipping is at most  $2\tau$ . However, under node-DP, such an upper bound does not hold. Suppose  $G$  is a graph where all  $N$  nodes have degree exactly  $\tau$ , and its neighboring graph  $G'$  is constructed by adding a new node connected to all  $N$  nodes in  $G$ . After clipping,  $G$  remains unchanged, while  $G'$  becomes an empty graph (since all nodes in  $G'$  have degree  $\tau + 1 > \tau$ ). Consequently, the edge distance between clipped  $G$  and  $G'$  is  $N \cdot \tau/2$ , which violates the distance-preserving requirement.

---

**Algorithm 2:** Distance-Preserving Graph Clipping
 

---

**Input:** Graph  $G = (V, E)$ , degree upper bound  $\tau$   
**Output:** Clipped graph  $\text{Clip}(G, \tau)$   
 /\* Assume a natural order of edges in  $E$  \*/

```

1  $\Lambda = \langle e_1, e_2, \dots, e_n \rangle$ 
2  $E' \leftarrow \emptyset$ 
3 for each node  $v \in V$  do
4    $E'_v \leftarrow \{e \in E(v) \mid \text{rank}_\Lambda(e \mid E(v)) \leq \tau\}$ 
5 end
6 for each edge  $e = (u, v) \in E$  do
7   if  $e \in E'_u$  and  $e \in E'_v$  then
8      $E' \leftarrow E' \cup \{e\}$ 
9   end
10 end
11 return  $\text{Clip}(G, \tau) = (V, E')$ 

```

---

Another commonly used edge-DP clipping mechanism is shown in Algorithm 2. Here, we first establish a natural ordering of edges. In practice, nodes in real-world graphs (e.g., social networks) typically have unique identifiers (e.g., user IDs), which can be used to define a natural node ordering. Edges are then ordered lexicographically by representing each edge with the lower-ordered node first. This edge ordering remains consistent between any pair of neighboring graphs and is also adopted in prior work such as [7]. Based on this ordering, the incident edges of each node form an ordered list, and the first  $\tau$  edges refer to the top  $\tau$  incident edges in that list. We retain an edge only if it appears among the first  $\tau$  edges of both its endpoints. Under edge-DP, this mechanism ensures that the edge distance is bounded by 3, as each edge, aside from itself, can only affect the existence of at most one edge for each of its endpoints. However, this distance-preserving property does not hold under node-DP. Consider the pair of neighboring graphs shown in Figure 1. Graph  $G$  is a cycle over  $N$  nodes  $v_1$  to  $v_N$ , while its neighboring graph  $G'$  adds a new node  $v_0$  connected to all  $N$  nodes in  $G$ . Nodes are ordered by their subscripts, i.e.,  $v_0 \prec v_1 \prec \dots \prec v_N$ , and edges follow the lexicographic order described above. When clipping with  $\tau = 2$ , the retained edges for both graphs are highlighted in blue in the figure. Graph  $G$  retains all edges, while  $G'$  retains only three:  $(v_0, v_1)$ ,  $(v_0, v_2)$ , and  $(v_1, v_2)$ . This results in a large edge distance of  $O(N)$  between  $\text{Clip}(G, 2)$  and  $\text{Clip}(G', 2)$ .

Even though Algorithm 2 is not distance-preserving for arbitrary values of  $\tau$ , an observation is that, given a graph  $G$ , when only a small number of nodes have degree at least  $\tau$ , the edge distance between  $G$  and any its node-DP neighboring graph  $G'$  remains well bounded after the clipping. More precisely, if  $\tau$  is the  $k$ -th largest degree of graph  $G$ , then the edge distance is bounded by

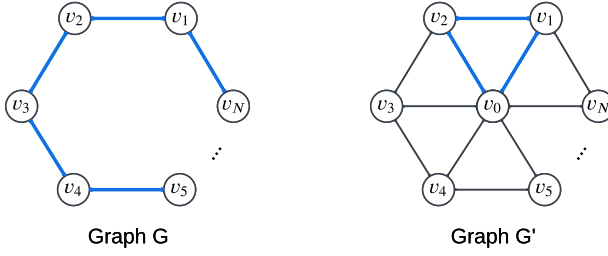


Fig. 1. An example of neighboring graphs where the distance-preserving property does not hold under node-DP.

$\tau + k$ . The high-level idea is that we classify all nodes in  $G$  into two classes: saturated ones, with degree  $\geq \tau$ , and unsaturated ones. The number of saturated nodes is  $k$ . Suppose  $G$ 's neighboring graph  $G'$  has an additional node  $u$ . When we do clipping,  $u$  retains at most  $\tau$  edges, introducing at most  $\tau$  additional edges in  $G'$ . Furthermore, each edge from  $u$  to a saturated node can replace at most one edge in  $G$ , resulting in at most  $k$  fewer edges in  $G'$ .

For example, let us consider two node-DP neighboring graphs  $G$  and  $G'$  as shown in Figure 2. Graph  $G$  consists of 1000 3-stars and 3 4-stars, and  $G'$  differs from  $G$  by adding a node  $u$  that connects to the center of every star in  $G$ . The saturated nodes of  $G$  are highlighted in red in the figure. Applying Algorithm 2 to clip both graphs with  $\tau = 4$ , we have  $k = N_\tau(G) = 3$ , since only 3 nodes in  $G$  are saturated with  $\geq 4$  edges. After clipping, the node  $u$  can retain up to 4 edges not present in  $G$ . Moreover, each 4-star center differs by at most 1 edge from its original 4 edges between  $G$  and  $G'$ , introducing an edge distance up to  $k = 3$ . As a result, the edge distance between  $G$  and  $G'$  after clipping is bounded by  $\tau + k = 7$ .

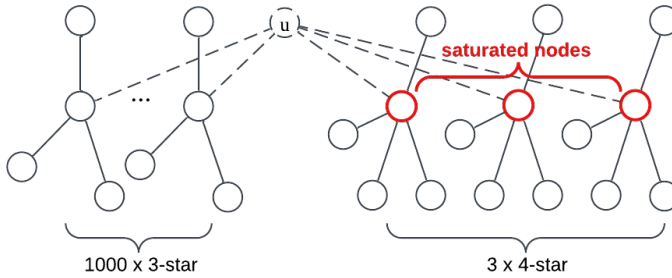


Fig. 2. Demonstration of the distance-preserving property for our node-DP clipping mechanism.

The formal statement is as below.

**LEMMA 4.1.** *Given any two neighboring graphs  $G$  and  $G'$  such that  $G \sim_{\text{node}} G'$ , let  $\tau = \deg^k(G)$ . The edge distance between  $\text{Clip}(G, \tau)$  and  $\text{Clip}(G', \tau)$  is bounded by  $\tau + k$ , where  $\text{Clip}(G, \tau)$  and  $\text{Clip}(G', \tau)$  are clipped graphs with Algorithm 2.*

**PROOF.** Without loss of generality, we assume  $G \subseteq G'$  and let the differing node be  $u$ . Let  $G = (V, E)$  and  $G' = (V', E')$ . We use  $\bar{G} = (\bar{V}, \bar{E})$  to denote  $\text{Clip}(G, \tau)$ , and use  $\bar{G}' = (\bar{V}', \bar{E}')$  to denote  $\text{Clip}(G', \tau)$ . By definition of  $\deg^k(G)$ ,  $G$  has at most  $k$  nodes with degree  $\geq \tau$ .

Consider all different edges between  $\bar{G}$  and  $\bar{G}'$ :

**1. The number of edges  $e$  that  $e \in \overline{E'}$  but  $e \notin \overline{E}$  is at most  $\tau$ .**

If there exists an edge  $e = (v_1, v_2) \in \overline{E'}$  and  $v_1, v_2 \in V$ , then  $e$  is among the  $\tau$  smallest-order edges for  $v_1$  and  $v_2$  in both  $G'$  and  $G$ , which indicates that  $e \in \overline{E}$ . Therefore, such edge  $e = (v_1, v_2)$  in this category has  $v_1 = u$  or  $v_2 = u$ . Since  $u$  can keep at most  $\tau$  edges after clipping, the number of this kind of edges is bounded by  $\tau$ .

**2. The number of edges  $e$  that  $e \in \overline{E}$  but  $e \notin \overline{E'}$  is at most  $k$ .**

For any such edge  $e = (v_1, v_2) \in \overline{E}$ ,  $e$  is among the  $\tau$  smallest-order for both  $v_1$  and  $v_2$  in  $G$ . The condition  $e \notin \overline{E'}$  occurs only if  $v_1$  or  $v_2$  is a saturated node in  $G$ , and a new edge  $e' = (u, v) \in E'$  where  $v \in \{v_1, v_2\}$  replaces  $e$  in  $\overline{G'}$  as one of the  $\tau$  smallest-order edges for that node. Since the number of saturated nodes in  $G$  is bounded by  $k$ , the number of edges in this category is at most  $k$ .

To conclude, the total edge difference between  $\overline{G}$  and  $\overline{G'}$  is bounded by  $\tau + k$ .  $\square$

Therefore, using  $\tau$  as the degree upper bound in Algorithm 2 to do the clipping can bound the edge distance between  $G$  and its neighboring graph by

$$g(G, \tau) = \tau + N_\tau(G) \quad (1)$$

where  $N_\tau(G)$  is how many nodes in  $G$  with degree at least  $\tau$ . Then, we can further invoke any edge-DP mechanism on the clipped graph with privacy budgets divided by  $g(G, \tau)$  to achieve node-DP.

