

Privacy-Preserving Triangle Counting in Signed Graphs

Zening Li

Beijing Institute of Technology
zening-li@outlook.com

Fusheng Jin

Beijing Institute of Technology
jfs21cn@bit.edu.cn

Rong-Hua Li

Beijing Institute of Technology
lironghuabit@126.com

Guoren Wang

Beijing Institute of Technology
wanggrbit@gmail.com

ABSTRACT

Triangle counting is a fundamental operator in graph analysis. Since graph data usually contain sensitive information about entities, the direct release of triangle counts poses privacy concerns. While recent studies have addressed privacy-preserving triangle counting, they mainly concentrate on unsigned graphs. In this paper, we study a new problem of developing triangle counting algorithms in signed graphs that satisfy centralized differential privacy and local differential privacy, respectively. The inclusion of edge signs and more classes of triangles leads to increased complexity and overwhelms the statistics with noise. To overcome this issue, we first propose a novel smooth-sensitivity computation algorithm to achieve differential privacy under the centralized model. In addition, to handle large signed graphs, we devise a computationally efficient function that calculates a smooth upper bound on local sensitivity. Finally, we release the approximate triangle counts after the addition of Laplace noise that is calibrated to the smooth upper bound on local sensitivity. In the local model, we propose a two-phase framework tailored for balanced and unbalanced triangle counting. The first phase utilizes the Generalized Randomized Response mechanism to perturb data, followed by a novel response mechanism in the second phase. Extensive experiments conducted over real-world datasets demonstrate that our proposed methods can achieve an excellent trade-off between privacy and utility.

PVLDB Reference Format:

Zening Li, Rong-Hua Li, Fusheng Jin, and Guoren Wang.
Privacy-Preserving Triangle Counting in Signed Graphs. PVLDB, 14(1):
XXX-XXX, 2020.
doi:XX.XX/XXX.XX

PVLDB Artifact Availability:

The source code, data, and/or other artifacts have been made available at
URL_TO_YOUR_ARTIFACTS.

1 INTRODUCTION

Many real-world relationships can be represented by signed graphs with positive and negative links, as a result of which signed graph analysis has attracted much attention and nurtured numerous applications [1, 22, 23, 27, 30, 31, 33]. Triangle counting is a fundamental

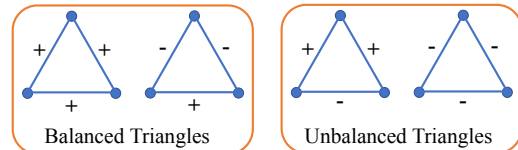


Figure 1: Examples of balanced and unbalanced triangles

primitive in signed graph analysis. A triangle is considered balanced if it encompasses an odd number of positive edges and unbalanced otherwise [3, 5] (refer to Figure 1 for illustration). Statistics based on balanced and unbalanced triangle counts reveal important structural information about signed graphs. And they are used to perform many computational tasks on signed graphs, such as link prediction [22], network reconstruction [27], and community detection [19, 31].

However, most real-world signed graphs associated with people or human-related activities, such as social and economic networks, hold sensitive information about the relationships between individuals. These positive or negative relationships reveal intricate details about personal interactions, preferences, and social dynamics. For instance, economic transaction networks between financial institutions can be modeled to reflect lending relationships, where the signs indicate the health of these transactions, i.e., timely repayments or defaults. The direct release of balanced and unbalanced triangle counts in such networks can inadvertently disclose sensitive relationship patterns, which raises privacy concerns. Therefore, it is necessary to develop privacy-preserving triangle counting algorithms.

Differential privacy (DP) [10] has recently become the dominant paradigm for protecting individual privacy in data analysis and offers robust privacy protections in the presence of powerful and realistic adversaries. It requires that the output distributions of a differentially private query are similar for databases that differ in only one entry. Differential privacy can be bifurcated into centralized DP (CDP) and local DP (LDP) based on the underlying architecture. In the centralized model, a trusted curator holds all the sensitive data and releases sanitized versions of the statistics. In contrast, local DP operates under the assumption that the data curator is untrusted. Under this premise, each user perturbs his/her data locally and transmits the perturbed data to the untrusted data curator. Numerous studies have been conducted on differentially private graph analysis algorithms, spanning both centralized [4, 7, 20, 21, 26, 40] and local models [14–16, 28, 29, 38]. Despite the considerable advancements in this field, it is noteworthy that these studies focus on unsigned graphs. The exploration of balanced and unbalanced triangle counting in signed graphs, which is the focus of this paper, remains underexplored in the current literature.

This work is licensed under the Creative Commons BY-NC-ND 4.0 International License. Visit <https://creativecommons.org/licenses/by-nc-nd/4.0/> to view a copy of this license. For any use beyond those covered by this license, obtain permission by emailing info@vldb.org. Copyright is held by the owner/author(s). Publication rights licensed to the VLDB Endowment.

Proceedings of the VLDB Endowment, Vol. 14, No. 1 ISSN 2150-8097.
doi:XX.XX/XXX.XX

To fill this gap, in this paper, we study the problem of balanced and unbalanced triangle counting under centralized DP and local DP, where both graph structure and the sign of edges are considered private information. The inclusion of edge signs introduces unique challenges for privacy preservation. (i) Noise scale: a standard DP technique directly adds noise to the statistical estimates. However, as the noise scale is proportional to the number of nodes, it causes excessive perturbation to the release statistics. (ii) Complexity of smooth sensitivity: to reduce the introduction of noise, we explore the technique of smooth sensitivity as an alternative. However, the computation of smooth sensitivity is notably more complex in signed graphs. (iii) Scalability: beyond the computational complexity, the calculation of smooth sensitivity is also extremely expensive.

To solve the mentioned issues, we propose several effective approaches under centralized DP and local DP, respectively. In the centralized model, in order to reduce the introduction of noise, we employ the smooth sensitivity framework to compute the number of balanced and unbalanced triangles in a signed graph. Specifically, we first analyze the local sensitivity and local sensitivity at distance t of the triangle count query function. Then, we propose a novel and efficient method for the computation of smooth sensitivity, with a time complexity of $O(m \cdot d_{\max} + n^2)^1$. Furthermore, to handle large signed graphs, we devise a computationally efficient function that calculates a smooth upper bound on local sensitivity, with a reduced time complexity of $O(m \cdot d_{\max})$. The final step is to release triangle counts after adding Laplace noise calibrated to the smooth upper bound on local sensitivity. In the local model, we propose a two-phase framework tailored for balanced and unbalanced triangle counting. The first phase employs the Generalized Randomized Response mechanism to perturb the data. In the second phase, we propose a novel response mechanism with an unbiased correction to the final statistics. This mechanism injects noise into the query response of each node, which is also calibrated to the smooth upper bound on local sensitivity. Finally, to evaluate the effectiveness of our proposed approaches, we conduct extensive experiments on various real-world datasets. The results show that our proposed methods establish state-of-the-art performance and achieve a better privacy-utility trade-off than baseline methods. To summarize, our main contributions of this paper are as follows:

- We propose innovative approaches for estimating the number of balanced and unbalanced triangles under both centralized DP and local DP. To the best of our knowledge, this is the first exploration of privacy-preserving triangle counting in signed graphs.
- In the centralized model, we propose a novel and efficient method for the computation of smooth sensitivity. To handle the large signed graphs, we devise a computationally efficient function that calculates a smooth upper bound on local sensitivity, reducing the computational complexity from $O(m \cdot d_{\max} + n^2)$ to $O(m \cdot d_{\max})$.
- In the local model, we propose a two-phase framework for triangle counting. The second phase includes a novel response mechanism that allows each node to independently add noise calibrated to the smooth upper bound on local sensitivity to its reported statistics.
- We conduct extensive experiments on five real-world datasets. The experimental results demonstrate that our proposed methods

achieve excellent privacy-utility trade-offs in both centralized and local models.

2 PRELIMINARIES

In this section, we introduce the formal definition of the research problem, followed by a concise overview of differential privacy and its properties.

2.1 Problem Statement

We consider an undirected signed graph $G = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{v_1, v_2, \dots, v_n\}$ denotes the set of n nodes and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ represents the set of m edges. Each edge in G carries a sign "+" or "-". An edge with the sign "+" denotes a positive edge, while an edge with the sign "-" represents a negative edge. The degree of v_i is expressed as d_i , and the maximum degree is denoted as d_{\max} . Let \mathcal{G} be the collection of all possible signed graphs on n nodes. Any signed graph $G \in \mathcal{G}$ can also be represented as a symmetric adjacency matrix $\mathbf{A} \in \{0, +1, -1\}^{n \times n}$, where $a_{i,j} = +1$ indicates a positive edge, $a_{i,j} = -1$ denotes a negative edge, and $a_{i,j} = 0$ means no edge exists between v_i and v_j . Let $\mathbf{a}_i = (a_{i,1}, \dots, a_{i,n}) \in \{0, +1, -1\}^n$ denote the adjacency vector of the node v_i , which corresponds to the i -th row of the adjacency matrix \mathbf{A} .

In this paper, we consider a query function $f_{\Delta}: \mathcal{G} \rightarrow (T^b, T^u)$ that takes a signed graph $G \in \mathcal{G}$ as input and outputs the number of balanced and unbalanced triangles, denoted as T^b and T^u . Note that we assume the set of nodes \mathcal{V} is public, so the identities of the nodes are non-private. Similar assumptions are widely made in previous privacy-preserving triangle counting studies [14, 16, 20, 26, 40]. In contrast, the set of edges \mathcal{E} and the signs of edges are considered sensitive information. Our ultimate objective is to develop algorithms that can accurately estimate the balanced and unbalanced triangle counts while protecting individual privacy.

2.2 Differential Privacy for Signed Graphs

Differential privacy [10, 11] has become the standard framework for data release that offers robust privacy protection in the presence of powerful and realistic adversaries. A randomized algorithm satisfies DP when the distributions of its outputs are similar for any pair of neighboring databases. Hence, the formal definition of differential privacy revolves around the concept of neighboring databases. In this paper, we employ the edit distance to measure the distinction between two databases. The edit distance quantifies the minimum number of operations required to transform one object into another. We exclusively focus on edge operations since the graph structure and edge signs are considered private information.

Centralized Differential Privacy. In the centralized model, a trusted curator first collects the private data from all users and then releases the sanitized statistical data. Next, we formally define centralized differential privacy for signed graphs.

Definition 2.1 (Distance between signed graphs, Neighbors). The distance between n -node signed graphs G and \tilde{G} , denoted as $d(G, \tilde{G})$, is the minimum number of primitive operations required to convert G to \tilde{G} . There are three types of primitive edit operations: (1) edge insertion to introduce a new signed edge between a pair of nodes; (2) edge deletion to remove a single edge between a pair of nodes;

¹Note that m denotes the number of edges, n represents the number of nodes, and d_{\max} is the maximum degree.

(3) edge substitution to alter the sign of a given edge. Signed graphs G and \tilde{G} are considered neighbors if $d(G, \tilde{G}) = 1$.

Definition 2.2 (Signed Edge Centralized Differential Privacy). Given $\epsilon > 0$ and $\delta > 0$, a randomized algorithm \mathcal{A} satisfies (ϵ, δ) -signed edge CDP if and only if for any two neighboring signed graphs $G, \tilde{G} \in \mathcal{G}$, and for any possible output $O \in \text{Range}(\mathcal{A})$, we have

$$\Pr[\mathcal{A}(G) = O] \leq e^\epsilon \cdot \Pr[\mathcal{A}(\tilde{G}) = O] + \delta. \quad (1)$$

The parameter ϵ is the privacy budget that controls the trade-off between privacy protection level and utility. The parameter δ is treated as the probability of failure and is usually chosen to be much smaller than the inverse of the number of data records. When $\delta = 0$, the algorithm is ϵ -signed edge centralized differential privacy.

Local Differential Privacy. In contrast to centralized DP, local differential privacy [8] operates under the assumption of a local model, where each user does not trust the data curator. In this model, each user locally perturbs his/her own data by a randomized perturbation algorithm and sends the obfuscated data to the untrusted data curator.

Definition 2.3 (Distance between adjacency vectors, Neighbors). The distance between two adjacency vectors $\mathbf{a}_i, \tilde{\mathbf{a}}_i \in \{0, +1, -1\}^n$, denoted as $d(\mathbf{a}_i, \tilde{\mathbf{a}}_i)$, is defined as the least number of elementary operations required to transform \mathbf{a}_i into $\tilde{\mathbf{a}}_i$. These operations include (1) the insertion of a signed edge connected to v_i ; (2) the removal of an edge from node v_i ; (3) the alteration of the sign of an edge linked to v_i . Adjacency vectors \mathbf{a}_i and $\tilde{\mathbf{a}}_i$ are neighbors when $d(\mathbf{a}_i, \tilde{\mathbf{a}}_i) = 1$.