However, this approach has two problems. First, the value of  $g(G, \tau)$  depends on how well  $\tau$  approximates  $\deg(G)$ . A  $\tau$  that is too small or too large yields a large  $g(G, \tau)$ . Second,  $g(G, \tau)$  is instance-specific and can differ significantly between  $G$  and its neighboring graph  $G'$ . For example, consider an empty graph  $G$ , and we insert one node connected to all nodes to obtain  $G'$ . Then,  $g(G, 1) = 1$  while  $g(G', 1) = N + 2$ . Therefore, directly using  $g(G, \tau)$  to allocate the privacy budget would break the privacy [40]. This provides key insights for designing our maximum degree approximation mechanism. Specifically, the output  $\tau^*$  should satisfy the following three properties:

1.  $\tau^*$  is not too large, i.e.,  $\tau^* = \tilde{O}(\deg(G))$ .
2.  $\tau^*$  is not too small so that  $N_{\tau^*}(G)$  can be well bounded.
3.  $\tau^*$  derives a DP-compliant upper bound of  $g(G, \tau^*)$  that can be used in group privacy.

Next, we explore approaches to determine  $\tau^*$  that satisfies all three properties while preserving node-DP.

### 4.3 Maximum Degree Estimation

Before discussing the node-DP solution to obtain a good  $\tau^*$  that satisfies the three listed properties, let us start with developing a corresponding edge-DP solution.

**4.3.1 Edge-DP Maximum Degree Estimation.** The most standard approach to realize edge-DP maximum degree estimation is to use the Laplace mechanism. Since any edge can influence the graph's maximum degree by up to 1, one can obtain an edge-DP maximum degree estimation by adding a noise of scale  $1/\epsilon$  to  $\deg(G)$ . However, the result estimation  $\tau^*$  fails to satisfy our second target property. For example, assume graph  $G$  has all  $N$  nodes have the same degree. Using the Laplace mechanism, the output  $\tau^*$  satisfies  $\tau^* \leq \deg(G)$  under a probability of  $1/2$ . Therefore, under the same probability,  $N_{\tau^*}(G) = N$ , as all  $N$  nodes have their degree higher than  $\tau^*$ .

This motivates the design of a new edge-DP maximum degree estimation mechanism that outputs  $\tau^*$  with the above 3 properties. Instead of using additive error to ensure privacy, we resort to another privacy-preserving algorithm, SVT, to produce the expected  $\tau^*$ . We propose the following query function used in the SVT:

$$Q_{\text{Del-Deg}}(G, \tau) = -\frac{1}{2} \sum_{\substack{v \in V \\ \deg_G(v) \geq \tau}} (\deg_G(v) - \tau). \quad (2)$$

Generally speaking, we sum the node degrees exceeding  $\tau$  in  $G$ , and  $Q_{\text{Del-Deg}}(G, \tau)$  denotes the opposite of half the degree sum.  $Q_{\text{Del-Deg}}(G, \tau)$  has three properties as below.

LEMMA 4.2.  $Q_{\text{Del-Deg}}(G, \tau)$  has the following properties:

1. **Monotonicity:**  $Q_{\text{Del-Deg}}(G, \tau)$  is monotonic and becomes 0 when  $\tau \geq \deg(G)$ .
2. **Bounded Sensitivity:**  $Q_{\text{Del-Deg}}(G, \tau)$  has  $\text{GS}_Q = 1$  under edge-DP.
3. **Sensitivity-Monotonicity:** For any two neighboring graphs  $G \sim_{\text{edge}} G'$  and any  $\tau$ ,  $Q_{\text{Del-Deg}}(G, \tau) \geq Q_{\text{Del-Deg}}(G', \tau)$  if  $G \subseteq G'$ .

PROOF. The proofs for all properties are provided below.

1. **Monotonicity:** The sum of node degrees exceeding  $\tau$  decreases as  $\tau$  increases. When  $\tau \geq \deg(G)$ , no node's degree exceeds  $\tau$ . Since  $Q_{\text{Del-Deg}}(G, \tau)$  is defined as  $-1/2$  times this sum, it increases with  $\tau$  and becomes 0 when  $\tau \geq \deg(G)$ .
2. **Bounded Sensitivity:** Adding or removing an edge influences the degree of two nodes by 1, thus altering the exceeding degree by at most 2. Therefore, the global sensitivity of  $Q_{\text{Del-Deg}}(G, \tau)$  is 1.
3. **Sensitivity Monotonicity:** If  $G'$  differs from  $G$  by one additional edge, then  $\deg_{G'}(v) \geq \deg_G(v)$  for all  $v \in V$ . Consequently,  $Q_{\text{Del-Deg}}(G', \tau) \leq Q_{\text{Del-Deg}}(G, \tau)$ .  $\square$

Using  $Q_{\text{Del-Deg}}(G, \tau)$  as the query function in SVT, we develop a new edge-DP maximum degree estimation mechanism as in Algorithm 3. The algorithm iteratively increases  $\tau$  until the noisy sum of node degrees exceeding  $\tau$  falls below a noisy threshold.

---

### Algorithm 3: Edge-DP Maximum Degree Estimation

---

**Input:** Graph  $G$ , privacy budget  $\epsilon$ , parameter  $\beta$

**Output:**  $\tau$

```

1  $T \leftarrow -4 \ln(2/\beta)/\epsilon$ 
2  $\tilde{T} \leftarrow T + \text{Lap}(2/\epsilon)$ 
3 for  $\tau = 1, 2, 3, 4, \dots$  do
4    $\tilde{Q}_\tau(G) \leftarrow Q_{\text{Del-Deg}}(G, \tau) + \text{Lap}(2/\epsilon)$ 
5   if  $\tilde{Q}_\tau(G) > \tilde{T}$  then
6     end
7 end
8 return  $\tau$ 

```

---

With Lemma 4.2 and Lemma 3.7, it can be shown that Algorithm 3 satisfies  $\epsilon$ -edge-DP. Furthermore, Algorithm 3 runs in polynomial time, as each  $Q_{\text{Del-Deg}}(G, \tau)$  can be solved in  $O(N)$  time. For the utility, we have:

LEMMA 4.3. Given any  $\epsilon$  and  $\beta$ , for any graph  $G$ , Algorithm 3 outputs a  $\tau$  such that with probability at least  $1 - \beta$ ,  $\tau \leq \deg(G)$ , and

$$|Q_{\text{Del-Deg}}(G, \tau)| \leq \frac{4}{\epsilon} \ln \deg(G) + \frac{8}{\epsilon} \ln \frac{2}{\beta}. \quad (3)$$

Due to space constraints, we defer this proof and some of the subsequent proofs to the appendix of the full version [27]. (3) further leads to an error of at most  $\Delta Q^{(\rho-1)}(G)$  with  $\rho = \tilde{O}(1/\varepsilon)$ , which implies Algorithm 3 is  $(\tilde{O}(1/\varepsilon), \tilde{O}(1/\varepsilon))$ -down neighborhood optimal. It can be further shown that setting  $\tau^* = \tau + 1$  satisfies our three desired properties. Since our main focus is on node-DP, we do not elaborate on the details here.

*Comparison with Laplace mechanism.* In the worst case, the output of Algorithm 3 has an additive error of  $O(\log(\deg(G)/\beta)/\varepsilon)$ . In comparison, the Laplace mechanism achieves a tighter additive error of  $O(\log(1/\beta)/\varepsilon)$ , which generally yields better performance on typical networks with power-law degree distributions. However, the key advantage of Algorithm 3 lies in achieving low rank error rather than low additive error. As we will discuss later, when integrated into our framework to upgrade any edge-DP solution to provide node-DP, the error is amplified by a factor of  $\tilde{O}(\deg(G))$ . As a result, the Laplace mechanism incurs an additive error of  $\tilde{O}(\deg(G)/\varepsilon)$ , which breaks the utility. In contrast, Algorithm 3 achieves a comparable  $\tilde{O}(\deg(G)/\varepsilon)$  rank error and still preserves strong utility guarantees.

**4.3.2 Node-DP Exponential-Time Maximum Degree Approximation.** So far, we have an effective solution for maximum degree estimation that satisfies three desired properties under edge-DP. However, directly extending this approach to node-DP is problematic. The main issue is that the property of bounded sensitivity for  $Q_{\text{Del-Deg}}(G, \tau)$  does not hold under node-DP. Since two neighboring graphs differing by a node can have a degree sum difference of up to  $2N$ , the global sensitivity for this query becomes  $N$  under node-DP. Therefore, a query function satisfying the three desired properties under node-DP is required.

Here, we introduce:

$$Q_{\text{Del-N}}(G, \tau) = -\min \{|N_G - N_{G^*}| : G^* \subseteq G, \deg(G^*) \leq \tau\}, \quad (4)$$

which is the opposite of the minimum number of nodes to remove from  $G$  to bound its degree with  $\tau$ . We observe that  $Q_{\text{Del-N}}(G, \tau)$  has three properties as below under node-DP, which is similar to  $Q_{\text{Del-Deg}}(G, \tau)$  under edge-DP.

LEMMA 4.4.  $Q_{\text{Del-N}}(G, \tau)$  satisfies the following 3 properties:

1. **Monotonicity:**  $Q_{\text{Del-N}}(G, \tau)$  is monotonic and becomes 0 when  $\tau \geq \deg(G)$ .
2. **Bounded Sensitivity:**  $Q_{\text{Del-N}}(G, \tau)$  has  $\text{GS}_Q = 1$  under node-DP.
3. **Sensitivity-Monotonicity:** For any two neighboring graphs  $G \sim_{\text{node}} G'$  and any  $\tau$ ,  $Q_{\text{Del-N}}(G, \tau) \geq Q_{\text{Del-N}}(G', \tau)$  if  $G \subseteq G'$ .

PROOF. The proofs for all properties are provided below.