Definition 2.4 (Signed Edge Local Differential Privacy). Given $\epsilon > 0$ and $\delta > 0$, a randomized algorithm \mathcal{A} satisfies (ϵ, δ) -signed edge LDP if and only if for any two neighboring adjacency vectors \mathbf{a}_i and $\tilde{\mathbf{a}}_i$, and for any possible output O of \mathcal{A} , we have

$$\Pr[\mathcal{A}(\mathbf{a}_i) = O] \leq e^\epsilon \cdot \Pr[\mathcal{A}(\tilde{\mathbf{a}}_i) = O] + \delta. \quad (2)$$

Similarly, when $\delta = 0$, \mathcal{A} satisfies ϵ -signed edge LDP.

2.3 Differential Privacy Mechanisms

Various methods have been proposed to achieve differential privacy. One prevalent approach is to inject carefully chosen random noise into the true results. In this paper, we utilize the Laplace distribution to add noise. A Laplace random variable with mean 0 and standard deviation $\sqrt{2\lambda}$ has density $h(z) = \frac{1}{2\lambda} e^{-|z|/\lambda}$, denoted as $\text{Lap}(\lambda)$. The Laplace mechanism [10], a fundamental technique that satisfies ϵ -DP, introduces independent and identically distributed noise from $\text{Lap}(GS_f/\epsilon)$ to each element of the output produced by a query function $f: \mathcal{D} \rightarrow \mathbb{R}^k$, where GS_f is the global sensitivity of f .

Definition 2.5 (Global Sensitivity [10]). Given a query function $f: \mathcal{D} \rightarrow \mathbb{R}^k$, the global sensitivity of f is defined as

$$GS_f = \max_{x, \tilde{x} \in \mathcal{D}: d(x, \tilde{x})=1} \|f(x) - f(\tilde{x})\|_1. \quad (3)$$

where $\|\cdot\|_1$ represents the ℓ_1 norm.

The amount of noise added by the Laplace mechanism depends on GS_f and the privacy budget ϵ . It is crucial to note that GS_f is an inherent characteristic of the function f , independent of any input data. However, for the query functions considered in this article, this mechanism introduces a substantial amount of noise to the results,

which can compromise the performance. In [26], the authors propose a local measure of sensitivity.

Definition 2.6 (Local Sensitivity [26]). The local sensitivity of a function $f: \mathcal{D} \rightarrow \mathbb{R}^k$ on a database $x \in \mathcal{D}$ is

$$LS_f(x) = \max_{\tilde{x} \in \mathcal{D}: d(x, \tilde{x})=1} \|f(x) - f(\tilde{x})\|_1. \quad (4)$$

From the above definition, it follows that the local sensitivity is not only related to the query function f , but also to the concrete instance x . For many problems, $LS_f(x)$ tends to be smaller than GS_f [12, 20, 21, 26]. Unfortunately, local sensitivity does not satisfy the requirement of differential privacy, since $LS_f(x)$ itself contains information about the database. To solve this issue, Nissim et al. [26] employ a β -smooth upper bound on local sensitivity, rather than the local sensitivity itself, for noise calibration. Specifically, for $\beta > 0$, a function $S_{f,\beta}: \mathcal{D} \rightarrow \mathbb{R}$ is a β -smooth upper bound on local sensitivity of f , if $\forall x \in \mathcal{D}$, $S_{f,\beta}(x) \geq LS_f(x)$ and $\forall x, \tilde{x} \in \mathcal{D}$ with $d(x, \tilde{x}) = 1$, $S_{f,\beta}(x) \leq e^\beta S_{f,\beta}(\tilde{x})$. $LS_f(x)$ may have multiple smooth bounds, and the smooth sensitivity is the smallest one that meets the condition.

Definition 2.7 (Smooth Sensitivity [26]). For any query function $f: \mathcal{D} \rightarrow \mathbb{R}^k$ and $\beta > 0$, the β -smooth sensitivity of f at $x \in \mathcal{D}$ is

$$S_{f,\beta}^*(x) = \max_{\tilde{x} \in \mathcal{D}} (e^{-\beta d(x, \tilde{x})} \cdot LS_f(\tilde{x})). \quad (5)$$

To compute smooth sensitivity efficiently, we construct a function known as the local sensitivity at distance t . This function, denoted as $LS_f(x, t)$, can be calculated as follows:

$$LS_f(x, t) = \max_{\tilde{x} \in \mathcal{D}: d(x, \tilde{x}) \leq t} LS_f(\tilde{x}). \quad (6)$$

Here, $LS_f(x, t)$ represents the maximum local sensitivity $LS_f(\tilde{x})$ of all databases \tilde{x} where the maximum distance between \tilde{x} and x is t . In other words, we permit at most t modifications to the database x before computing its local sensitivity. Then the β -smooth sensitivity $S_{f,\beta}^*(x)$ can be expressed in terms of $LS_f(x, t)$ as

$$S_{f,\beta}^*(x) = \max_{t=0,1,\dots,n} e^{-\beta t} LS_f(x, t). \quad (7)$$

THEOREM 2.8 (NOISE CALIBRATION TO SMOOTH BOUND [26]). Given a query function $f: \mathcal{D} \rightarrow \mathbb{R}^k$, suppose $S_{f,\beta}(x)$ is a β -smooth upper bound on local sensitivity of f , where $\beta = \frac{\epsilon}{4(k+\ln(2/\delta))}$. Then the algorithm $\mathcal{A}_{f,\epsilon}(x) = f(x) + (z_1, \dots, z_k)$ satisfies (ϵ, δ) -DP, where the z_i are drawn i.i.d. from $\text{Lap}(2S_{f,\beta}(x)/\epsilon)$.

Generalized Randomized Response (GRR) [18] is another pre-dominant perturbation mechanism that satisfies ϵ -LDP. Specifically, each user locally perturbs their private discrete value $x \in \mathcal{D}$ via GRR mechanism. This mechanism outputs the true value x with probability $\frac{e^\epsilon}{e^\epsilon + k - 1}$, or an arbitrary value $\tilde{x} \neq x$ with probability $\frac{1}{e^\epsilon + k - 1}$, where $k = |\mathcal{D}|$ is the domain size.

Moreover, differential privacy has some important properties that assist in the development of sophisticated algorithms.

PROPOSITION 2.9 (SEQUENTIAL COMPOSITION [9]). Given the sequence of computations $\mathcal{A}_1, \dots, \mathcal{A}_k$, if each \mathcal{A}_i satisfies (ϵ_i, δ_i) -DP, then their sequential execution on the same dataset satisfies $(\sum_i \epsilon_i, \sum_i \delta_i)$ -DP.

PROPOSITION 2.10 (POST-PROCESSING [9]). Suppose \mathcal{A} satisfies (ϵ, δ) -DP, and let \mathcal{B} be an arbitrary deterministic mapping function. Then the composed algorithm $\mathcal{B}(\mathcal{A}(\cdot))$ satisfies (ϵ, δ) -DP.

3 THE PROPOSED CENTRALIZED METHOD

In this section, we propose methods that utilize the smooth sensitivity framework to estimate the number of balanced and unbalanced triangles in signed graphs under centralized DP. Our approaches not only protect individual privacy but also maintain excellent query performance. First of all, we present techniques to compute the local sensitivity and the local sensitivity at distance t of the triangle count query function, which is the foundation for the implementation of differential privacy. Then, we propose efficient algorithms for the calculation of smooth upper bounds on local sensitivity, a critical factor in reducing the impact of data perturbations. Of these bounds, smooth sensitivity is the smallest upper bound. However, its calculation becomes impractical for large signed graphs due to substantial computational demands. In such cases, we choose a computationally feasible smooth upper bound as an alternative to smooth sensitivity. Consequently, we devise a computationally efficient function that calculates the smooth upper bound on local sensitivity of the triangle count $f_\Delta(G)$. In the end, we inject noise proportional to the smooth upper bound on local sensitivity into the response of the triangle counting query. For small signed graphs, the method based on smooth sensitivity offers superior utility than that based on the smooth upper bound on local sensitivity. In contrast, for large signed graphs, the approach based on the smooth upper bound on local sensitivity not only ensures computational efficiency but also achieves comparable utility to the smooth-sensitivity-based method.

In the remainder of this paper, we use $[n]$ to represent the set $\{1, 2, \dots, n\}$ and $[a, b]$ to denote $\{a, a+1, \dots, b\}$.

3.1 The Calculation of Local Sensitivity

This section elucidates the methods employed to compute the local sensitivity and the local sensitivity at distance t of the function f_Δ . This function takes a signed graph G as input and yields the number of balanced triangles T^b and unbalanced triangles T^u .

Local Sensitivity. The local sensitivity of f_Δ , denoted as $LS_\Delta(G)$, is defined as

$$LS_\Delta(G) = \max_{\tilde{G} \in \mathcal{G}: d(G, \tilde{G}) \leq 1} |\tilde{T}^b - T^b| + |\tilde{T}^u - T^u|. \quad (8)$$

To compute the local sensitivity efficiently, we introduce a more specialized definition of local sensitivity, denoted as $LS_{i,j}(G)$. This metric assesses the maximum impact on the sensitivity of modifying the relationship between nodes v_i and v_j . Such modifications entail the addition of a signed edge (v_i, v_j) in its absence or, if the edge exists, altering its sign or deleting it. Let $w_{i,j}^+$ denote the number of positive wedges (two-hop paths) between nodes v_i and v_j where the product of $a_{i,k}$ and $a_{j,k}$ is positive. In contrast, $w_{i,j}^-$ represents the number of negative wedges where this product is negative.

LEMMA 3.1. *The local sensitivity of f_Δ is given by $LS_\Delta(G) = \max_{i,j \in [n]: i \neq j} LS_{i,j}(G)$, where*

$$LS_{i,j}(G) = \begin{cases} w_{i,j}^+ + w_{i,j}^-, & \text{if } a_{i,j} = 0, \\ \max(w_{i,j}^+ + w_{i,j}^-, 2|w_{i,j}^+ - w_{i,j}^-|), & \text{if } a_{i,j} \neq 0. \end{cases} \quad (9)$$

PROOF. We start by analyzing $LS_{i,j}(G)$. For the case $a_{i,j} = 0$, the addition of the signed edge between nodes v_i and v_j changes $f_\Delta(G)$ by $w_{i,j}^+ + w_{i,j}^-$, irrespective of its sign. Now consider the case where $a_{i,j} \neq 0$. If the edge (v_i, v_j) is deleted, it can be easily verified that f_Δ

Algorithm 1: Edge-Scan for Wedge Counting

Input: signed graph G represented as adjacency vectors $\{a_1, \dots, a_n\}$
Output: $\{(w_{i,j}^+, w_{i,j}^-)\}_{i,j \in [n]: i < j}$

- 1 Initialize $w_{i,j}^+ \leftarrow 0, w_{i,j}^- \leftarrow 0$ for each node pair (v_i, v_j) ;
- 2 **for** each node $v_i \in \mathcal{V}$ **do**
- 3 **for** each pair of edges $(v_i, v_j), (v_i, v_k)$ incident to v_i **do**
- 4 /* suppose $j < k$ */
- 5 **if** $a_{i,j} \cdot a_{i,k} = 1$ **then**
- 6 $w_{j,k}^+ \leftarrow w_{j,k}^+ + 1$;
- 7 **else**
- 8 $w_{j,k}^- \leftarrow w_{j,k}^- + 1$;
- 9 **return** $\{(w_{i,j}^+, w_{i,j}^-)\}_{i,j \in [n]: i < j}$

also changes by $w_{i,j}^+ + w_{i,j}^-$. If we alter the sign of edge (v_i, v_j) , the number of balanced and unbalanced triangles over the edge (v_i, v_j) interchange, which results in a change in f_Δ by $2|w_{i,j}^+ - w_{i,j}^-|$. Finally, $LS_\Delta(G)$ is the maximum of all $LS_{i,j}(G)$. \square

Algorithm 1 describes the process of computing the number of wedges between all pairs of nodes. It iterates over each node $v_i \in \mathcal{V}$ and then enumerates all edge pairs connected to v_i to determine the number of positive and negative wedges between specific node pairs. Since the signed graphs are undirected, it follows that $w_{i,j}^+ = w_{j,i}^+$, and the same holds for $w_{i,j}^-$. It is sufficient to compute the number of wedges for the node pairs v_i and v_j whose node indexes satisfy $i < j$. Let Γ represent the number of node pairs where $w_{i,j}^+ \neq 0$ or $w_{i,j}^- \neq 0$, expressed as $\Gamma = \sum_{i=0}^n \sum_{j=i+1}^n \mathbb{I}(w_{i,j}^+ \neq 0 \text{ or } w_{i,j}^- \neq 0)$. Here, the symbol $\mathbb{I}(\cdot)$ is an indicator function that returns 1 if the condition inside the parentheses is true and 0 if it is false. The time complexity of Algorithm 1 is $O(m \cdot d_{\max})$. And the computation of $LS_\Delta(G)$ can be accomplished in time $O(\Gamma)$.

Local Sensitivity at Distance. In the rest of this section, we explain the computation of the local sensitivity at distance t of f_Δ , denoted as $LS_\Delta(G, t)$. For the special scenario where $t = 0$, $LS_\Delta(G, t)$ represents the local sensitivity of f_Δ at G . When $s \geq 1$, to compute $LS_\Delta(G, t)$ efficiently, we define $LS_{i,j}(G, t)$ as the maximum of $LS_{i,j}(\cdot)$ attained on signed graphs whose distance from G are at most t , expressed as

$$LS_{i,j}(G, t) = \max_{\tilde{G} \in \mathcal{G}: d(G, \tilde{G}) \leq t} LS_{i,j}(\tilde{G}). \quad (10)$$

Thus, $LS_\Delta(G, t) = \max_{i,j \in [n]: i < j} LS_{i,j}(G, t)$. As stated in Lemma 3.1, the calculation of local sensitivity for \tilde{G} over the node pair (v_i, v_j) depends on whether $\tilde{a}_{i,j}$ equals 0 or not. However, we can first introduce a signed edge (v_i, v_j) if it is absent. Therefore, regardless of the existence of edge (v_i, v_j) , we need to explore modification strategies that can maximize $\tilde{w}_{i,j}^+ + \tilde{w}_{i,j}^-$ and $2|\tilde{w}_{i,j}^+ - \tilde{w}_{i,j}^-|$ in \tilde{G} , respectively. For convenience, we employ $W_{i,j}^1(G, t)$ to represent the maximum of $\tilde{w}_{i,j}^+ + \tilde{w}_{i,j}^-$, and $W_{i,j}^2(G, t)$ to denote the maximum of $2|\tilde{w}_{i,j}^+ - \tilde{w}_{i,j}^-|$.

To comprehend $W_{i,j}^1(G, t)$ for positive t , we modify t edges in G to obtain a signed graph \tilde{G} where $\tilde{w}_{i,j}^+ + \tilde{w}_{i,j}^-$ reaches its maximum. It is crucial to emphasize that among the various modifications to G , the only adjustment that has the potential to increase $w_{i,j}^+ + w_{i,j}^-$ is the addition of edges adjacent to either nodes v_i or v_j . The sign of the added edge does not affect the overall change. Therefore, \tilde{G} should be obtained by strategic addition of t edges to G . The specific

Algorithm 2: Local Sensitivity at Distance

Input: signed graph G represented as adjacency vectors $\{a_1, \dots, a_n\}$, distance $t \geq 1$, $\{(w_{i,j}^+, w_{i,j}^-)\}_{i,j \in [n], i < j}$

Output: local sensitivity at distance t of f_Δ : $LS_\Delta(G, t)$

```

1 Initialize modification volume  $c \leftarrow t$ ,  $LS_\Delta(G, t) \leftarrow 0$ ;
2 for each node pair  $(v_i, v_j)$ , where  $i, j \in [n]$ ,  $i < j$  do
3    $t \leftarrow c$ ;
4    $b_{i,j} \leftarrow d_i + d_j - 2(w_{i,j}^+ + w_{i,j}^- + |a_{i,j}|)$ ;
5   if  $b_{i,j} \geq t$  then  $W_{i,j}^1(G, t) \leftarrow w_{i,j}^+ + w_{i,j}^- + t$ ;
6   else  $W_{i,j}^1(G, t) \leftarrow w_{i,j}^+ + w_{i,j}^- + \lfloor (t + b_{i,j})/2 \rfloor$ ;
7    $W_{i,j}^1(G, t) \leftarrow \min(W_{i,j}^1(G, t), n - 2)$ ;
8   if  $a_{i,j} = 0$  then  $t \leftarrow t - 1$ ;
9   if  $\min(w_{i,j}^+, w_{i,j}^-) \geq t$  then
10     $W_{i,j}^2(G, t) \leftarrow 2(|w_{i,j}^+ - w_{i,j}^-| + 2t)$ ;
11  else
12     $W_{i,j}^2(G, t) \leftarrow 2(|w_{i,j}^+ - w_{i,j}^-| + 2\min(w_{i,j}^+, w_{i,j}^-))$ ;
13     $t \leftarrow t - \min(w_{i,j}^+, w_{i,j}^-)$ ;
14    if  $b_{i,j} \geq t$  then  $W_{i,j}^2(G, t) \leftarrow W_{i,j}^2(G, t) + 2t$ ;
15    else  $W_{i,j}^2(G, t) \leftarrow W_{i,j}^2(G, t) + 2\lfloor (t + b_{i,j})/2 \rfloor$ ;
16   $W_{i,j}^2(G, t) \leftarrow \min(W_{i,j}^2(G, t), 2n - 4)$ ;
17   $LS_{i,j}(G, t) \leftarrow \max(W_{i,j}^1(G, t), W_{i,j}^2(G, t))$ ;
18   $LS_\Delta(G, t) \leftarrow \max(LS_\Delta(G, t), LS_{i,j}(G, t))$ ;
19 return  $LS_\Delta(G, t)$ 

```

allocation strategy is described in Lines 5-7 of Algorithm 2. Let $b_{i,j}$ denote the number of nodes connected to exactly one of two nodes v_i and v_j . For these nodes v_k , adding one edge, i.e., the missing edge (v_i, v_k) or (v_j, v_k) , is sufficient to increase $w_{i,j}^+ + w_{i,j}^-$ by 1. Conversely, for nodes v_k that are not adjacent to both v_i and v_j , the addition of two edges, i.e., (v_i, v_k) and (v_j, v_k) , is required to achieve the same increase. Thus, if the distance $t \leq b_{i,j}$, the increase of $w_{i,j}^+ + w_{i,j}^-$ is at most t , as indicated in Line 5. However, a nuanced allocation approach is required when $t > b_{i,j}$. First, we can add $b_{i,j}$ edges so that $w_{i,j}^+ + w_{i,j}^-$ increases $b_{i,j}$. Then, for the rest of $t - b_{i,j}$ modifications, we can find $\lfloor (t - b_{i,j})/2 \rfloor$ nodes v_k , where $a_{i,k} = a_{j,k} = 0$, and add the edges (v_i, v_k) and (v_j, v_k) to increase $w_{i,j}^+ + w_{i,j}^-$ by $\lfloor (t - b_{i,j})/2 \rfloor$, achieving a total increase of $w_{i,j}^+ + w_{i,j}^-$ by $\lfloor (t + b_{i,j})/2 \rfloor$, as stated in Line 6. Note that the upper bound of $w_{i,j}^+ + w_{i,j}^-$ is $n - 2$. This constraint ensures that the modifications adhere to the inherent structure of the graph. Next, we introduce the calculation of $W_{i,j}^2(G, t)$. Recall that $W_{i,j}^2(G, t)$ needs to be computed only if $a_{i,j} \neq 0$. However, in cases where $a_{i,j} = 0$, we can introduce an edge (v_i, v_j) beforehand. As a result, the total number of modifications required reduces to $t - 1$, as outlined in Line 8.

Different from $W_{i,j}^1(G, t)$, $W_{i,j}^2(G, t)$ represents the maximum of $2|w_{i,j}^+ - w_{i,j}^-|$ across all signed graphs \tilde{G} that satisfy $d(G, \tilde{G}) \leq t$. Thus, our modifications to G should increase $2|w_{i,j}^+ - w_{i,j}^-|$ as much as possible. The only effective modification is to operate on edges adjacent to nodes v_i or v_j . Specifically, the addition or deletion of one edge can result in a maximum increase of 2, whereas the alteration of the sign of an edge yields an increase of up to 4. In this paper, we adopt a greedy strategy to calculate $W_{i,j}^2(G, t)$, as outlined in Lines 9-16 of Algorithm 2. The approach prioritizes edge sign alterations until further modifications cease to yield increases since they offer higher potential increases.

In the context of different nodes v_i and v_j , it is essential to consider two scenarios due to the presence of an absolute: either $w_{i,j}^+ \geq w_{i,j}^-$ or $w_{i,j}^+ < w_{i,j}^-$. However, the analytical approach for both scenarios remains consistent. For simplicity, we assume $w_{i,j}^+ \geq w_{i,j}^-$ so that the objective function reduces to $2(w_{i,j}^+ - w_{i,j}^-)$. If $t \leq w_{i,j}^-$, it is possible to increase the objective by $4t$, as described in Line 10. This increase can be achieved by reversing the sign of t edges, where each edge belongs to a different negative wedge, and each adjustment contributes an increment of 4 to $2(w_{i,j}^+ - w_{i,j}^-)$. Consider the scenario where $t > w_{i,j}^-$. We can first reverse the sign of $w_{i,j}^-$ edges to increase $2(w_{i,j}^+ - w_{i,j}^-)$ by $4w_{i,j}^-$. Then the only feasible modification to further increase $2(w_{i,j}^+ - w_{i,j}^-)$ is to add edges connected to either nodes v_i or v_j . Hence, for the rest of $t - w_{i,j}^-$ modifications, the modification procedure is similar to that employed in the calculation of $W_{i,j}^1(G, t)$, and the sole difference is that the increment becomes twice as large, as outlined in Lines 14-15. Different from $w_{i,j}^+ + w_{i,j}^-$, the upper bound of $2(w_{i,j}^+ - w_{i,j}^-)$ is $2n - 4$. Upon the completion of calculations for $W_{i,j}^1(G, t)$ and $W_{i,j}^2(G, t)$, it becomes feasible to determine the local sensitivity at the distance t over the edge (v_i, v_j) , as stated in Line 17. This procedure is then iteratively applied to all node pairs (v_i, v_j) to compute $LS_\Delta(G, t)$. Algorithm 2 encompasses all the steps involved in the calculation of the local sensitivity at distance t of f_Δ . And the time complexity of this algorithm is $O(n^2)$.

3.2 Smooth Upper Bound on Local Sensitivity

The previous subsection details the calculation of the local sensitivity and the local sensitivity at distance t of the triangle count query function. Next, we introduce methods to calculate the smooth upper bounds on local sensitivity based on the above computation.

Smooth Sensitivity. Smooth sensitivity is the smallest upper bound that satisfies the criterion for a smooth upper bound on local sensitivity. Thus, the mechanism that utilizes smooth sensitivity to calibrate noise can add less noise and provide better utility. For $\beta > 0$, the smooth sensitivity of f_Δ , denoted as $S_{\Delta, \beta}^*(G)$, is expressed as

$$S_{\Delta, \beta}^*(G) = \max_{t \in [0, 2n-3]} e^{-\beta t} LS_\Delta(G, t). \quad (11)$$

Since $LS_\Delta(G, t)$ remains constant for all $t \geq 2n - 3$, it is sufficient to consider $t \leq 2n - 3$. To compute the smooth sensitivity, one can calculate $e^{-\beta t} LS_\Delta(G, t)$ for each relevant t and select the maximal value as the smooth sensitivity of f_Δ . However, the computational complexity of $LS_\Delta(G, t)$, denoted by $O(n^2)$, coupled with the need to evaluate each possible value of t , raises the overall time complexity to $O(n^3)$. This complexity makes the method impractical for most applications.

In the calculation of smooth sensitivity, the predominant source of time complexity is the computation of local sensitivity at distance t , which involves numerous redundant and unproductive calculations. To address this issue, we propose an efficient approach to compute the smooth sensitivity of f_Δ . This method eliminates duplicate and irrelevant data based on our established criteria and retains only those elements that are critical to the smooth sensitivity computation. Then, we can determine the local sensitivity at each distance t sequentially, thereby substantially reducing computational overhead.

First, we prepare the requisite data for the computation of smooth sensitivity, described in Lines 2-7 of Algorithm 3. Specifically, we compute $w_{i,j}^+$, $w_{i,j}^-$ and $b_{i,j}$ for each node pair (v_i, v_j) whose node

IDs satisfy $i < j$. The calculations of $w_{i,j}^+$ and $w_{i,j}^-$ are performed in accordance with Algorithm 1, and $b_{i,j}$ is determined via the formula $b_{i,j} = d_i + d_j - 2(w_{i,j}^+ + w_{i,j}^- + |a_{i,j}|)$. For node pairs (v_i, v_j) with $w_{i,j}^+ \neq 0$ or $w_{i,j}^- \neq 0$, we construct a list Q consisting of triplets $(w_{i,j}^+, w_{i,j}^-, b_{i,j})$. For node pairs where both $w_{i,j}^+$ and $w_{i,j}^-$ equal zero, we choose the maximum of $b_{i,j}$ and incorporate the triplet $(0, 0, b_{i,j})$ into the list Q .