1. **Monotonicity:** As  $\tau$  increases, fewer nodes need to be removed from  $G$  to ensure all node degrees are at most  $\tau$ . When  $\tau \geq \deg(G)$ , no removals are necessary, hence  $Q_{\text{Del-N}}(G, \tau) = 0$ . Since  $Q_{\text{Del-N}}(G, \tau)$  is defined as the opposite of the number of nodes to remove, its value increases with  $\tau$ .
2. **Bounded Sensitivity:** Let neighboring graphs  $G \sim_{\text{node}} G'$  and  $G \subseteq G'$ . Let  $S_G$  be a minimal node set to remove that bounds the degree of  $G$  by  $\tau$ . Then,  $S_G \cup \{v\}$  is a valid node removal set for  $G'$ , where  $v$  is the differing node. Thus,  $Q_{\text{Del-N}}(G', \tau) \geq Q_{\text{Del-N}}(G, \tau) + 1$ . Conversely, let  $S_{G'}$  be a minimal node removal set for  $G'$ . Since  $G \subseteq G'$ , removing  $S_{G'}$  from  $G$  also suffices to bound its degree by  $\tau$ . Therefore,  $Q_{\text{Del-N}}(G', \tau) \leq Q_{\text{Del-N}}(G, \tau)$ . To conclude, the global sensitivity of  $Q_{\text{Del-N}}(G, \tau)$  is bounded by 1.
3. **Sensitivity Monotonicity:** Follows directly from the argument in Property 2.  $\square$

Following the same steps of Algorithm 3, we can incorporate  $Q_{\text{Del-N}}(G, \tau)$  with SVT to obtain a maximum degree estimation  $\tau$  under node-DP. Similar to Lemma 4.3, it follows that  $\tau \leq \deg(G)$  and  $|Q_{\text{Del-N}}(G, \tau)| = \tilde{O}(1/\epsilon)$ . This implies that  $\tau$  upper-bounds the maximum degree of  $G$  after removing a minimal subset of  $\tilde{O}(1/\epsilon)$  nodes. However, such a  $\tau$  can not directly serve as  $\tau^*$ , as it fails to satisfy the second required property, which requires a bounded  $N_{\tau^*}(G)$ . Specifically,  $|Q_{\text{Del-N}}(G, \tau)| \leq N_{\tau}(G)$ , as removing all nodes with degree at least  $\tau$  yields a feasible solution to  $Q_{\text{Del-N}}(G, \tau)$ . Therefore, a bounded value of  $|Q_{\text{Del-N}}(G, \tau)|$  does not necessarily imply that  $N_{\tau}(G)$  is bounded.

To construct such a  $\tau^*$  satisfying our three desired properties under node-DP, we set

$$\tau^* = \tau + |Q_{\text{Del-N}}(G, \tau)| + \text{Lap}\left(\frac{c}{\epsilon}\right) + \text{polylog}, \quad (5)$$

where  $c$  is some constant. The high-level idea behind this construction is as follows. To satisfy the second desired property, we first increment  $\tau$  by  $|Q_{\text{Del-N}}(G, \tau)| + 1$  to get  $\tau'$ . As a result,

$$N_{\tau'}(G) \leq |Q_{\text{Del-N}}(G, \tau)|. \quad (6)$$

This follows from the definition of  $Q_{\text{Del-N}}(G, \tau)$ , which ensures that removing  $|Q_{\text{Del-N}}(G, \tau)|$  nodes from  $G$  can result in a graph  $G^*$  where all node degrees are at most  $\tau$ . Therefore, every node in  $G^*$  has a degree at most  $\tau + |Q_{\text{Del-N}}(G, \tau)|$  in  $G$ . Consequently, at most  $|Q_{\text{Del-N}}(G, \tau)|$  number of nodes in  $G$  have a degree exceeding this threshold, ensuring that  $N_{\tau'}(G) \leq |Q_{\text{Del-N}}(G, \tau)|$ .

However, using  $\tau'$  as  $\tau^*$  introduces an additional privacy concern:  $\tau'$  depends on  $G$ , as it includes the instance-specific term  $|Q_{\text{Del-N}}(G, \tau)|$ . Fortunately,  $|Q_{\text{Del-N}}(G, \tau)|$  has a global sensitivity of 1 by Lemma 4.4. Thus, adding a Laplace noise with scale  $2/\epsilon$  to  $\tau'$  ensures  $\epsilon/2$ -node-DP, and we denote the result as  $\tau''$ . Since  $\tau''$  is smaller than  $\tau'$  with probability  $1/2$ , it still violates the second property. To address this, we further increment  $\tau''$  by a logarithmic term and denote the result as  $\tau^*$ . This added logarithmic term ensures that  $\tau^*$  exceeds  $\tau'$  with high probability, thereby maintaining  $N_{\tau^*}(G)$  bounded by  $|Q_{\text{Del-N}}(G, \tau)|$ .

So far, we have already shown that  $\tau^*$  satisfies the first two desired properties. For the third property, we can use  $2\tau^*$  as an upper bound of  $g(G, \tau^*)$  since  $\tau^*$  already incorporates a high-probability bound on  $N_{\tau^*}(G)$  through the term  $|Q_{\text{Del-N}}(G, \tau)|$ .

---

**Algorithm 4:** Node-DP Maximum Degree Approximation
 

---

**Input:** Graph  $G$ , privacy budgets  $\epsilon$  and  $\delta$ , parameter  $\beta$

**Output:**  $\tau^*$

```

1  $T \leftarrow -8 \ln(4/\beta)/\epsilon$ 
2  $\tilde{T} \leftarrow T + \text{Lap}(4/\epsilon)$ 
3 for  $\tau = 1, 2, 4, 8, \dots$  do
4    $\tilde{Q}_{\tau}(G) \leftarrow Q_{\text{Del-N}}(G, \tau) + \text{Lap}(4/\epsilon)$ 
5   if  $\tilde{Q}_{\tau}(G) > \tilde{T}$  then
6     end
7 end
8  $\tau^* \leftarrow \tau + |Q_{\text{Del-N}}(G, \tau)| + \text{Lap}(\frac{2}{\epsilon}) + \frac{2}{\epsilon} \ln \max(\frac{1}{\delta}, \frac{2}{\beta}) + 1$ 
9 return  $\tau^*$ 

```

---

Finally, our node-DP mechanism for maximum degree approximation works as follows. We first invoke a similar algorithm as Algorithm 3, but replaces  $Q_{\text{Del-Deg}}(G, \tau)$  with  $Q_{\text{Del-N}}(G, \tau)$ . Besides, we evaluate  $\tau$  on a logarithmic scale of  $\tau = 1, 2, 4, 8, \dots$  to reduce the number of iterations, since

the computation of  $Q_{\text{Del-N}}(G, \tau)$  is costly. This approach is acceptable because, according to the first property of  $\tau^*$ , we only require an approximation rather than an accurate estimation of the maximum degree. After obtaining  $\tau$  from the SVT, we construct  $\tau^*$  following (5) with carefully designed parameters. The detailed algorithm is shown as Algorithm 4.

For privacy, Algorithm 4 satisfies  $\epsilon$ -node-DP. Specifically,  $\tau$  obtained in the SVT preserves  $\epsilon/2$ -node-DP, while the remaining  $\epsilon/2$  privacy budget is allocated to privatize  $\tau^*$ . The parameter  $\delta$  in Line 8 is used for utility bounds and does not affect the privacy guarantees. For utility, we allocate an error failure probability of  $\beta/2$  to the computation of  $\tau$  and  $\beta/2$  to that of  $\tau^*$ . Line 8 further ensures that  $|Q_{\text{Del-N}}(G, \tau)|$  upper-bounds  $N_{\tau^*}(G)$  with failure probability at most  $\delta$ . The following utility guarantees hold:

LEMMA 4.5. *Given any  $\epsilon$ ,  $\beta$ , and  $\delta$ , for any graph  $G$ , Algorithm 4 returns a  $\tau^*$  such that with probability at least  $1 - \beta$ ,*

$$\tau^* \leq 2\text{deg}(G) + \frac{8}{\epsilon} \ln \log(4\text{deg}(G)) + \frac{16}{\epsilon} \ln \frac{4}{\beta} + \frac{4}{\epsilon} \ln \max\left(\frac{1}{\delta}, \frac{2}{\beta}\right) + 1,$$

and

$$N_{\tau^*}(G) \leq \frac{8}{\epsilon} \ln \log(4\text{deg}(G)) + \frac{16}{\epsilon} \ln \frac{4}{\beta}.$$

Furthermore, with probability at least  $1 - \delta$ ,

$$\tau^* + N_{\tau^*}(G) \leq 2\tau^*.$$

So far, we have obtained a node-DP maximum degree approximation  $\tau^*$  satisfying all three desired properties. However, the computation of  $\tau^*$  incurs significant computational costs. Specifically, computing  $Q_{\text{Del-N}}(G, \tau)$  requires examining all possible combinations of node deletions, resulting in exponential runtime in the worst case.

**4.3.3 Node-DP Polynomial-Time Maximum Degree Approximation.** In this section, we show how to reduce the runtime of our node-DP maximum degree approximation mechanism. Recall that the bottleneck lies in computing  $Q_{\text{Del-N}}(G, \tau)$ , which requires exponential time. Our key idea is to approximate its value by leveraging Linear Programming (LP) and integrate the result with our proposed node-DP algorithm. We will show that the enhanced algorithm still outputs a  $\tau^*$  that complies with the three desired properties.