To compute local sensitivity at distance t efficiently, we propose a series of rules to filter out the duplicate and irrelevant triplets in Q . Recall that the computation of the local sensitivity at distance t primarily considers two cases: the maximum of $\tilde{w}_{i,j}^+ + \tilde{w}_{i,j}^-$ and the maximum of $2|\tilde{w}_{i,j}^+ - \tilde{w}_{i,j}^-|$. Let $W^1(G, t)$ denote the maximal value of $\tilde{w}_{i,j}^+ + \tilde{w}_{i,j}^-$ across all \tilde{G} which satisfy $d(G, \tilde{G}) \leq t$, and define $W^2(G, t)$ as the maximal value of $2|\tilde{w}_{i,j}^+ - \tilde{w}_{i,j}^-|$ within the same subset of \tilde{G} . Thus, it follows that $W^1(G, t) = \max_{i,j \in [n]; i < j} W_{i,j}^1(G, t)$, $W^2(G, t) = \max_{i,j \in [n]; i < j} W_{i,j}^2(G, t)$, and $LS_\Delta(G, t) = \max(W^1(G, t), W^2(G, t))$. It is important to note that we do not consider whether $a_{i,j}$ equals zero when computing $LS_\Delta(G, t)$. In the cases where $a_{i,j} = 0$, the number of modifications required to compute $W_{i,j}^2(G, t)$ is reduced from t to $t - 1$. However, our algorithm still executes t modifications to G , which does not compromise individual privacy, as $LS_\Delta(G, t)$ is a monotonically non-decreasing function of t . As per Algorithm 2, we conclude that $W_{i,j}^1(G, t)$ can be formulated as

$$W_{i,j}^1(G, t) = \begin{cases} w_{i,j}^s + t, & \text{if } t \leq b_{i,j}, \\ \min(w_{i,j}^s + \lfloor (t + b_{i,j})/2 \rfloor, n - 2), & \text{if } t > b_{i,j}, \end{cases} \quad (12)$$

where $w_{i,j}^s = w_{i,j}^+ + w_{i,j}^-$, and $W_{i,j}^2(G, t)$ can be expressed as

$$W_{i,j}^2(G, t) = \begin{cases} 2w_{i,j}^a + 4t, & \text{if } t \leq w_{i,j}^m, \\ 2w_{i,j}^a + 2w_{i,j}^m + 2t, & \text{if } w_{i,j}^m < t \leq w_{i,j}^m + b_{i,j}, \\ \min(2(w_{i,j}^a + w_{i,j}^m + \lfloor (t + w_{i,j}^m + b_{i,j})/2 \rfloor), 2n - 4), & \text{otherwise,} \end{cases} \quad (13)$$

where $w_{i,j}^a = |w_{i,j}^+ - w_{i,j}^-|$ and $w_{i,j}^m = \min(w_{i,j}^+, w_{i,j}^-)$. To streamline our analysis, we transform the triplet $(w_{i,j}^+, w_{i,j}^-, b_{i,j})$ into the pair $(w_{i,j}^s, b_{i,j})$ for the calculation of $W^1(G, t)$. Similarly, we convert it into a triplet $(w_{i,j}^a, w_{i,j}^m, b_{i,j})$ for the computation of $W^2(G, t)$. This conversion facilitates a more intuitive presentation of the methods involved in these calculations without additional overhead.

To compute $W^1(G, t)$, we first sort the $\Gamma + 1$ pairs $(w_{i,j}^s, b_{i,j})$ in a non-increasing order by $w_{i,j}^s$. If multiple pairs have the same value of $w_{i,j}^s$, only the pair with the maximum of $b_{i,j}$ is kept, denoted as Q_0 . We then process the list in the order of descending $w_{i,j}^s$. Each pair (w^s, b) is preserved only if, relative to the previously retained pair (\tilde{w}^s, \tilde{b}) , it satisfies $2(\tilde{w}^s - w) + \tilde{b} < b$. The set of pairs that meet this condition is represented as $\mathcal{L}^1 = \{(w_1^s, b_1), \dots, (w_{k_1}^s, b_{k_1})\}$. Equally pivotal in our analysis is the determination of breakpoints, utilized to select the appropriate data from the filtered list to compute $W^1(G, t)$ or $W^2(G, t)$. In what follows, we set the initial breakpoint as 0 and the final breakpoint as $2n - 3$. In the computation of $W^1(G, t)$, the other breakpoints are set to $t_i^1 = 2(w_i^s - w_{i+1}^s) + b_i$ for $i \in [k_1 - 1]$. The entire filtering process in computing $W^1(G, t)$ is shown in in Lines 8-12 of Algorithm 3. Then $W^1(G, t) = \min(w_i^s + \lfloor (t + \min(b_i, t))/2 \rfloor, n - 2)$ if $t \in [t_{i-1}^1, t_i^1)$.

The calculation of $W^2(G, t)$ is more complicated, primarily since the independent variable t must be compared with two different quantities: $w_{i,j}^m$ and $w_{i,j}^m + b_{i,j}$. To solve this problem, we maintain three distinct lists. First of all, we sort the $\Gamma + 1$ triplets $(w_{i,j}^a, w_{i,j}^m, b_{i,j})$

Algorithm 3: The Calculation of Smooth Sensitivity

Input: signed graph G represented as adjacency vectors $\{a_1, \dots, a_n\}$, the parameter β

Output: smooth sensitivity of f_Δ : $S_{\Delta, \beta}^*(G)$

```

1  $S_{\Delta, \beta}^*(G) \leftarrow 0$ ;  $b_{\max} \leftarrow 0$ ;  $Q \leftarrow \emptyset$ ;  $\mathcal{L}^c \leftarrow \emptyset$ ;  $t_0^c \leftarrow 0$  for  $c \in [4]$ ;
2  $\{(w_{i,j}^+, w_{i,j}^-)\}_{i,j \in [n]; i < j} \leftarrow$  Execute Algorithm 1;
3 for each node pair  $(v_i, v_j)$ , where  $i, j \in [n], i < j$  do
4    $b_{i,j} \leftarrow d_i + d_j - 2(w_{i,j}^+ + w_{i,j}^- + |a_{i,j}|)$ ;
5   if  $w_{i,j}^+ \neq 0$  or  $w_{i,j}^- \neq 0$  then  $Q.push((w_{i,j}^+, w_{i,j}^-, b_{i,j}))$ ;
6   else  $b_{\max} \leftarrow \max(b_{\max}, b_{i,j})$ ;
7  $Q.push((0, 0, b_{\max}))$ ;
   // Let  $w_{i,j}^s = w_{i,j}^+ + w_{i,j}^-$ ,  $w_{i,j}^a = |w_{i,j}^+ - w_{i,j}^-|$ ,  $w_{i,j}^m = \min(w_{i,j}^+, w_{i,j}^-)$ 
8 Sort  $Q$  in a non-increasing order by  $w_{i,j}^s$ . If multiple pairs have the
   same  $w_{i,j}^s$ , keep the one with the largest  $b_{i,j}$ , denoted as  $Q_0$ ;
9 for each pair  $(w_i^s, b_i) \in Q_0$  do
10  if  $2(w_{i-1}^s - w_i^s) + b_{i-1} < b_i$  then  $\mathcal{L}^1.push((w_i^s, b_i))$ ;
11  $k_1 \leftarrow \mathcal{L}^1.length()$ ;
12  $t_i^1 \leftarrow 2(w_i^s - w_{i+1}^s) + b_i$  for  $i \in [k_1 - 1]$ ;  $t_{k_1}^1 = 2n - 3$ ;
13 Sort  $Q$  in a non-increasing order by  $w_{i,j}^a$ . If multiple triplets have the
   same  $w_{i,j}^a$ , keep the one with the largest  $w_{i,j}^m$ , denoted as  $Q_1$ ;
14 for each triplet  $(w_i^a, w_i^m, b_i) \in Q_1$  do
15  if  $w_{i-1}^a - w_i^a + w_{i-1}^m < w_i^m$  then  $\mathcal{L}^2.push((w_i^a, w_i^m, b_i))$ ;
16  $k_2 \leftarrow \mathcal{L}^2.length()$ ;
17  $t_i^2 = w_i^a - w_{i+1}^a + w_i^m$  for  $i \in [k_2 - 1]$ ;  $t_{k_2}^2 = 2n - 3$ ;
18 Sort  $Q$  in a non-increasing order by  $w_{i,j}^a$ . If several triplets have the
   same  $w_{i,j}^a$ , keep the one with the largest  $3w_{i,j}^m + b_{i,j}$ , denoted as  $Q_2$ ;
19 for each triplet  $(w_i^a, w_i^m, b_i) \in Q_2$  do
20  if  $2(w_{i-1}^a - w_i^a) + 3w_{i-1}^m + b_{i-1} < 3w_i^m$  then
21     $\mathcal{L}^3.push((w_i^a, w_i^m, b_i))$ ;
22  $k_3 \leftarrow \mathcal{L}^3.length()$ ;
23  $t_i^3 = (2(w_i^a - w_{i+1}^a) + 3w_i^m + b_i)/3$  for  $i \in [k_3 - 1]$ ;  $t_{k_3}^3 = 2n - 3$ ;
24 Sort  $Q$  in a non-increasing order by  $w_{i,j}^a + w_{i,j}^m$ . If several triplets
   have the same  $w_{i,j}^a + w_{i,j}^m$ , keep the one with the largest  $w_{i,j}^m + b_{i,j}$ ,
   denoted as  $Q_3$ ;
25 for each triplet  $(w_i^a, w_i^m, b_i) \in Q_3$  do
26  if  $2(w_{i-1}^a - w_i^a) + 3w_{i-1}^m - 3w_i^m + b_{i-1} < b_i$  then
27     $\mathcal{L}^4.push((w_i^a, w_i^m, b_i))$ ;
28  $k_4 \leftarrow \mathcal{L}^4.length()$ ;
29  $t_i^4 = 2(w_i^a - w_{i+1}^a) + 3w_{i+1}^m - 2w_i^m + b_i$  for  $i \in [k_4 - 1]$ ;  $t_{k_4}^4 = 2n - 3$ ;
30 for  $t = 0$  to  $2n - 3$  do
31   Select  $(w_i^s, b_i)$  from  $\mathcal{L}^1$  where  $t \in [t_{i-1}^1, t_i^1)$ ;
32    $W^1(G, t) = \min(w_i^s + \lfloor (t + \min(b_i, t))/2 \rfloor, n - 2)$ ;
33   Select three triplets  $(w_i^{a,c}, w_i^{m,c}, b_i^c)$  from  $\mathcal{L}^c$  where
      $t \in [t_{i-1}^c, t_i^c)$  and  $c \in [2, 4]$ ;
34    $W^2(G, t) = \min(\max_{c \in [2,4]} 2w_i^{a,c} + 2\min(w_i^{m,c}, t) + 2\lfloor (t + \min(t, w_i^{m,c} + b_i^c))/2 \rfloor, 2n - 4)$ ;
35    $LS_\Delta(G, t) = \max(W^1(G, t), W^2(G, t))$ ;
36    $S_{\Delta, \beta}^*(G) = \max(S_{\Delta, \beta}^*(G), e^{-\beta t} LS_\Delta(G, t))$ ;
37 return  $S_{\Delta, \beta}^*(G)$ 

```

in a non-increasing order based on two different criteria: $w_{i,j}^a$ and $w_{i,j}^a + w_{i,j}^m$. For the first criterion, if several triplets share the same $w_{i,j}^a$ value, we retain only the triplet with the largest $w_{i,j}^m$. This list is represented as Q_1 . In addition, we also employ the first criterion to construct Q_2 but retain the triplet with the largest $3w_{i,j}^m + b_{i,j}$ in case

of identical $w_{i,j}^a$ value. Similarly, for the second criterion, if multiple triplets have the same value of $w_{i,j}^a + w_{i,j}^m$, we keep the triplet with the largest $w_{i,j}^m + b_{i,j}$, and this list is denoted as Q_3 .

Then, we process the sorted lists Q_1 , Q_2 , and Q_3 , respectively. The first treatment involves a traversal of the list Q_1 in the order of decreasing $w_{i,j}^a$. For each triplet (w^a, w^m, b) , it is preserved only if the condition $\bar{w}^a - w^a + \bar{w}^m < w^m$ holds, where $(\bar{w}^a, \bar{w}^m, \bar{b})$ is the previous retained triplet. The collection of triplets that meet this criterion is denoted as $\mathcal{L}^2 = \{(w_1^a, w_1^m, b_1), \dots, (w_{k_2}^a, w_{k_2}^m, b_{k_2})\}$. The breakpoints are calculated by the formula $t_i^2 = w_i^a - w_{i+1}^a + w_i^m$ for $i \in [k_2 - 1]$.

Go through the list Q_2 in the order of decreasing $w_{i,j}^a$. A triplet (w^a, w^m, b) is retained only if it meets the criterion $2(\bar{w}^a - w^a) + 3\bar{w}^m + \bar{b} < 3w^m$, when compared with the previously preserved triplet $(\bar{w}^a, \bar{w}^m, \bar{b})$. The triplets that satisfy this condition form the list $\mathcal{L}^3 = \{(w_1^a, w_1^m, b_1), \dots, (w_{k_3}^a, w_{k_3}^m, b_{k_3})\}$. To determine the breakpoints for \mathcal{L}^3 , we set $t_i^3 = (2(w_i^a - w_{i+1}^a) + 3w_i^m + b_i)/3$ for $i \in [k_3 - 1]$.