We begin by approximating  $Q_{\text{Del-N}}(G, \tau)$  with an LP formulation, denoted as  $Q_{\text{LP-Del-N}}(G, \tau)$ . In this formulation, we allow the fractional existence of nodes and edges in  $G$ . Specifically, for each node  $v \in V$ , we assign a value  $x_v \in [0, 1]$  to indicate whether to delete the node  $v$  from  $G$ , i.e., 0 represents complete retention and 1 represents complete removal. In contrast, for each edge  $e \in E$ , we assign a value  $y_e$  from  $[0, 1]$  to indicate whether to maintain  $e$ , i.e., 0 represents removal and 1 represents retention. The LP  $Q_{\text{LP-Del-N}}(G, \tau)$  maximizes the opposite of node removal count while ensuring constraints on node degrees and node-edge relationships, indicated by (7) and (8) respectively. The LP formulation of  $Q_{\text{LP-Del-N}}(G, \tau)$  is given by:

$$\text{maximize } Q_{\text{LP-Del-N}}(G, \tau) = - \sum_{v \in V} x_v$$

$$\text{s.t. } y_e \geq 1 - x_{v'} - x_{v''}, \quad e = (v', v'') \in G, \quad (7)$$

$$\sum_{e' \in E(v)} y_{e'} \leq \tau, \quad v \in V, \quad (8)$$

$$y_e \in [0, 1], \quad e \in E,$$

$$x_v \in [0, 1], \quad v \in V.$$

It is trivial to see  $Q_{LP-Del-N}(G, \tau)$  is a relaxed version of  $Q_{Del-N}(G, \tau)$ :  $Q_{Del-N}(G, \tau)$  can be formulated as an integer linear programming where each variable is from a discrete domain, i.e.,  $x_v \in \{1, 0\}$  and  $y_e \in \{1, 0\}$ , where the values have the same interpretations as in  $Q_{LP-Del-N}(G, \tau)$ . Different from  $Q_{Del-N}(G, \tau)$  that requires exponential runtime,  $Q_{LP-Del-N}(G, \tau)$  can be solved in polynomial time. Moreover, it retains the same properties as  $Q_{Del-N}(G, \tau)$  under node-DP, as outlined below.

LEMMA 4.6.  $Q_{LP-Del-N}(G, \tau)$  satisfies the following 3 properties under node-DP:

1. **Monotonicity:**  $Q_{LP-Del-N}(G, \tau)$  is monotonic and becomes 0 when  $\tau \geq \deg(G)$ .
2. **Bounded Sensitivity:**  $Q_{LP-Del-N}(G, \tau)$  has  $GS_Q = 1$ .
3. **Sensitivity-Monotonicity:**  $Q_{LP-Del-N}(G, \tau)$  is sensitivity-monotonic.

PROOF. Assume that  $\{x_v\}$  and  $\{y_e\}$  is the optimal solution for  $Q_{LP-Del-N}(G, \tau)$ . The proofs of all properties are given below.

1. **Monotonicity:** As  $\tau$  increases, the values of  $y_e$  can increase, leading to lower values of  $x_v$ . When  $\tau \geq \deg(G)$ , all edges can be retained, resulting in  $x_v = 0$  for all  $v \in V$ . Since  $Q_{LP-Del-N}(G, \tau)$  is the opposite of the sum of  $x_v$ , it increases with  $\tau$  and becomes 0 when  $\tau \geq \deg(G)$ .
2. **Bounded Sensitivity:** Let neighboring graphs  $G \sim_u G'$  such that  $G \subseteq G'$  and  $G' = (V', E')$ . A feasible solution for  $G'$  can be constructed by keeping the values of  $\{x_v\}$  and  $\{y_e\}$  from  $G$ , and set  $x_u = 1$  and  $y'_e = 0$  for  $e \in E'(u)$ . Thus,  $Q_{LP-Del-N}(G, \tau) - 1 \leq Q_{LP-Del-N}(G', \tau)$ . Conversely, a feasible solution for  $G$  can be constructed by applying the assignments from the optimal solution of  $G'$  to the corresponding nodes and edges in  $G$ . Therefore,  $Q_{LP-Del-N}(G', \tau) \leq Q_{LP-Del-N}(G, \tau)$ . To conclude,  $Q_{LP-Del-N}(G, \tau)$  has a global sensitivity of 1.
3. **Sensitivity-Monotonicity:** Follows directly from the argument in Property 2.  $\square$

These three properties ensure that by replacing  $Q_{Del-N}(G, \tau)$  with  $Q_{LP-Del-N}(G, \tau)$ , Algorithm 4 still preserves  $\varepsilon$ -node-DP. However, the utility guarantee for the output  $\tau^*$  no longer holds. Since  $Q_{LP-Del-N}(G, \tau)$  is a relaxation of  $Q_{Del-N}(G, \tau)$ , its value is greater than  $Q_{Del-N}(G, \tau)$ . As a result, when using  $Q_{LP-Del-N}(G, \tau)$ , the SVT may stop earlier with a smaller  $\tau$ . While such an  $\tau$  ensures a bounded  $|Q_{LP-Del-N}(G, \tau)|$ , it does not necessarily imply a bounded  $|Q_{Del-N}(G, \tau)|$ , which is essential to satisfy the second property.

To address this issue, we establish a relationship between  $Q_{LP-Del-N}(G, \tau)$  and  $Q_{Del-N}(G, \tau)$ . The high-level idea is as follows. Let  $\{x_v\}$  and  $\{y_e\}$  denote the optimal solution to the LP  $Q_{LP-Del-N}(G, \tau)$ . Recall that  $x_v$  represents the fractional removal of node  $v$  from  $G$ , and  $y_e$  represents the fractional retention of edge  $e$ . Now we construct a graph  $G'$  by removing all nodes  $v$  with  $x_v > 1/3$  from  $G$ . On one hand, since only nodes with  $x_v > 1/3$  are removed, the total number of deleted nodes is at most 3 times  $\sum_{v \in V} x_v$ , which is equivalent to  $3 \cdot |Q_{LP-Del-N}(G, \tau)|$ . On the other hand, for every edge in  $G'$ , both endpoints  $v$  and  $v'$  satisfy  $x_v, x_{v'} \leq 1/3$ , which implies  $y_e \geq 1/3$ . Therefore, it can be inferred that the maximum degree of  $G'$  is bounded by  $3\tau$ .

Thus, we obtain the relationship between  $Q_{LP-Del-N}(G, \tau)$  and  $Q_{Del-N}(G, \tau)$ :

LEMMA 4.7. For any  $G$  and any  $\tau$ ,

$$|Q_{Del-N}(G, 3\tau)| \leq 3 \cdot |Q_{LP-Del-N}(G, \tau)|. \quad (9)$$

Therefore, by feeding  $Q_{LP-Del-N}(G, \tau)$  into the SVT, we obtain a  $\tau$  such that  $\tau \leq 2 \deg(G)$  and  $|Q_{LP-Del-N}(G, \tau)|$  is bounded by  $\tilde{O}(1/\varepsilon)$ . Following a similar analysis to Section 4.3.2, we modify Line 8 of Algorithm 4 to compute  $\tau^*$  as:

$$\tau^* \leftarrow 3\tau + 3 \cdot |Q_{LP-Del-N}(G, \tau)| + \text{Lap}\left(\frac{6}{\varepsilon}\right) + \frac{6}{\varepsilon} \ln \max\left(\frac{1}{\delta}, \frac{2}{\beta}\right) + 1, \quad (10)$$

which yields an  $\tau^*$  that satisfies the three desired properties:

LEMMA 4.8. *Given any  $\epsilon$ ,  $\beta$ , and  $\delta$ , for any graph  $G$ , the polynomial-time node-DP maximum degree approximation algorithm returns a  $\tau^*$  such that with probability at least  $1 - \beta$ ,*

$$\tau^* \leq 6\deg(G) + \frac{24}{\epsilon} \ln \log(4\deg(G)) + \frac{48}{\epsilon} \ln \frac{4}{\beta} + \frac{12}{\epsilon} \ln \max\left(\frac{1}{\delta}, \frac{2}{\beta}\right) + 1,$$

and

$$N_{\tau^*}(G) \leq \frac{24}{\epsilon} \ln \log(4\deg(G)) + \frac{48}{\epsilon} \ln \frac{4}{\beta}.$$

Furthermore, with probability at least  $1 - \delta$ ,

$$\tau^* + N_{\tau^*}(G) \leq 2\tau^*.$$

The revised algorithm maintains the same  $\epsilon$ -node-DP guarantee as Algorithm 4. For utility, the allocation of error failure probabilities remains unchanged, and the upper bounds on  $\tau^*$  and  $N_{\tau^*}(G)$  increase by roughly a factor of 3 under the same parameters. In terms of runtime, since each invocation of  $Q_{\text{LP-DeI-N}}(G, \tau)$  runs in polynomial time and there are  $O(\log \deg(G))$  iterations, the overall algorithm also runs in polynomial time.

#### 4.4 Combined Solution

By combining the distance-preserving clipping mechanism from Section 4.2 with the node-DP polynomial-time maximum degree approximation algorithm from Section 4.3, we develop the N2E framework that reduces an arbitrary node-DP graph analytical task to an edge-DP one as follows. Given a graph analytical task  $Q$ , an edge-DP mechanism  $M_Q^{\text{edge}}$ , an input graph  $G$ , privacy budgets  $\epsilon$  and  $\delta$ , and error failure probability  $\beta$ , the framework proceeds in three steps:

- Compute  $\tau^*$  using the node-DP polynomial-time maximum degree approximation on input graph  $G$ , with privacy budgets  $2\epsilon/3$  and  $\delta/2$ , and error failure probability  $2\beta/3$ .
- Clip graph  $G$  using the distance-preserving clipping mechanism described in Algorithm 2, with the degree upper bound set to  $\tau^*$ . Denote the resulting graph as  $\bar{G}$ .
- Apply the edge-DP mechanism  $M_Q^{\text{edge}}$  to the clipped graph  $\bar{G}$  with privacy budgets  $\epsilon/6\tau^*$  and  $\delta/4\tau^*$ , and error failure probability  $\beta/3$ .

The framework outputs the result of the edge-DP mechanism, denoted as  $M_Q^{\text{node}}(G, \epsilon, \delta, \beta)$ . For privacy,

THEOREM 4.9. *The N2E framework satisfies  $(\epsilon, \delta)$ -node-DP.*

For utility, we have:

THEOREM 4.10. *Given any  $\epsilon$ ,  $\beta$ , and  $\delta$ , for any graph  $G$ , task  $Q$ , and edge-DP mechanism  $M_Q^{\text{edge}}$ , with probability at least  $1 - \beta$ , the N2E framework outputs a node-DP result with error*

$$|Q(G) - Q(\bar{G})| + \text{Err}_Q^{\text{edge}}(\bar{G}, \epsilon/6\tau^*, \delta/4\tau^*, \beta/3),$$

where  $\bar{G}$  is obtained by clipping edges of  $O(\log(\log(\deg(G))/\beta)/\epsilon)$  nodes from  $G$ , and  $\text{Err}_Q^{\text{edge}}(\bar{G}, \epsilon/6\tau^*, \delta/4\tau^*, \beta/3)$  denotes the error of the edge-DP mechanism on task  $Q$  given clipped graph  $\bar{G}$ , privacy budgets  $\epsilon/6\tau^*$  and  $\delta/4\tau^*$ , and error failure probability  $\beta/3$ .

Specifically, the first component of the error,  $|Q(G) - Q(\bar{G})|$ , represents the clipping bias introduced by the difference between the query result on the original graph  $G$  and the clipped graph  $\bar{G}$ . For example, when  $Q$  is the edge count, this bias is exactly the number of clipped edges. By

Theorem 4.10, this component is at most  $\Delta Q^{(\rho-1)}$ , where  $\rho = \tilde{O}(1/\varepsilon)$  with constant probability. This implies that only a negligible portion of the graph is affected by clipping. Moreover, according to Lemma 3.9, this part of the error is optimal. It follows that any structural impact on the graph (e.g., connectivity) introduced by the distance-preserving clipping mechanism is both unavoidable and minimal within the N2E framework.

The second component of the error,  $\text{Err}_Q^{\text{edge}}(\bar{G}, \varepsilon/6\tau^*, \delta/4\tau^*, \beta/3)$ , captures the error introduced by the edge-DP mechanism. In practice, this error approximates  $\text{Err}_Q^{\text{edge}}(G, \varepsilon/6\tau^*, \delta/4\tau^*, \beta/3)$ , since many edge-DP mechanisms for common graph analytical tasks exhibit similar or smaller error on the clipped graph  $\bar{G}$  compared to  $G$ . Moreover, by Lemma 4.8 that gives the upper bound of  $\tau^*$ , this error amplifies the original edge-DP error of  $\text{Err}_Q^{\text{edge}}(G, \varepsilon, \delta, \beta)$  by only a factor of  $\tilde{O}(\text{deg}(G))$ , which achieves our desired utility.

## 5 Application

In this section, we apply our N2E framework to three common graph analytical tasks: edge counting, maximum degree estimation, and degree distribution under node-DP. We analyze the error for each task individually. For convenience, all results stated in this section hold with constant probability. Technical details are deferred to the appendix of the full version [27].

*Edge counting.* We use the Laplace mechanism as the edge-DP mechanism within the N2E framework. For the clipping bias, the total number of edges clipped to obtain  $\bar{G}$  is bounded by  $\tilde{O}(\text{deg}(G)/\varepsilon)$ . The error from the edge-DP mechanism is bounded by  $\tilde{O}(\text{deg}(G)/\varepsilon + 1/\varepsilon^2)$ . As a result, the total error matches our target of upgrading the edge-DP error of  $O(1/\varepsilon)$  to node-DP by only a factor of  $\tilde{O}(\text{deg}(G))$  with a small additive term of  $\tilde{O}(1/\varepsilon)$ .

*Maximum degree estimation.* Applying the Laplace mechanism within the N2E framework incurs an additive error of  $\tilde{O}(\text{deg}(G)/\varepsilon)$ , which has limited utility. To address this, we adopt Algorithm 3 as the edge-DP mechanism and use edge distance for evaluation. This algorithm incurs an edge distance of  $\tilde{O}(1/\varepsilon)$  under edge-DP. Within the N2E framework, the edge distance from clipping bias is bounded by  $\tilde{O}(\text{deg}(G)/\varepsilon)$ . The application of Algorithm 3 adds an edge distance of  $\tilde{O}(\text{deg}(G)/\varepsilon + 1/\varepsilon^2)$ . Therefore, the total edge distance is asymptotically dominated by the edge-DP mechanism, achieving our target of upgrading the edge-DP error by  $\tilde{O}(\text{deg}(G))$  with an additive term of  $\tilde{O}(1/\varepsilon)$ .

*Degree distribution.* A straightforward solution for this task is to integrate the Laplace mechanism into N2E, which introduces a clipping bias of  $\tilde{O}(\text{deg}(G)/\varepsilon)$  and an error of  $\tilde{O}(\text{deg}^{1.5}(G)/\varepsilon + \text{deg}^{0.5}(G)/\varepsilon^2)$  introduced by the Laplace noise. This approach satisfies  $(\varepsilon, \delta)$ -node-DP and achieves our goal of amplifying the edge-DP error of  $O(\text{deg}(G)^{0.5}/\varepsilon)$  by a factor of  $\tilde{O}(\text{deg}(G))$ , with an additive term of  $\tilde{O}(1/\varepsilon)$ . Additionally, we observe that directly combining our node-DP degree approximation with their clipping mechanism yields a pure  $\varepsilon$ -node-DP solution with asymptotically the same utility guarantees. We adopt this  $\varepsilon$ -node-DP solution in practice.

*Limitation of the N2E framework.* N2E is applicable to any graph analytical tasks with edge-DP solutions, either being  $\varepsilon$ -edge-DP or being  $(\varepsilon, \delta)$ -edge-DP. However, N2E is not suitable for tasks that lack corresponding edge-DP solutions or for simple node-DP tasks with trivial solutions, such as node counting. Moreover, the performance of N2E heavily depends on the underlying edge-DP mechanism. If the edge-DP mechanism has poor utility or high computational cost, these limitations will carry over to N2E.

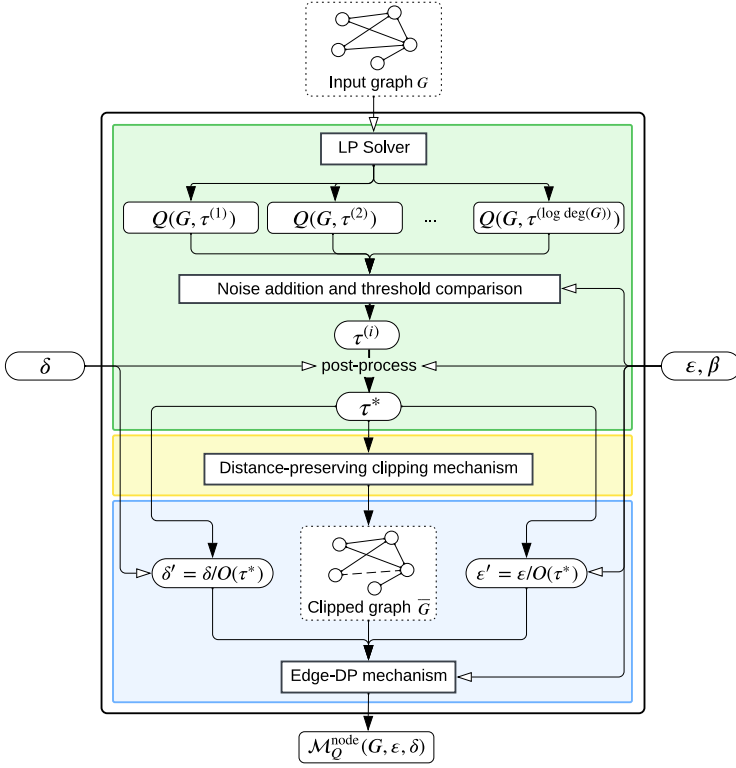


Fig. 3. Overview of the N2E framework.

## 6 Implementation

We implemented our N2E framework on top of CPLEX [37]. An overview is shown in Figure 3, where the three main steps are highlighted in different colors. Specifically, the input graph  $G$  is utilized in both the LP construction and the clipping mechanism. Regarding input parameters, the framework includes three procedures that consume  $\epsilon$  and  $\beta$ , two of which are involved in the maximum degree approximation. The parameter  $\delta$  is used in the post-processing step to compute  $\tau^*$  and in the edge-DP mechanism.

*Runtime optimization.* In practice, the first step of the N2E framework, maximum degree approximation, accounts for the majority of the runtime in our experiments. Although this procedure achieves polynomial-time complexity with LP, solving the LP remains computationally expensive. This is because the number of variables in the LP, which is equal to the sum of nodes and edges, can be substantial for real-world datasets. This issue is further amplified as one LP is to be solved in each SVT iteration, and the SVT typically takes multiple iterations to identify the desired  $\tau$ .

To reduce sequential runtime, we first parallelized the LPs in SVT iterations within the maximum degree approximation, as illustrated in Figure 3. The LP result is 0 for  $\tau \geq \deg(G)$ . To avoid unnecessary computation, we implement early termination. If any process returns a qualified  $\tau$  that exceeds the SVT threshold and all smaller candidates are unqualified, the process pool is terminated early and returns the corresponding  $\tau$ .

To further reduce the runtime for each candidate  $\tau$ , we leverage the dual simplex method provided by the LP solver CPLEX, which solves LPs by moving towards the optimal solution from above.