For the sorted triplet list Q_3 , we traverse this list in the order of decreasing $w_{i,j}^a + w_{i,j}^m$. For each (w^a, w^m, b) , keep this triplet only if $2(\bar{w}^a - w^a) + 3\bar{w}^m - 3w^m + \bar{b} < b$ relative to the previously retained triplet $(\bar{w}^a, \bar{w}^m, \bar{b})$. The final set of retained triplets is represented as $\mathcal{L}^4 = \{(w_1^a, w_1^m, b_1), \dots, (w_{k_4}^a, w_{k_4}^m, b_{k_4})\}$. Let $t_i^4 = 2(w_i^a - w_{i+1}^a) + 3w_i^m - 2w_{i+1}^m + b_i$ for $i \in [k_4 - 1]$. The comprehensive filtering procedure in the computation of $W^2(G, t)$ is delineated in Lines 13-29 of Algorithm 3.

To finalize the computation of $W^2(G, t)$, it is essential to choose the relevant triplets from the three filtered lists. Given a specific distance t , similar to computing $W^1(G, t)$, we select three triplets from \mathcal{L}^2 , \mathcal{L}^3 , and \mathcal{L}^4 respectively, based on the comparison of t with their respective breakpoint sequences, denoted as $(w_i^{a,c}, w_i^{m,c}, b_i^c)$, where $c \in \{2, 3, 4\}$. Then it follows that $W^2(G, t) = \min(\max_{c \in \{2,3,4\}} 2w_i^{a,c} + 2 \min(w_i^{m,c}, t) + 2 \lfloor (t + \min(t, w_i^{m,c} + b_i^c))/2 \rfloor, 2n - 4)$.

Recall that $LS_\Delta(G, t) = \max(W^1(G, t), W^2(G, t))$, we compute $e^{-\beta t} LS_\Delta(G, t)$ at each distance t sequentially and select the maximum as the smooth sensitivity. The pseudo-code for calculating the smooth sensitivity of f_Δ is presented in Algorithm 3.

In addition, we derive a theorem that delineates an explicit condition under which the maximum between $W^1(G)$ and $W^2(G)$ equals to the smooth sensitivity, where $W^1(G) = \max_{i,j \in [n]; i < j} w_{i,j}^s$ and $W^2(G) = \max_{i,j \in [n]; i < j} 2w_{i,j}^a$. Under this condition, the randomized algorithm \mathcal{A} injects noise proportional to $\max(W^1(G), W^2(G))$, without the necessity to calculate the local sensitivity at distance t .

THEOREM 3.2. *Given a signed graph G , if $W^1(G) \geq \frac{1}{\beta}$ and $W^2(G) \geq \frac{4}{\beta}$, then $S_{\Delta, \beta}^*(G) = \max(W^1(G), W^2(G))$.*

PROOF. Let $U^1(G, t) = W^1(G) + t$ and $U^2(G, t) = W^2(G) + 4t$. Then $U^1(G, t) \geq W^1(G, t)$ and $U^2(G, t) \geq W^2(G, t)$. Incorporating these inequalities into the smooth sensitivity calculation, we derive

$$\begin{aligned} S_{\Delta, \beta}^*(G) &= \max_{t \in [0, 2n-3]} e^{-\beta t} LS_\Delta(G, t) \\ &= \max_{t \in [0, 2n-3]} \{e^{-\beta t} W^1(G, t), e^{-\beta t} W^2(G, t)\} \\ &\leq \max_{t \in [0, 2n-3]} \{e^{-\beta t} U^1(G, t), e^{-\beta t} U^2(G, t)\} \\ &= \max_{t \in [0, 2n-3]} \{e^{-\beta t} (W^1(G) + t), e^{-\beta t} (W^2(G) + 4t)\}. \end{aligned} \quad (14)$$

Consider the functions $h(t) = e^{-\beta t} \cdot (W^1(G) + t)$ and $g(t) = e^{-\beta t} \cdot (W^2(G) + 4t)$. The objective is to find their maximum values over the continuous interval $[0, 2n - 3]$. For the function $h(t)$, its derivative $h'(t) = e^{-\beta t} (1 - \beta(W^1(G) + t))$ indicates that $h(t)$ is strictly concave, with a unique maximum at $t = \frac{1}{\beta} - W^1(G)$. Notably, when $W^1(G) \geq \frac{1}{\beta}$, $h(t)$ exhibits a monotonically decreasing trend over the interval $[0, \infty)$. Therefore, in the case where $W^1(G) \geq \frac{1}{\beta}$, the maximum of $h(t)$ is attained at $t = 0$ and equals $W^1(G)$. A similar analysis of $g(t)$ indicates that, under the condition $W^2(G) \geq \frac{4}{\beta}$, its maximum also occurs at $t = 0$ and is equal to $W^2(G)$. As a result, if $W^1(G) \geq \frac{1}{\beta}$ and $W^2(G) \geq \frac{4}{\beta}$, it follows that $S_{\Delta, \beta}^*(G) = \max(W^1(G), W^2(G))$. \square

The time complexity of smooth sensitivity calculation is analyzed methodically herein. In the initial phase of data preparation, the time complexity is delineated as $O(m \cdot d_{\max} + n^2)$. This complexity arises from the time required to compute $w_{i,j}^+$ and $w_{i,j}^-$, which is $O(m \cdot d_{\max})$, and the time needed to calculate $b_{i,j}$, quantified as $O(n^2)$. In the phase dedicated to the elimination of duplicate and irrelevant data, the complexity remains at $O(\Gamma)$, because both the creation of sorted lists via Bucket Sort and the generation of filtered lists are achievable within $O(\Gamma)$. The computation of smooth sensitivity in the final phase requires $O(n)$ time. Consequently, the overall time complexity of our proposed method stands at $O(m \cdot d_{\max} + n^2)$.

Note that the time complexity analysis of the smooth sensitivity computation focuses on the worst-case scenario. As shown in our experiments, due to the sparsity of real-world signed networks, our approach based on smooth sensitivity can also handle signed graphs with 1 million nodes and 30 million edges.

Smooth Upper Bound on LS_Δ . For large signed graphs, computing the smooth sensitivity of the triangle count f_Δ is non-trivial due to the excessive time complexity. To circumvent this problem, we relax the requirements for the computation of smooth sensitivity and build a computationally efficient function that computes a smooth upper bound on local sensitivity of f_Δ . Given an arbitrary database x and a query function f , LS_f may have multiple smooth bounds, and the following proposition provides a method to determine the smooth upper bounds on local sensitivity.

PROPOSITION 3.3. [26] *Define $S_{f, \beta}(x) = \max_{t \in [0, n]} e^{-\beta t} U(x, t)$, where $U(x, t)$ satisfies*

- (1) $LS_f(x) \leq U(x, 0)$ for all x .
- (2) $U(x, t) \leq U(\tilde{x}, t + 1)$ for all x, \tilde{x} such that $d(x, \tilde{x}) = 1$.

Then $S_{f, \beta}(x)$ is a β -smooth upper bound on local sensitivity.

Inspired by the Theorem 3.2, we develop a function that adheres to the criterion for a smooth upper bound on LS_Δ and is also computationally efficient.

THEOREM 3.4. *Define $U(G, t) = \max(U^1(G, t), U^2(G, t))$, where $U^1(G, t) = W^1(G) + t$ and $U^2(G, t) = W^2(G) + 4t$. Then the function $S_{\Delta, \beta}(G) = \max_{t \in [0, 2n-3]} e^{-\beta t} U(G, t)$ is a β -smooth upper bound on local sensitivity of f_Δ .*

PROOF. Initially, we prove that $U(G, t)$ fulfills the first condition of Proposition 3.3. Consider the scenario where $t = 0$. It follows that $U^1(G, 0) = W^1(G)$ and $U^2(G, 0) = W^2(G)$. Recall that $W^1(G) = \max_{i,j \in [n]; i < j} (w_{i,j}^+ + w_{i,j}^-)$ and $W^2(G) = \max_{i,j \in [n]; i < j} 2|w_{i,j}^+ - w_{i,j}^-|$,

Algorithm 4: Smooth Upper Bound on Local Sensitivity

Input: signed graph G represented as adjacency vectors $\{a_1, \dots, a_n\}$, the parameter β
Output: smooth upper bound on local sensitivity of f_Δ : $S_{\Delta,\beta}(G)$

```

1  $W^1(G) \leftarrow 0; W^2(G) \leftarrow 0;$ 
2  $\{(w_{i,j}^+, w_{i,j}^-)\}_{i,j \in [n]; i < j} \leftarrow \text{Execute Algorithm 1};$ 
3 for each wedge pair  $(w_{i,j}^+, w_{i,j}^-)$ , where  $w_{i,j}^+ \neq 0$  or  $w_{i,j}^- \neq 0$  do
4    $W^1(G) \leftarrow \max(W^1(G), w_{i,j}^+ + w_{i,j}^-);$ 
5    $W^2(G) \leftarrow \max(W^2(G), 2|w_{i,j}^+ - w_{i,j}^-|);$ 
6  $t_1 = \frac{1}{\beta} - W^1(G); t_2 = \frac{1}{\beta} - W^2(G)/4;$ 
7 if  $t_1 \leq 0$  then  $S_{\Delta,\beta}(G) \leftarrow W^1(G);$ 
8 else
9    $S_{\Delta,\beta}(G) \leftarrow e^{-\beta \lceil t_1 \rceil} (W^1(G) + \lceil t_1 \rceil);$ 
10   $S_{\Delta,\beta}(G) \leftarrow \max(S_{\Delta,\beta}(G), e^{-\beta \lfloor t_1 \rfloor} (W^1(G) + \lfloor t_1 \rfloor));$ 
11 if  $t_2 \leq 0$  then  $S_{\Delta,\beta}(G) \leftarrow \max(S_{\Delta,\beta}(G), W^2(G));$ 
12 else
13   $S_{\Delta,\beta}(G) \leftarrow \max(S_{\Delta,\beta}(G), e^{-\beta \lceil t_2 \rceil} (W^2(G) + 4 \lceil t_2 \rceil));$ 
14   $S_{\Delta,\beta}(G) \leftarrow \max(S_{\Delta,\beta}(G), e^{-\beta \lfloor t_2 \rfloor} (W^2(G) + 4 \lfloor t_2 \rfloor));$ 
15 return  $S_{\Delta,\beta}(G)$ 

```

it is easily verified that $LS_\Delta(G) \leq \max(W^1(G), W^2(G))$. Therefore, we ascertain that $LS_\Delta(G) \leq U(G, 0)$.

Next, we demonstrate that $U(G, t)$ satisfies the second condition. It is notable that a single modification to G results in, at most, 1 and 4 changes in $W^1(G)$ and $W^2(G)$, respectively. Consequently, for all G and \tilde{G} such that $d(G, \tilde{G}) = 1$, it is inferred that $U^1(G, t) \leq U^1(\tilde{G}, t+1)$ and $U^2(G, t) \leq U^2(\tilde{G}, t+1)$. This deduction leads to the conclusion that $U(G, t) \leq U(\tilde{G}, t+1)$. \square

In practice, it is feasible to compute $S_{\Delta,\beta}(G)$ efficiently. And the function $S_{\Delta,\beta}(G)$ can also be expressed as

$$S_{\Delta,\beta}(G) = \max_{t \in [0, 2n-3]} \{e^{-\beta t} (W^1(G) + t), e^{-\beta t} (W^2(G) + 4t)\}. \quad (15)$$

As elucidated in the proof of Theorem 3.2, the peak values of $h(t) = e^{-\beta t} (W^1(G) + t)$ and $g(t) = e^{-\beta t} (W^2(G) + 4t)$ are achieved at $t = \frac{1}{\beta} - W^1(G)$ and $t = \frac{1}{\beta} - W^2(G)/4$, respectively. Given the discrete constraints of t and its bounded domain, $S_{\Delta,\beta}(G)$ can be efficiently computed in constant time, $\mathcal{O}(1)$, based on the values of $W^1(G)$ and $W^2(G)$, as outlined in Lines 6-14 of Algorithm 4. The values of $W^1(G)$ and $W^2(G)$ can be computed in time $\mathcal{O}(m \cdot d_{\max})$, as stated in Lines 2-5. Algorithm 4 describes the process of the computation of the smooth upper bound on local sensitivity of f_Δ , and the worst-case time complexity of Algorithm 4 is $\mathcal{O}(m \cdot d_{\max})$. Empirical evidence from our experiments demonstrates that our method based on the smooth upper bound on local sensitivity is more efficient than the smooth-sensitivity-based method, especially for large signed graphs.

Privacy Guarantee. The algorithms for calculating smooth upper bounds on local sensitivity of f_Δ can be used in conjunction with Theorem 2.8 to obtain efficient differentially private algorithms for estimating balanced and unbalanced triangle counts.

THEOREM 3.5. *Given $\epsilon > 0$ and $\delta > 0$, let $\beta = \frac{\epsilon}{8+4 \ln(2/\delta)}$. The randomized algorithm $\mathcal{A}(G) = f_\Delta(G) + (2S_{\Delta,\beta}(G)/\epsilon) \cdot Z$ satisfies (ϵ, δ) -signed edge centralized differential privacy where Z is sampled from 2-dimensional Laplace distribution.*

4 TRIANGLE COUNTING IN LOCAL MODEL

Local differential privacy has been extensively accepted and adopted in academic research and industry applications due to its robust framework for privacy preservation. Unlike centralized DP, LDP provides enhanced privacy protection because data obfuscation is performed at the node level rather than by a data curator. In this section, we explore the techniques employed to compute the number of balanced and unbalanced triangles in a signed graph under LDP. Inspired by the work of Imola et al. [14], we propose a two-phase framework tailored for balanced and unbalanced triangle counting. The first phase uses the Generalized Randomized Response mechanism to perturb the adjacency vectors. Subsequently, in the second phase, we propose an innovative response mechanism. This mechanism injects noise into the response of each node about the triangle count query, which is calibrated to a smooth upper bound on local sensitivity. This two-phase approach achieves a better trade-off between data utility and privacy preservation.

The Two-Phase Framework. In the local model, each node $v_i \in \mathcal{V}$ independently maintains its own data, represented as an adjacency vector a_i . This vector contains information about the connections of node v_i and the signs of these connections. However, when the query function is f_Δ , nodes cannot locally calculate and submit the obfuscated counts of balanced and unbalanced triangles due to their limited insight into the complete graph structure. Specifically, node v_i is inherently unable to perceive any triangle (v_i, v_j, v_k) since it lacks information about the existence of the edge (v_j, v_k) and its sign in the signed graph G .

To address this inadequacy, we propose a two-phase framework to privately response to the query function f_Δ . In the first phase, each node v_i independently employs the GRR mechanism to perturb the elements $a_{i,1}, \dots, a_{i,i-1}$ in its own adjacency vector. These elements correspond to the connections between the node v_i and nodes with smaller IDs. Given the allocated privacy budget ϵ_1 for this phase, the perturbation scheme of the GRR mechanism is as follows:

$$\Pr[\hat{a}_{i,j} | a_{i,j}] = \begin{cases} \frac{e^{\epsilon_1}}{e^{\epsilon_1} + 2}, & \text{if } \hat{a}_{i,j} = a_{i,j}, \\ \frac{1}{e^{\epsilon_1} + 2}, & \text{if } \hat{a}_{i,j} \neq a_{i,j}. \end{cases} \quad (16)$$

The data curator then collects the distorted data from all nodes and constructs a noisy signed graph, denoted as \hat{G} , under the premise of an undirected structure. In fact, the curator can compute the number of balanced and unbalanced triangles directly from \hat{G} . However, this estimation method infuses substantial noise into the statistical query results since each edge and its associated sign are modified with a certain probability. This means that the state of any triangle in \hat{G} is influenced by three independent random variables. As a matter of fact, for any triangle that involves a certain node, only one edge and its associated sign are unknown to that node. If the data curator publishes the noisy signed graph \hat{G} , nodes can obtain information about the edges they are unaware of from \hat{G} . Hence, we can introduce an additional round of interaction between nodes and the data curator to reduce noise injection.

In the second phase, each node v_i locally computes the number of noisy triangles formed by (v_i, v_j, v_k) where the information of edge (v_j, v_k) can be obtained from \hat{G} . Here, a noisy triangle is identified as balanced if $a_{i,j} \cdot a_{i,k} \cdot \hat{a}_{j,k}$ equals 1, and as unbalanced if it equals -1 . This approach ensures that only one edge in each noisy triangle

is obfuscated, which enhances the overall utility, albeit at the cost of increased communication overhead per node. However, the direct release of these noisy balanced and unbalanced triangle counts leads to two primary issues. The first issue is the introduction of statistical bias into the estimates of these triangle counts. This bias arises from the noise added during the adjacency vector obfuscation process, which could skew the final estimates. Secondly, and perhaps more importantly, the direct release of such data potentially jeopardizes the privacy of the edges and their respective signs in G . This concern stems from the fact that each node still employs its own real data when calculating balanced and unbalanced triangle counts.

To address the first problem, we utilize an empirical estimation technique [14, 15, 35] to derive an unbiased estimate of $f_{\Delta}(G)$ from these noisy counts. Specifically, let T_i^b and T_i^u respectively represent the number of noisy balanced and unbalanced triangles computed by node v_i , where the IDs of the three nodes in each triangle satisfy the condition $i > j > k$. In addition, node v_i is also required to calculate the number of 2-stars centered on itself under the same constraints of node IDs, denoted as s_i . The data curator then estimates the number of balanced and unbalanced triangles in G , denoted as \bar{T}^b and \bar{T}^u respectively, based on the data collected from each node. Let q be set to $1/(e^{\epsilon_1} + 2)$ and these estimates are calculated as follows:

$$\bar{T}^b = \frac{1}{1-3q} \sum_{i=1}^n (T_i^b - q \cdot s_i), \quad (17)$$

$$\bar{T}^u = \frac{1}{1-3q} \sum_{i=1}^n (T_i^u - q \cdot s_i). \quad (18)$$

LEMMA 4.1. *The computed values \bar{T}^b and \bar{T}^u as delineated in Equations (17) and (18), serve as unbiased estimates of the true counts of balanced and unbalanced triangles in G , respectively, i.e., $\mathbb{E}[\bar{T}^b] = T^b$ and $\mathbb{E}[\bar{T}^u] = T^u$.*

PROOF. First, we prove that $\mathbb{E}[\bar{T}^b] = T^b$. For each node v_i , T_i^b denotes the number of balanced triangles (v_i, v_j, v_k) computed by v_i , where $i > j > k$ and $a_{i,j} \cdot a_{i,k} \cdot \hat{a}_{j,k} = 1$. The random variable in each triangle is $\hat{a}_{j,k}$. Thus, we have

$$\mathbb{E}[T_i^b] = \sum_{\substack{i>j>k \\ a_{i,j} \cdot a_{i,k} = 1}} \mathbb{E}[\mathbb{I}(\hat{a}_{j,k} = 1)] + \sum_{\substack{i>j>k \\ a_{i,j} \cdot a_{i,k} = -1}} \mathbb{E}[\mathbb{I}(\hat{a}_{j,k} = -1)], \quad (19)$$

where $\mathbb{I}(\cdot)$ represents the indicator function. Under the perturbation schema of GRR mechanism, it follows that $\mathbb{E}[\mathbb{I}(\hat{a}_{j,k} = 1)] = q(1 + (e^{\epsilon_1} - 1)\mathbb{I}(a_{j,k} = 1))$, which simplifies to $q \cdot e^{\epsilon_1}$ if $a_{j,k} = 1$ and q otherwise. Likewise, $\mathbb{E}[\mathbb{I}(\hat{a}_{j,k} = -1)] = q(1 + (e^{\epsilon_1} - 1)\mathbb{I}(a_{j,k} = -1))$. Then the expectation of \bar{T}^b can be expressed as

$$\begin{aligned} \mathbb{E}[\bar{T}^b] &= \mathbb{E}\left[\frac{1}{1-3q} \sum_{i=1}^n (T_i^b - q \cdot s_i)\right] \\ &= \frac{1}{1-3q} \sum_{i=1}^n (\mathbb{E}[T_i^b] - \sum_{\substack{i>j>k \\ a_{i,j} \cdot a_{i,k} = 1}} q - \sum_{\substack{i>j>k \\ a_{i,j} \cdot a_{i,k} = -1}} q) \\ &= \sum_{i=1}^n \left(\sum_{\substack{i>j>k \\ a_{i,j} \cdot a_{i,k} = 1}} \mathbb{I}(a_{j,k} = 1) + \sum_{\substack{i>j>k \\ a_{i,j} \cdot a_{i,k} = -1}} \mathbb{I}(a_{j,k} = -1) \right) \\ &= T^b. \end{aligned} \quad (20)$$

The expectation analysis of unbalanced triangle counts is similar to that of balanced triangle counts. For each node $v_i \in \mathcal{V}$, the expected

value of T_i^u can be formulated as follows:

$$\mathbb{E}[T_i^u] = \sum_{\substack{i>j>k \\ a_{i,j} \cdot a_{i,k} = -1}} \mathbb{E}[\mathbb{I}(\hat{a}_{j,k} = 1)] + \sum_{\substack{i>j>k \\ a_{i,j} \cdot a_{i,k} = 1}} \mathbb{E}[\mathbb{I}(\hat{a}_{j,k} = -1)]. \quad (21)$$

Consequently, we ascertain that:

$$\begin{aligned} \mathbb{E}[\bar{T}^u] &= \mathbb{E}\left[\frac{1}{1-3q} \sum_{i=1}^n (T_i^u - q \cdot s_i)\right] \\ &= \frac{1}{1-3q} \sum_{i=1}^n (\mathbb{E}[T_i^u] - \sum_{\substack{i>j>k \\ a_{i,j} \cdot a_{i,k} = 1}} q - \sum_{\substack{i>j>k \\ a_{i,j} \cdot a_{i,k} = -1}} q) \\ &= \sum_{i=1}^n \left(\sum_{\substack{i>j>k \\ a_{i,j} \cdot a_{i,k} = -1}} \mathbb{I}(a_{j,k} = 1) + \sum_{\substack{i>j>k \\ a_{i,j} \cdot a_{i,k} = 1}} \mathbb{I}(a_{j,k} = -1) \right) \\ &= T^u. \end{aligned} \quad (22)$$

Hence, \bar{T}^u offers an unbiased estimation of the actual unbalanced triangle counts. \square

To address the privacy concerns, each node needs to perturb its computed statistics via an LDP mechanism before sending them to the data curator. A simple implementation is to utilize the Laplace mechanism for data perturbation. However, this approach is less practical due to the high global sensitivity, quantified as $2(n-2)$. To circumvent this issue, Imola et al. [14] employ graph projection to reduce sensitivity. This approach removes some neighbors from the neighbor list so that the maximum degree d_{\max} is bounded by a predetermined threshold \hat{d}_{\max} . As a result, the global sensitivity is reduced to $2(\hat{d}_{\max} - 1)$. However, there are two main limitations of this approach. The first limitation is that the determination of the threshold \hat{d}_{\max} requires an additional privacy budget. In addition, since real-world graphs obey power-law degree distributions [16, 23], the direct addition of Laplace noise proportional to $2(\hat{d}_{\max} - 1)$ would substantially diminish the utility of the data. In order to circumvent these limitations, we propose an innovative approach that employs a smooth upper bound on local sensitivity to calibrate the noise.

Due to the presence of empirical estimation, the release statistics for each node v_i are adjusted to $(T_i^b - qs_i, T_i^u - qs_i)$. Consequently, we need to devise a function that can efficiently calculate the smooth upper bound on local sensitivity for such statistical estimates. For node v_i , the local sensitivity of these estimates can be expressed as

$$LS_{\Delta}(\mathbf{a}_i) = \max_{\tilde{\mathbf{a}}_i: d(\mathbf{a}_i, \tilde{\mathbf{a}}_i) \leq 1} |\tilde{T}_i^b - T_i^b - q(\tilde{s}_i - s_i)| + |\tilde{T}_i^u - T_i^u - q(\tilde{s}_i - s_i)|. \quad (23)$$

Let d'_i represent the number of nodes that are connected to v_i and have smaller IDs than v_i . A specific smooth upper bound on local sensitivity is described in the subsequent theorem.