Since we are only interested in  $\tau$  candidates whose LP result exceeds the SVT threshold, we stop the computation early if the current LP upper bound of  $\tau$  already falls below the threshold. This strategy significantly reduces runtime, especially in cases where resource constraints prevent parallelization.

## 7 Experiment

We conduct experiments on the three graph analytical tasks under node-DP discussed in Section 5, along with an additional task of two-line path counting, and compare the performance of our N2E framework with state-of-the-art node-DP algorithms. For edge counting and two-line path counting, we compare against R2T [11] and OPT<sup>2</sup> [12]. For maximum degree estimation, so far, there is no existing solution under node-DP. As we have shown before, the naive solution of directly adding Laplace noise to the query result does not have utility. Therefore, we compare it with a baseline that returns the node count with Laplace noise, referred to as N-Count. For degree distribution, we compare our results with DLL-Histogram [7].

### 7.1 Setup

Dataset	Nodes	Edges	Max Degree	Connectivity <sup>5</sup>
EUEmail	265,009	364,481	7,636	0.21
Deezer	143,884	846,915	420	0.17
Amazon1	262,111	899,792	420	0.08
Amazon2	334,863	925,872	549	0.06
DBLP	317,080	1,049,866	343	0.18
Amazon0601	403,394	2,443,408	2,752	0.10
Web-Google	875,713	4,322,051	6,332	0.01

Table 2. Summary of graph datasets used in the experiment.

*Datasets.* Our experiments use seven real-world graph datasets: EUEmail, Deezer, Amazon1, Amazon2, DBLP, Amazon0601, and Web-Google, all obtained from SNAP [34]. These datasets span various domains, including e-commerce, academic collaboration, email communication, and web networks. Details of all datasets are provided in Table 2. All graphs are preprocessed to be undirected for consistency. For the Deezer and Amazon1 datasets, we follow the same preprocessing procedure as in [11] to match their experimental results.

*Queries.* We use four node-DP graph analysis queries: edge counting ( $Q_{EC}$ ), two-line path counting ( $Q_{TP}$ ), maximum degree estimation ( $Q_{MD}$ ), and degree distribution ( $Q_{DD}$ ).

For  $Q_{EC}$  and  $Q_{TP}$ , we use additive error as the evaluation metric. For  $Q_{MD}$ , we use the rank error<sup>6</sup> for evaluation. For  $Q_{DD}$ , we construct the degree histogram on a logarithmic scale of degrees and report the L1 error for evaluation.

*Experimental parameters.* All experiments are conducted on a Linux server equipped with dual Intel Xeon 22-core 2.20 GHz CPUs and 1 TB of memory. Each experiment is repeated for 10 rounds, and we report the average result after excluding the top 2 and bottom 2 runs to mitigate the randomness of the DP mechanism. All algorithms are limited to a 6-hour runtime; any algorithm exceeding this limit is terminated.

<sup>5</sup>Connectivity is computed as the ratio of intra-cluster to inter-cluster edges based on Louvain clustering [3].

<sup>6</sup>For results exceeding the true value, we compute the rank error as the absolute error.

We set  $\beta = 0.1$  and  $\delta = 2^{-30}$  in all experiments. By default, we set  $\epsilon = 0.8$  for  $Q_{EC}$ ,  $Q_{TP}$  and  $Q_{MD}$ . Since the degree histogram involves multiple queries, we relax the default  $\epsilon$  to 3.2 for  $Q_{DD}$ . The division of  $\epsilon$ ,  $\delta$ , and  $\beta$  across the procedures in Figure 3 is flexible and does not asymptotically affect the theoretical guarantees. Based on empirical results, we allocate  $(\epsilon, \delta, \beta)$  as  $(20\%, 0\%, 20\%)$ ,  $(20\%, 100\%, 0.01\%)$ , and  $(60\%, 0\%, 79.99\%)$  to the first, second, and third procedures, respectively. Both R2T and DLL-Histogram rely on a degree upper bound, which is set to the smallest power of 2 exceeding the number of nodes for each dataset. For example, the bound is set to  $2^{18} = 262,144$  for Deezer.

## 7.2 Experiment Results

Dataset		EUEmail	Deezer	Amazon1	Amazon2	DBLP	Amazon0601	Web-Google	
Query result		$3.6 \times 10^5$	$8.5 \times 10^5$	$9.0 \times 10^5$	$9.3 \times 10^5$	$1.0 \times 10^6$	$2.4 \times 10^6$	$4.3 \times 10^6$	
$Q_{EC}$	N2E	Rel. err (%)	6.20	0.18	0.12	0.19	0.23	0.24	0.76
		Err. std (%)	0.77	0.07	0.07	0.17	0.14	0.09	0.20
		Time (s)	279.85	66.39	65.41	81.66	71.65	267.66	1363.91
	OPT <sup>2</sup>	Rel. err (%)	10.51	0.47	0.35	0.23	0.45	0.40	1.73
		Err. std (%)	1.37	0.05	0.02	0.02	0.02	0.05	0.94
		Time (s)	76.46	170.87	180.86	141.66	181.56	521.03	1370.32
R2T	Rel. err (%)	23.32	0.98	0.87	0.75	0.71	1.72	5.68	
	Err. std (%)	1.02	0.16	0.13	0.04	0.03	0.18	0.38	
	Time (s)	3.32	10.43	10.70	9.94	11.78	26.38	55.86	
Query result		$2.0 \times 10^8$	$2.2 \times 10^7$	$9.1 \times 10^6$	$9.8 \times 10^6$	$2.2 \times 10^7$	$7.2 \times 10^7$	$7.3 \times 10^8$	
$Q_{TP}$	N2E	Rel. err (%)	43.13	3.16	11.34	9.52	7.43	10.71	9.33
		Err. std (%)	2.84	1.54	3.51	2.58	3.09	4.16	3.48
		Time (s)	269.00	66.56	65.72	79.59	68.88	270.52	1358.04
	OPT <sup>2</sup>	Rel. err (%)	Over time limit			4.91	Over time limit		
		Err. std (%)	Over time limit			0.20	Over time limit		
		Time (s)	Over time limit			12498.95	Over time limit		
R2T	Rel. err (%)	Out of memory	4.94	11.97	8.92	6.22	24.35	Out of memory	
	Err. std (%)	Out of memory	1.04	0.31	0.19	0.82	1.88	Out of memory	
	Time (s)	Out of memory	280.20	113.80	122.51	284.82	953.21	Out of memory	
Query result		7,636	420	420	549	343	2,752	6,332	
$Q_{MD}$	N2E	Rel. err (%)	0.37	3.10	1.87	1.12	1.10	0.65	0.18
		Err. std (%)	0.12	0.40	0.24	0.33	0.09	0.08	0.09
		Time (s)	269.22	66.89	65.99	80.09	70.22	271.99	1359.70
	N-Count	Rel. err (%)	3370.52	34158.06	62307.38	60894.99	92343.05	14558.22	13729.96
		Err. std (%)	0.01	0.10	0.15	0.10	0.24	0.04	0.00
		Time (s)	-	-	-	-	-	-	-
Query result		$2.6 \times 10^5$	$1.4 \times 10^5$	$2.6 \times 10^5$	$3.4 \times 10^5$	$3.2 \times 10^5$	$4.0 \times 10^5$	$8.8 \times 10^5$	
$Q_{DD}$	N2E	Rel. err (%)	8.17	3.38	1.21	1.05	1.94	3.75	8.76
		Err. std (%)	1.20	0.93	0.11	0.32	0.29	0.36	1.79
		Time (s)	30.10	63.67	64.39	65.95	72.68	153.51	272.08
	DLL-Histogram	Rel. err (%)	23.96	86.65	90.17	83.82	33.47	17.24	53.01
		Err. std (%)	1.72	28.45	72.77	79.19	20.02	13.23	44.56
		Time (s)	12.18	8.27	15.27	20.20	21.41	33.34	66.74

Table 3. Mean relative error (%), relative error std (%), and runtime (s) for all methods across four query types on seven real-world datasets. The top row in each query block gives the true query result.

*Utility and efficiency.* The experimental results, including relative errors with standard deviations and runtimes across all queries and datasets, are summarized in Table 3. N2E consistently achieves

higher utility in the majority of query-dataset combinations, considering both the mean and standard deviation of the relative errors.

For  $Q_{EC}$ , N2E achieves a relative error below 10% for all datasets, and under 1% for six out of seven datasets. Compared to the two baseline methods, N2E improves upon the error of  $OPT^2$  by up to 2.5 $\times$ , and R2T by up to 7.5 $\times$ . In terms of runtime, N2E is faster than  $OPT^2$  on six out of seven datasets and performs particularly well on graphs with smaller maximum degrees.

For  $Q_{TP}$ , N2E is the only mechanism that completes on all datasets within the given time and memory constraints.  $OPT^2$  fails to run on most datasets due to exceeding the time limit, while R2T runs out of memory on two datasets. Among the cases where baselines complete, N2E achieves comparable errors while reducing runtime by over 100 $\times$ . Notably, the error of N2E on the EUEmail dataset is relatively large due to clipping bias, which truncates the high contributions from nodes with very large degrees.

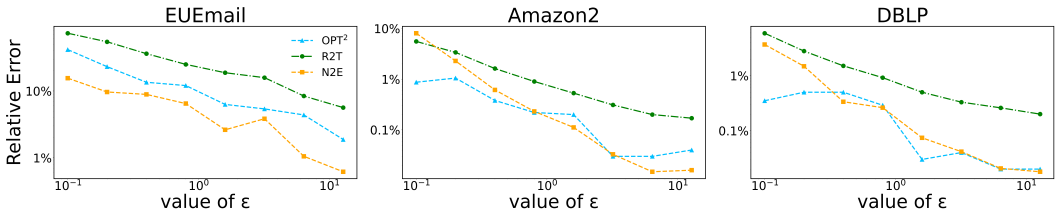
For  $Q_{MD}$ , N2E maintains a relative rank error below 4% across all datasets, demonstrating its robustness to variations in the maximum degree. In comparison, N-Count provides no utility, exhibiting relative errors exceeding 3,000% on all datasets. The runtime of N2E for  $Q_{MD}$  is nearly identical to that for  $Q_{EC}$  and  $Q_{TP}$  across all datasets. This is expected, as all three queries follow the same steps of the N2E framework and involve an edge-DP mechanism with similar computational overhead.