THEOREM 4.2. *For any node v_i , let $U(\mathbf{a}_i, t) = \max(d'_i + t, 2(d'_i + t - 1))$. The function $S_{\Delta, \beta}(\mathbf{a}_i) = \max_{t \in [0, i-1-d'_i]} e^{-\beta t} U(\mathbf{a}_i, t)$ serves as a β -smooth upper bound on local sensitivity for node v_i .*

PROOF. For any node v_i , the statistics T_i^b , T_i^u , and s_i are computed under the premise that node IDs satisfy the condition $i > j > k$. Thus, in the subsequent proof, we restrict our focus to modifications of the elements from the 1-st to the $(i-1)$ -th in \mathbf{a}_i . Moreover, because the maximum of d'_i is $i-1$, the number of modifications is restricted to $i-1-d'_i$. First of all, we prove that $U(\mathbf{a}_i, 0) \geq LS_{\Delta}(\mathbf{a}_i)$. In the scenario where the neighbor $\tilde{\mathbf{a}}_i$ is obtained by the addition of a signed edge connected to v_i , s_i and \tilde{s}_i can be expressed as $s_i = \binom{d'_i}{2}$

Algorithm 5: The Two-Phase Framework

Input: signed graph G represented as adjacency vectors $\{\mathbf{a}_1, \dots, \mathbf{a}_n\}$, the privacy budget ϵ_1, ϵ_2 , the invalidation probability δ

Output: private estimates for balanced triangles \hat{T}^b and unbalanced triangles \hat{T}^u

```

1 Initialize  $q \leftarrow \frac{1}{e^{\epsilon_1+2}}, \beta \leftarrow \frac{\epsilon_2}{8+4\ln(2/\delta)}$ ;
  /* First phase                                     */
  // Node side
2 for each node  $v_i \in \mathcal{V}$  do
3    $\hat{\mathbf{a}}_i \leftarrow (\text{GRR}_{\epsilon_1}(\mathbf{a}_{i,1}), \dots, \text{GRR}_{\epsilon_1}(\mathbf{a}_{i,i-1}))$ ;
4   Send  $\hat{\mathbf{a}}_i = \{\hat{a}_{i,1}, \dots, \hat{a}_{i,i-1}\}$  to the server;
  // Server side
5 Construct a noisy signed graph  $\hat{G}$  based on  $\{\hat{\mathbf{a}}_1, \dots, \hat{\mathbf{a}}_n\}$ ;
  /* Second phase                                     */
  // Node side
6 for each node  $v_i \in \mathcal{V}$  do
7    $T_i^b \leftarrow |\{(v_i, v_j, v_k) : i > j > k, a_{i,j} \cdot a_{i,k} \cdot \hat{a}_{j,k} = 1\}|$ ;
8    $T_i^u \leftarrow |\{(v_i, v_j, v_k) : i > j > k, a_{i,j} \cdot a_{i,k} \cdot \hat{a}_{j,k} = -1\}|$ ;
9    $s_i \leftarrow |\{(v_i, v_j, v_k) : i > j > k, a_{i,j} \cdot a_{i,k} \neq 0\}|$ ;
10   $\hat{T}_i^b \leftarrow (T_i^b - q \cdot s_i) + \text{Lap}(2S_{\Delta,\beta}(\mathbf{a}_i)/\epsilon_2)$ ;
11   $\hat{T}_i^u \leftarrow (T_i^u - q \cdot s_i) + \text{Lap}(2S_{\Delta,\beta}(\mathbf{a}_i)/\epsilon_2)$ ;
12  Send the perturbed data  $\hat{T}_i^b, \hat{T}_i^u$  to the server;
  // Server side
13 [s]  $\hat{T}^b \leftarrow \frac{1}{1-3q} \sum_{i=1}^n \hat{T}_i^b; \hat{T}^u \leftarrow \frac{1}{1-3q} \sum_{i=1}^n \hat{T}_i^u$ ;
14 return  $\hat{T}^b, \hat{T}^u$ 

```

and $\tilde{s}_i = \binom{\tilde{d}'_i}{2} = \binom{d'_i+1}{2}$, respectively. Then we deduce that $\tilde{s}_i - s_i = d'_i$. In addition, since we consider an additional constraint that $\hat{a}_{j,k} = 1$ or $\hat{a}_{j,k} = -1$ for the computation of T_i^b and T_i^u , it follows that $(\hat{T}_i^b + \hat{T}_i^u) - (T_i^b + T_i^u) \leq \tilde{s}_i - s_i$. Given that $q = 1/(e^{\epsilon_1} + 2)$, it is easily verified that $|\hat{T}_i^b - T_i^b - q(\tilde{s}_i - s_i)| + |\hat{T}_i^u - T_i^u - q(\tilde{s}_i - s_i)| \leq d'_i$. If $\tilde{\mathbf{a}}_i$ is obtained by the deletion of a signed edge attached to v_i , then $s_i - \tilde{s}_i = d'_i - 1$. This leads to $|\hat{T}_i^b - T_i^b - q(\tilde{s}_i - s_i)| + |\hat{T}_i^u - T_i^u - q(\tilde{s}_i - s_i)| \leq d'_i - 1$. In the case where $\tilde{\mathbf{a}}_i$ is obtained by the alteration of the sign of an edge linked to node v_i , s_i and \tilde{s}_i remain equal. From this, we derive that $|\hat{T}_i^b - T_i^b - q(\tilde{s}_i - s_i)| + |\hat{T}_i^u - T_i^u - q(\tilde{s}_i - s_i)| \leq 2(d'_i - 1)$. Based on the above derivation, we can conclude that $U(\mathbf{a}_i, 0) \geq LS_{\Delta}(\mathbf{a}_i)$.

Given that any one modification to \mathbf{a}_i results in a variation of d'_i by at most 1, we deduce the inequality $U(\mathbf{a}_i, t) \leq U(\tilde{\mathbf{a}}_i, t + 1)$ holds for all pairs of \mathbf{a}_i , and $\tilde{\mathbf{a}}_i$, provided that $d(\mathbf{a}_i, \tilde{\mathbf{a}}_i) \leq 1$. \square

For each node $v_i \in \mathcal{V}$, the smooth upper bound on local sensitivity $S_{\Delta,\beta}(\mathbf{a}_i)$ can be expressed as

$$S_{\Delta,\beta}(\mathbf{a}_i) = \max_{t \in [0, d'_i - 1]} \{e^{-\beta t}(d'_i + t), 2e^{-\beta t}(d'_i + t - 1)\}. \quad (24)$$

The calculation of $S_{\Delta,\beta}(\mathbf{a}_i)$ is similar to that of the smooth upper bound on local sensitivity under centralized DP. For any number d'_i , the maximum values of the functions $h(t) = e^{-\beta t}(d'_i + t)$ and $g(t) = 2e^{-\beta t}(d'_i + t - 1)$ are achieved at $t = \frac{1}{\beta} - d'_i$ and $t = \frac{1}{\beta} - d'_i + 1$, respectively. Given the discrete constraints and the specified interval for t , the value of $S_{\Delta,\beta}(\mathbf{a}_i)$ can be determined in constant time.

Algorithm 5 summarizes the overall protocol of our two-phase framework. It takes a signed graph G , the privacy budgets ϵ_1 and ϵ_2 for the first and second phases, and an invalidation probability δ as inputs. In the first phase, each node v_i applies GRR_{ϵ_1} to the elements $\mathbf{a}_{i,1}, \dots, \mathbf{a}_{i,i-1}$ for smaller node IDs in his/her adjacency vector \mathbf{a}_i . Then node v_i submits the obfuscated data $\hat{\mathbf{a}}_i = \{\hat{a}_{i,1}, \dots, \hat{a}_{i,i-1}\} \in$

Table 1: Dataset Statistics

DATASET	NODES	EDGES	T^b	T^u	AVG. DEG.
WikiElections	7,115	100,693	458,597	148,682	28.30
Epinions	131,580	711,210	4,368,206	541,870	10.81
WikiPolitics	138,587	715,883	2,659,365	318,661	10.33
Youtube	1,134,890	2,987,624	1,626,212	1,430,174	5.27
Pokec	1,632,803	30,622,564	17,321,674	15,235,784	37.51

$\{0, +1, -1\}^{i-1}$ to the server. Finally, the server constructs a noisy signed graph \hat{G} based on the data collected from all nodes.

In the second phase, each node v_i initially computes the statistics T_i^b , T_i^u and s_i under the assumption that the node IDs follow the order $i > j > k$. Subsequently, each node v_i introduces the Laplace noise $\text{Lap}(2S_{\Delta,\beta}(\mathbf{a}_i)/\epsilon_2)$ to both $T_i^b - qs_i$ and $T_i^u - qs_i$ and sends the perturbed statistics \hat{T}_i^b and \hat{T}_i^u to the server. The server then releases the unbiased estimates \hat{T}^b and \hat{T}^u , formulated as $\hat{T}^b = \frac{1}{1-3q} \sum_{i=1}^n \hat{T}_i^b$ and $\hat{T}^u = \frac{1}{1-3q} \sum_{i=1}^n \hat{T}_i^u$, respectively.

Privacy Guarantee. The subsequent theorem formalizes the privacy guarantees provided by the two-phase framework.

THEOREM 4.3. *Algorithm 5 satisfies $(\epsilon_1 + \epsilon_2, \delta)$ -signed edge local differential privacy.*

PROOF. In the first phase, since the GRR mechanism is employed to perturb the elements of adjacency vectors, it ensures ϵ_1 -signed edge LDP. In the second phase, each node v_i injects Laplace noise calibrated to the smooth upper bound on local sensitivity $S_{\Delta,\beta}(\mathbf{a}_i)$ into the statistical query response. This step provides (ϵ_2, δ) -signed edge LDP. In accordance with the sequential composition property of differential privacy, as delineated in Proposition 2.9, the two-phase framework satisfies $(\epsilon_1 + \epsilon_2, \delta)$ -signed edge LDP. \square

5 EXPERIMENTS

In this section, we conduct extensive experiments to evaluate the privacy-utility performance of the proposed approaches under both centralized and local DP.

5.1 Experimental Setup

Datasets. In our experiments, we make use of five real-world datasets. The important statistics of these datasets are summarized in Table 1. All datasets are publicly available. WikiElections and WikiPolitics can be obtained from the Koblenz network collection². The rest of the datasets are available from the Stanford network dataset collection³. WikiElections, Epinions, and WikiPolitics are all real-world signed graphs. WikiElections and WikiPolitics contain links between users from English Wikipedia. WikiElections is the network of users that voted for and against each other in admin elections. WikiPolitics contains interpreted interactions between users who have edited pages about politics. Each of these interactions is labeled as either positive or negative. Epinions is who-trusts-whom networks that consist of individual users interconnected by trust and distrust links. In addition, Youtube and Pokec are social networks utilized herein to assess the scalability of our proposed algorithms. In accordance with

²<http://konect.cc/networks/>

³<http://snap.stanford.edu/data/>

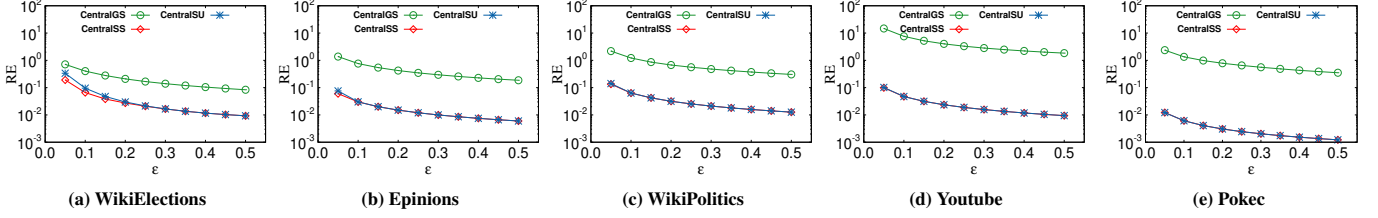


Figure 2: Trade-offs between privacy and relative error (RE) of various mechanisms under centralized DP.

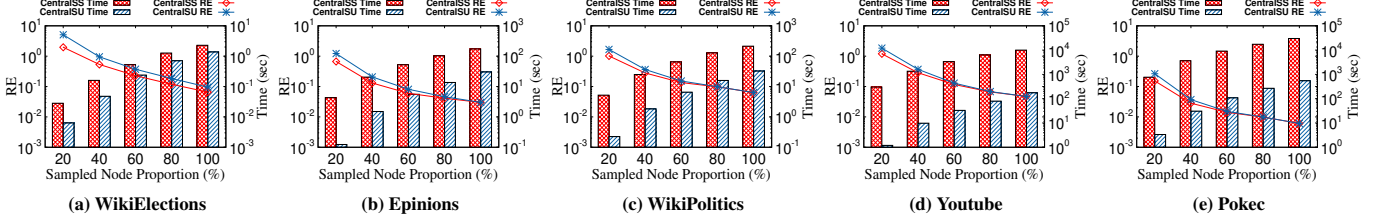


Figure 3: Comparison of both relative error (RE) and runtime between CentralSS and CentralSU across varying sampled node proportions at $\epsilon = 0.1$.

the method outlined in [23], we generate signed labels by randomly allocating 70% of the edges as positive and the remainder as negative.

Competitors and Parameter Selection. To the best of our knowledge, we are the first to investigate the problem of privacy-preserving triangle counting in signed graphs. Given the absence of prior work, we introduce baseline methods and comprehensively evaluate our proposed approaches on five datasets.

In the centralized DP, our proposed approaches inject Laplace noise into the query response, and the difference lies in the noise scale. The approach that calibrates the noise based on the smooth sensitivity is denoted as CentralSS, while the one that calibrates the noise based on the smooth upper bound on local sensitivity is referred to as CentralSU. In addition, we also implement a baseline approach, CentralGS, which injects Laplace noise proportional to the global sensitivity into the response. In order to provide a fair comparison, the baseline approach releases true counts with probability δ , and the perturbed counts otherwise. All approaches satisfy (ϵ, δ) -centralized DP, which has similar semantics to ϵ -centralized DP when $1/\delta$ is at least the number of possible edges in the signed graph [20]. Thus, for all centralized methods, δ is set as $1/(10\binom{n}{2})$, where n is the number of nodes in the signed graph.