For  $Q_{DD}$ , N2E outperforms DLL-Histogram with up to 80 $\times$  reduction in relative error across all datasets, albeit with up to 8 $\times$  higher runtime. Across datasets, N2E demonstrates better utility in graphs with lower maximum degrees, as higher degrees lead to larger noise scales added to the histogram. The runtime of N2E on  $Q_{DD}$  is lower than that on  $Q_{EC}$ ,  $Q_{TP}$ , and  $Q_{MD}$ . This arises from the distinct clipping mechanism used. In the latter three queries, the distance-preserving clipping mechanism as defined in Algorithm 2 evaluates edge rankings of both endpoints, introducing significant computational overhead for graphs with many edges. In contrast, the clipping mechanism in [7] used for  $Q_{DD}$  considers only the degrees of the endpoints, resulting in reduced computational cost.

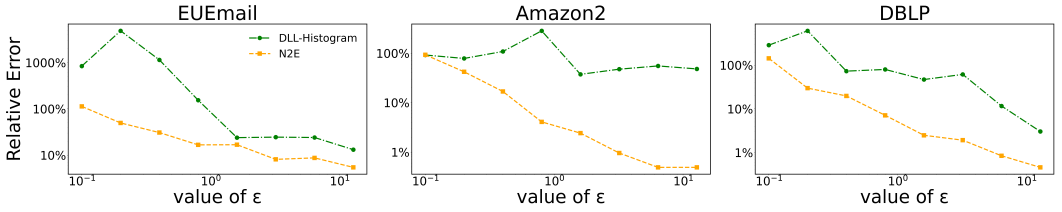
*Consistency with theoretical guarantees.* As shown in Section 5, the absolute errors of  $Q_{EC}$  and  $Q_{DD}$  depend heavily on the maximum degree of the input graph, with higher maximum degrees leading to larger errors. Theoretical results of  $Q_{TP}$  also exhibit this dependency. However, since we use relative error for evaluation, the results are influenced by factors such as graph size and skewness, i.e., the uniformity of the degree distribution. In general, datasets with larger graph sizes and less skewed degree distributions tend to exhibit lower relative errors. For example, DBLP generally yields lower relative errors on the three queries due to its medium graph size and more balanced degree distribution. For  $Q_{MD}$ , the theoretical guarantee is provided in terms of edge distance. Since we evaluate using relative rank error, additional factors beyond graph size and skewness also influence the results. As a result, the performance trend remains consistent across the first three queries, but does not extend to  $Q_{MD}$ .

Cluster-level connectivity appears to have minimal impact on query performance. For example, in  $Q_{TP}$ , the Web-Google dataset achieves a relative error of 9.33%, nearly identical to the 10.71% error observed on Amazon0601, despite a 10 $\times$  difference in connectivity.

*Privacy parameter  $\epsilon$ .* To compare with the performance of N2E and baseline methods under different privacy guarantees, we vary the privacy budget  $\epsilon$  on a logarithmic scale from 0.1 to 12.8. Experiments are conducted on the EUEmail, Amazon2, and DBLP datasets to assess performance across two graph analytical tasks  $Q_{EC}$  and  $Q_{DD}$ . The results are shown in Figure 4. As shown, all methods demonstrate improved utility as  $\epsilon$  increases. For  $Q_{EC}$ , N2E outperforms R2T and achieves utility comparable to  $OPT^2$ , which is provably optimal. For  $Q_{DD}$ , N2E generally outperforms the



(a) Relative error (%) for  $Q_{EC}$  on the EUEmail, Amazon2, and DBLP datasets across varying  $\epsilon$  values.



(b) Relative error (%) for  $Q_{DD}$  on the EUEmail, Amazon2, and DBLP datasets across varying  $\epsilon$  values.

Fig. 4. Relative error (%) of N2E and baseline methods for  $Q_{EC}$  and  $Q_{DD}$  on three datasets under varying privacy budgets  $\epsilon$ .

baseline method across the entire range of  $\epsilon$ . While DLL-Histogram exhibits high variability with different  $\epsilon$  values due to randomness, N2E yields more stable results.

Furthermore, N2E generally achieves higher utility on  $Q_{EC}$  than on  $Q_{DD}$  under small  $\epsilon$  values, due to the higher base value of the edge counting result. In contrast, the utility for  $Q_{DD}$  is more affected under small  $\epsilon$  because it involves answering multiple queries while having a smaller base value.

Dataset	EUEmail	DBLP	Amazon0601
N2E (Rel. error (%))	6.20	0.23	0.24
$\tau^* = 1$	95.44	99.83	99.61
$\tau^* = 4$	93.14	85.08	92.30
$\tau^* = 16$	88.40	35.98	30.74
$\tau^* = 64$	75.26	4.38	6.22
$\tau^* = 256$	44.49	0.036	0.95
$\tau^* = 1024$	11.93	0.22	0.18
$\tau^* = 4096$	3.85	0.84	0.43
$\tau^* = 16384$	14.42	1.86	1.16

Table 4. Relative error (%) of  $Q_{EC}$  on three datasets using N2E and different clipping thresholds  $\tau^*$ .

*Optimality of  $\tau^*$ .* N2E relies on the maximum degree approximation  $\tau^*$  as the degree upper bound to clip the input graph. To assess the performance of  $\tau^*$  within N2E, we manually set different values of  $\tau^*$  on a logarithmic scale as the clipping threshold, and evaluate the results on three datasets: EUEmail, DBLP, and Amazon0601. The results for  $Q_{EC}$  and  $Q_{DD}$  are presented in Table 4 and Table 5, respectively.

Compared to the optimal value of  $\tau^*$ , N2E incurs only a constant-factor utility loss in error. This is primarily due to the allocation of part of the privacy budget for computing  $\tau^*$ . Additionally,

Dataset	EUEmail	DBLP	Amazon0601
N2E (Rel. error (%))	8.17	1.94	3.75
$\tau^* = 1$	29.47	172.76	194.24
$\tau^* = 4$	25.07	55.93	151.26
$\tau^* = 16$	24.35	20.44	31.88
$\tau^* = 64$	20.51	1.93	5.90
$\tau^* = 256$	10.58	0.37	1.07
$\tau^* = 1024$	2.79	1.39	1.24
$\tau^* = 4096$	9.02	6.76	6.28
$\tau^* = 16384$	36.19	24.89	17.45

Table 5. Relative error (%) of  $Q_{DD}$  on three datasets using N2E and different clipping thresholds  $\tau^*$ .

when comparing  $Q_{EC}$  and  $Q_{DD}$ , the optimal value of  $\tau^*$  can vary across queries on the same dataset. Despite this variation, the results show that the maximum degree approximation of N2E consistently delivers strong performance.

## 8 Future Work

One direction for future exploration is extending the N2E framework to hypergraphs. Unlike conventional graphs, where there is at most one edge between any pair of nodes, hypergraphs allow hyperedges that can connect multiple nodes, posing new challenges for privacy preservation. Another promising direction is adapting N2E to decentralized settings, where the graph data is distributed across multiple parties, and each party has access only to an individual node and its adjacent edges. A further potential direction is applying N2E to dynamic graphs, where the graph structure evolves over time. A key challenge in this setting is preserving utility, as each query on an updated graph consumes part of the privacy budget.

## Acknowledgments

This research is supported by the NTU–NAP Startup Grant (024584-00001) and the Singapore Ministry of Education Tier 1 Grant (RG19/25). We would also like to thank the anonymous reviewers who have made valuable suggestions on improving the presentation of the paper.

## References

- [1] Gergely Acs, Claude Castelluccia, and Rui Chen. 2012. Differentially private histogram publishing through lossy compression. In *2012 IEEE 12th International Conference on Data Mining*. IEEE, 1–10.
- [2] Jeremiah Blocki, Avrim Blum, Anupam Datta, and Or Sheffet. 2013. Differentially private data analysis of social networks via restricted sensitivity. In *Proceedings of the 4th conference on Innovations in Theoretical Computer Science*. 87–96.
- [3] Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. 2008. Fast unfolding of communities in large networks. *Journal of statistical mechanics: theory and experiment* 2008, 10 (2008), P10008.
- [4] Justin Y Chen, Badih Ghazi, Ravi Kumar, Pasin Manurangsi, Shyam Narayanan, Jelani Nelson, and Yinzhan Xu. 2023. Differentially private all-pairs shortest path distances: Improved algorithms and lower bounds. In *Proceedings of the 2023 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*. SIAM, 5040–5067.
- [5] Shixi Chen and Shuigeng Zhou. 2013. Recursive mechanism: towards node differential privacy and unrestricted joins. In *Proceedings of the 2013 ACM SIGMOD International Conference on Management of Data*. 653–664.
- [6] Graham Cormode, Cecilia Procopiuc, Divesh Srivastava, and Thanh TL Tran. 2012. Differentially private summaries for sparse data. In *Proceedings of the 15th International Conference on Database Theory*. 299–311.
- [7] Wei-Yen Day, Ninghui Li, and Min Lyu. 2016. Publishing graph degree distribution with node differential privacy. In *Proceedings of the 2016 International Conference on Management of Data*. 123–138.
- [8] Chengyuan Deng, Jie Gao, Jalaj Upadhyay, and Chen Wang. 2023. Differentially private range query on shortest paths. In *Algorithms and Data Structures Symposium*. Springer, 340–370.
- [9] Michael Dinitz, Satyen Kale, Silvio Lattanzi, and Sergei Vassilvskii. 2025. Almost Tight Bounds for Differentially Private Densest Subgraph. In *Proceedings of the 2025 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*. SIAM, 2908–2950.
- [10] Wei Dong, Zijun Chen, Qiyao Luo, Elaine Shi, and Ke Yi. 2024. Continual observation of joins under differential privacy. *Proceedings of the ACM on Management of Data* 2, 3 (2024), 1–27.
- [11] Wei Dong, Juanru Fang, Ke Yi, Yuchao Tao, and Ashwin Machanavajjhala. 2022. R2T: Instance-optimal Truncation for Differentially Private Query Evaluation with Foreign Keys. In *Proc. ACM SIGMOD International Conference on Management of Data*.
- [12] Wei Dong, Juanru Fang, Ke Yi, Yuchao Tao, and Ashwin Machanavajjhala. 2024. Instance-optimal Truncation for Differentially Private Query Evaluation with Foreign Keys. *ACM Transactions on Database Systems* 49, 4 (2024), 1–40.
- [13] Wei Dong, Dajun Sun, and Ke Yi. 2023. Better than Composition: How to Answer Multiple Relational Queries under Differential Privacy. *Proceedings of the ACM on Management of Data* 1, 2 (2023), 1–26.
- [14] Wei Dong and Ke Yi. 2021. Residual Sensitivity for Differentially Private Multi-Way Joins. In *Proc. ACM SIGMOD International Conference on Management of Data*.
- [15] Wei Dong and Ke Yi. 2022. A Nearly Instance-optimal Differentially Private Mechanism for Conjunctive Queries. In *Proc. ACM Symposium on Principles of Database Systems*.
- [16] Wei Dong and Ke Yi. 2023. Universal private estimators. In *Proceedings of the 42nd ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems*. 195–206.
- [17] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. 2006. Calibrating noise to sensitivity in private data analysis. In *Theory of cryptography conference*. Springer, 265–284.
- [18] Cynthia Dwork, Moni Naor, Omer Reingold, Guy N Rothblum, and Salil Vadhan. 2009. On the complexity of differentially private data release: efficient algorithms and hardness results. In *Proceedings of the forty-first annual ACM symposium on Theory of computing*. 381–390.
- [19] Cynthia Dwork and Aaron Roth. 2014. The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science* 9, 3–4 (2014), 211–407.
- [20] Talya Eden, Quanquan C Liu, Sofya Raskhodnikova, and Adam Smith. 2023. Triangle Counting with Local Edge Differential Privacy. In *50th International Colloquium on Automata, Languages, and Programming (ICALP 2023)*. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 52–1.
- [21] Chenglin Fan, Ping Li, and Xiaoyun Li. 2022. Private graph all-pairwise-shortest-path distance release with improved error rate. *Advances in Neural Information Processing Systems* 35 (2022), 17844–17856.
- [22] Juanru Fang, Wei Dong, and Ke Yi. 2022. Shifted Inverse: A General Mechanism for Monotonic Functions under User Differential Privacy. (2022).
- [23] Alireza Farhadi, MohammadTaghi Hajiaghayi, and Elaine Shi. 2022. Differentially private densest subgraph. In *International Conference on Artificial Intelligence and Statistics*. PMLR, 11581–11597.
- [24] Hendrik Fichtenberger, Monika Henzinger, and Wolfgang Ost. 2021. Differentially Private Algorithms for Graphs Under Continual Observation. In *29th Annual European Symposium on Algorithms (ESA 2021)*. Schloss Dagstuhl–Leibniz-Zentrum für Informatik.

- [25] Michael Hay, Chao Li, Gerome Miklau, and David Jensen. 2009. Accurate estimation of the degree distribution of private networks. In *2009 Ninth IEEE International Conference on Data Mining*. IEEE, 169–178.
- [26] Yizhang He, Kai Wang, Wenjie Zhang, Xuemin Lin, Ying Zhang, and Wei Ni. 2025. Robust Privacy-Preserving Triangle Counting under Edge Local Differential Privacy. In *Proc. ACM SIGMOD International Conference on Management of Data*. Association for Computing Machinery, New York, NY, USA.
- [27] Yihua Hu, Hao Ding, and Wei Dong. 2025. N2E: A General Framework to Reduce Node-Differential Privacy to Edge-Differential Privacy for Graph Analytics [Full Version]. (2025). <https://drive.google.com/drive/folders/1T9SLrU9z406RdMaWKPg3KP6aWEJ0o73B?usp=sharing>
- [28] Ziyue Huang, Yuting Liang, and Ke Yi. 2021. Instance-optimal Mean Estimation Under Differential Privacy. In *NeurIPS*.
- [29] Jacob Imola, Takao Murakami, and Kamalika Chaudhuri. 2021. Locally Differentially Private Analysis of Graph Statistics. In *30th USENIX Security Symposium (USENIX Security 21)*. USENIX Association, 983–1000.
- [30] Jacob Imola, Takao Murakami, and Kamalika Chaudhuri. 2022. Communication-Efficient Triangle Counting under Local Differential Privacy. In *31st USENIX Security Symposium (USENIX Security 22)*. USENIX Association, Boston, MA, 537–554.
- [31] Noah Johnson, Joseph P Near, and Dawn Song. 2018. Towards practical differential privacy for SQL queries. *Proceedings of the VLDB Endowment* 11, 5 (2018), 526–539.
- [32] Vishesh Karwa, Sofya Raskhodnikova, Adam Smith, and Grigory Yaroslavtsev. 2011. Private analysis of graph structure. *Proceedings of the VLDB Endowment* 4, 11 (2011), 1146–1157.
- [33] Shiva Prasad Kasiviswanathan, Kobbi Nissim, Sofya Raskhodnikova, and Adam Smith. 2013. Analyzing graphs with node differential privacy. In *Theory of Cryptography Conference*. Springer, 457–476.
- [34] Jure Leskovec and Andrej Krevl. 2014. SNAP: Stanford network analysis project.
- [35] Shang Liu, Hao Du, Yang Cao, Bo Yan, Jinfei Liu, and Masatoshi Yoshikawa. 2024. PGB: Benchmarking Differentially Private Synthetic Graph Generation Algorithms. *arXiv preprint arXiv:2408.02928* (2024).
- [36] Yuhua Liu, Sheng Wang, Yixuan Liu, Feifei Li, and Hong Chen. 2024. Unleash the Power of Ellipsis: Accuracy-enhanced Sparse Vector Technique with Exponential Noise. *arXiv preprint arXiv:2407.20068* (2024).
- [37] CPLEX User’s Manual. 1987. Ibm ilog cplex optimization studio. *Version 12, 1987-2018* (1987), 1.
- [38] Arvind Narayanan and Vitaly Shmatikov. 2009. De-anonymizing social networks. In *2009 30th IEEE symposium on security and privacy*. IEEE, 173–187.
- [39] Dung Nguyen and Anil Vullikanti. 2021. Differentially private densest subgraph detection. In *International Conference on Machine Learning*. PMLR, 8140–8151.
- [40] Kobbi Nissim, Sofya Raskhodnikova, and Adam Smith. 2007. Smooth sensitivity and sampling in private data analysis. In *Proceedings of the thirty-ninth annual ACM symposium on Theory of computing*. 75–84.
- [41] Wahbeh Qardaji, Weining Yang, and Ninghui Li. 2013. Understanding hierarchical methods for differentially private histograms. *Proceedings of the VLDB Endowment* 6, 14 (2013), 1954–1965.
- [42] Sina Sajadmanesh, Ali Shahin Shamsabadi, Aurélien Bellet, and Daniel Gatica-Perez. 2023. {GAP}: Differentially private graph neural networks with aggregation perturbation. In *32nd USENIX Security Symposium (USENIX Security 23)*. 3223–3240.
- [43] Alessandra Sala, Xiaohan Zhao, Christo Wilson, Haitao Zheng, and Ben Y Zhao. 2011. Sharing graphs using differentially private graph models. In *Proceedings of the 2011 ACM SIGCOMM conference on Internet measurement conference*. 81–98.
- [44] Adam Sealfon. 2016. Shortest paths and distances with differential privacy. In *Proceedings of the 35th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems*. 29–41.
- [45] Bang Wu, Xiangwen Yang, Shirui Pan, and Xingliang Yuan. 2021. Adapting membership inference attacks to GNN for graph classification: Approaches and implications. In *2021 IEEE International Conference on Data Mining (ICDM)*. IEEE, 1421–1426.
- [46] Jun Zhang, Graham Cormode, Cecilia M Procopiuc, Divesh Srivastava, and Xiaokui Xiao. 2015. Private release of graph statistics using ladder functions. In *Proceedings of the 2015 ACM SIGMOD international conference on management of data*. 731–745.
- [47] Sen Zhang, Weiwei Ni, and Nan Fu. 2020. Community preserved social graph publishing with node differential privacy. In *2020 IEEE International Conference on Data Mining (ICDM)*. IEEE, 1400–1405.
- [48] Xiaojian Zhang, Rui Chen, Jianliang Xu, Xiaofeng Meng, and Yingtao Xie. 2014. Towards accurate histogram publication under differential privacy. In *Proceedings of the 2014 SIAM international conference on data mining*. SIAM, 587–595.
- [49] Zhikun Zhang, Min Chen, Michael Backes, Yun Shen, and Yang Zhang. 2022. Inference attacks against graph neural networks. In *31st USENIX Security Symposium (USENIX Security 22)*. 4543–4560.

Received April 2025; revised July 2025; accepted August 2025