In the local model, our proposed two-phase framework is named LocalTwoSU. To justify the performance of our method, we compare it with two other approaches: LocalOne and LocalTwoGS. LocalOne estimates the statistical counts from \hat{G} . LocalTwoGS, similar to LocalTwoSU, operates within a two-phase framework. Nevertheless, the distinction arises in the second phase, where LocalTwoGS uses graph projection to diminish global sensitivity. Subsequently, each node reports the true counts with a probability of δ and, alternatively, provides the counts distorted by Laplace noise in other instances. Note that LocalOne satisfies ϵ -local DP, while LocalTwoGS and LocalTwoSU satisfy (ϵ, δ) -local DP. For parameter selection, we set $\epsilon_1 = 0.5\epsilon$ and $\epsilon_2 = 0.5\epsilon$ for the first phase and second phase,

respectively. For the invalidation probability δ , we set δ to $\frac{1}{10n}$, where n denotes the total number of nodes.

Evaluation Metric. We evaluate the performance of our approaches and baselines for balanced and unbalanced triangle counts across all aforementioned datasets. The accuracy of each method is measured by the *Relative Error* (RE) [14, 15, 29], as defined by the formula $\frac{|\hat{T}^b - T^b| + |\hat{T}^u - T^u|}{T^b + T^u}$. To ensure statistical reliability, each experiment is replicated 100 times across all datasets, with the mean outcomes subsequently reported.

Software and Hardware. We implement all approaches in C+++. All experiments are performed on a machine equipped with an AMD Ryzen Threadripper PRO 3995WX CPU and 256GB main memory. Our code is publicly available on GitHub⁴.

5.2 Experimental Results

Centralized Model. In the first set of experiments, we evaluate the performance of various differentially private methods as the privacy budget ϵ varies between 0.05 and 0.5. The relative errors of each technique for different privacy budgets are presented in Figure 2. The results show that our proposed approaches, CentralSS and CentralSU, both achieve good accuracy over all datasets. When the privacy budget is relatively large, for example, $\epsilon = 0.5$, their relative errors always stay below or close to 1%. With the decrease of ϵ , the accuracies drop, but they are still smaller than 15% even $\epsilon = 0.05$ except the small dataset WikiElections. Moreover, in all cases, our proposed solutions clearly outperform the benchmark method in terms of result accuracy, simply because our methods inject less noise into the true results. The improvement is remarkable since the relative errors are plotted on a logarithmic scale. In the comparison between CentralSS and CentralSU, CentralSS performs more effectively in small datasets and at small ϵ levels, while in other cases, the two methods are comparable. In the WikiElections

⁴<https://github.com/Zening-Li/LDP-GE>

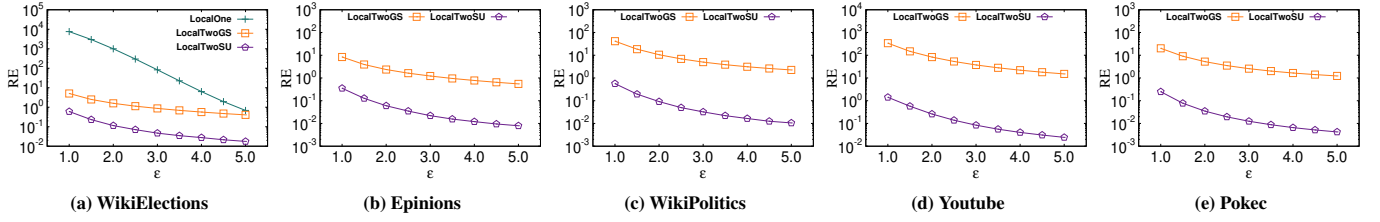


Figure 4: Trade-offs between privacy and relative error (RE) of various mechanisms under local DP.

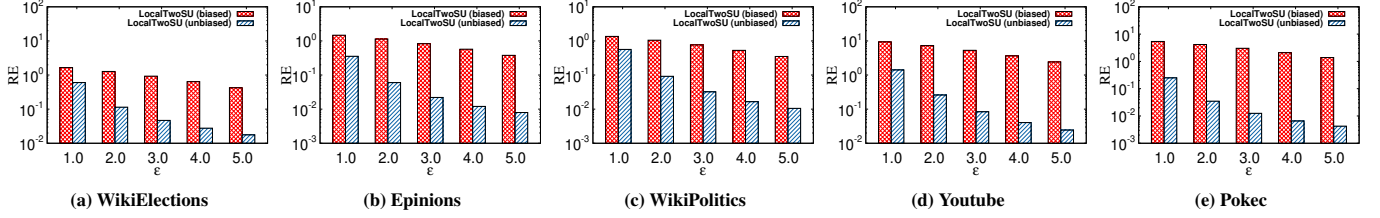


Figure 5: Comparison of relative error (RE) between biased and unbiased LocalTwoSU under local DP.

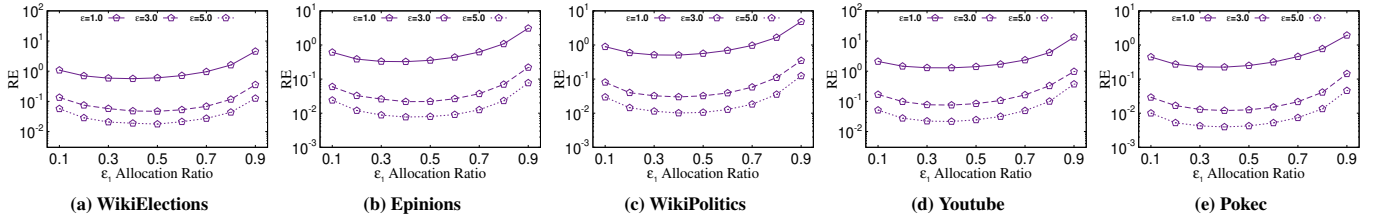


Figure 6: Impact of privacy budget allocation on LocalTwoSU performance under local DP.

dataset, for instance, CentralSS has better accuracy than CentralSU when $\epsilon \leq 0.25$. This phenomenon can be attributed to the fact that the smooth sensitivity is optimal over all smooth bounds. The noise added in CentralSU relies on the smooth upper bound on local sensitivity $S_{\Delta, \beta}(G)$, where the stationary point of $S_{\Delta, \beta}(G)$ is subject to β (i.e., ϵ), $W^1(G)$ and $W^2(G)$ jointly. When these influences take on small values, $S_{\Delta, \beta}(G)$ reaches its peak at increased t values, which could cause $S_{\Delta, \beta}(G)$ to exceed the smooth sensitivity.

In the second set of experiments, we evaluate the impact of the number of nodes n on the relative error and runtime. Specifically, we construct four subgraphs by randomly sampling 20%-80% nodes. The performance of the CentralSS and CentralSU in terms of relative error and runtime at $\epsilon = 0.1$ is summarized in Figure 3. It is observed that the relative errors of both of our proposed methods decrease with the increase in the number of nodes. This result demonstrates that we can accurately estimate triangle counts for large n in the centralized model. When the dataset size is small, such as the proportion of sampled nodes is less than or equal to 60%, the relative error of CentralSS is lower than that of CentralSU in all datasets. However, as the number of nodes increases, the runtime of both approaches increases, but CentralSU remains within a relatively acceptable runtime in all cases. For example, in the Youtube dataset, when the sampling node ratio is 100%, the runtime of CentralSU is

about 175 seconds, while the runtime of CentralSS is about 10^4 seconds. Hence, we infer CentralSS is preferable for smaller datasets, while CentralSU is better suited for larger datasets.

Local Model. To evaluate the effectiveness of our proposed method, LocalTwoSU, we compare its estimation accuracy with two other alternative methods, LocalOne and LocalTwoGS. We vary the privacy budget from 1.0 to 5.0, and the results of these methods are depicted in Figure 4. Note that LocalOne is only tested on the WikiElections dataset, the smallest in our study, because its $O(n^3)$ time complexity renders it impractical for other datasets [14, 16]. In summary, our proposed solution, LocalTwoSU, achieves good accuracy over all datasets. While LocalOne satisfies a stricter notion of privacy, it injects too much noise, which reduces the estimation accuracy. In contrast, LocalTwoSU consistently outperforms all competitors on all datasets because it adds less noise to the final results.

Subsequently, we evaluate the impact of empirical estimation on relative error. Figure 5 charts the performance of LocalTwoSU versus LocalTwo without empirical estimation under varied privacy budgets. It is evident that LocalTwoSU clearly outperforms the method w/o empirical estimation in all cases, which indicates that the unbiased correction does reduce the estimation error. Furthermore, in order to evaluate the impact of privacy budget allocation on estimation accuracy, we adjust the allocation ratio of ϵ_1 (the privacy budget for the first phase) from 0.1 to 0.9 and analyze the relative errors produced by our method under various privacy budgets $\epsilon \in \{1.0, 3.0, 5.0\}$. The

experiment results are illustrated in Figure 6. We observe that an allocation ratio between 0.4 and 0.5 consistently results in the lowest relative error, which indicates a balanced requirement for privacy budgets in both phases. Consequently, adjacency vector perturbation in the first phase and statistical estimation perturbation in the second are equally important for obtaining accurate outcomes.

6 RELATED WORK

In recent years, graph analysis under differential privacy framework has attracted much attention. This section reviews the relevant studies from three perspectives: graph analysis under centralized DP and local DP, as well as signed graph analysis.

Centralized Differentially Private Graph Analysis. Centralized differential privacy techniques are predicated on the existence of a trusted curator that collects individual sensitive data and responds to queries from analysts in a private manner. A substantial portion of graph analysis research under centralized DP has concentrated on estimating various graph statistics. Nissim et al. pioneer the concept of smooth sensitivity and apply it to estimate the minimum spanning tree cost and triangle counts [26]. This approach has been further extended to other subgraph count queries, such as k -stars and k -triangles [20]. Moreover, ladder functions are employed to provide empirical improvements in accuracy with efficient time complexities [40]. Meanwhile, researchers have developed novel techniques for the private release degree distributions and the estimation of local clustering coefficients [4, 7, 21].

Apart from graph statistics, researchers have developed various methods to address other graph-related problems under centralized DP. For instance, some papers have explored the problem of releasing subsets of nodes from the input graph, such as the vertex cover [13] and the densest subgraph [6, 25]. Another research direction is to generate differentially private synthetic graphs [17, 36]. Despite these advancements, there is still a lack of discussion in the literature on estimating balanced and unbalanced triangle counts in signed graphs. This topic remains unexplored under centralized DP.

Graph Analysis under Local Differential Privacy. In contrast to centralized DP, local DP assumes that the data curator is untrusted. Under this model, each user locally obfuscates their individual data and then transmits the perturbed data to the untrusted curator. More specifically, Qin et al. propose a multi-phase algorithm for synthetic graph generation [28]. In [38], the authors introduce the framework LF-GDPR to estimate various graph metrics privately, such as the clustering coefficient and modularity. Brito et al. [2] study the problem of publishing the weighted graphs under centralized and local DP frameworks.

Moreover, the problem of subgraph counting under local DP has attracted much attention in recent research [14–16, 29, 39]. Sun et al. [29] estimate the number of subgraphs under the assumption that each user permits his/her friends to see all his/her connections, but this assumption does not hold in many practical scenarios. Ye et al. [39] apply the randomized response mechanism on the adjacency matrix. However, this method introduces substantial bias to the estimation. Imola et al. estimate the number of k -stars and triangles via multiple rounds of interactions to reduce estimation errors [14] and employ edge sampling to improve communication efficiency [15]. Nevertheless, none of the current LDP mechanisms are specifically

tailored to privately release the number of balanced and unbalanced triangles for signed graphs.

Signed Graph Analysis. The rapid proliferation of online social media has rendered signed graphs increasingly prevalent. Significant advancements in signed graph analysis have been observed in recent years, as detailed in a comprehensive survey by Tang et al. [34]. The origins of signed graph analysis can be traced back to the realm of social psychology, and structural balance theory is first presented in [3]. Then various applications, such as link prediction [22, 24, 37], balancedness analysis [5, 27], community detection [19, 31, 41], and cohesive subgraph mining [23, 30, 32] have been the primary focus of this field. Recently, Arya et al. [1] proposed an efficient algorithm to estimate the number of balanced and unbalanced triangles. However, the direct release of graph statistics compromises individual privacy, an issue our research endeavors to address.

7 CONCLUSION

In this paper, we propose a series of algorithms for counting balanced and unbalanced triangles under centralized and local DP, respectively. In the centralized model, our smooth-sensitivity-based method sets a new benchmark for effectiveness. Meanwhile, our differentially private solution based on the smooth upper bound on local sensitivity not only demonstrates superior efficiency but also provides reliable estimations for most signed graphs. In the local model, our proposed response mechanism introduces less noise to the true results, enhancing data utility. As a direction for future research, we aim to extend our algorithmic developments to encompass other subgraph counts in signed graphs, such as cliques and 4-cycle.

REFERENCES

- [1] Aikta Arya, Pradumn Kumar Pandey, and Akrati Saxena. 2023. Balanced and Unbalanced Triangle Count in Signed Networks. *TKDE* 35, 12 (2023), 12491–12496.
- [2] Felipe T Brito, Victor AE Farias, Cheryl Flynn, Subhabrata Majumdar, Javam C Machado, and Divesh Srivastava. 2023. Global and Local Differentially Private Release of Count-Weighted Graphs. *PACMOD* 1, 2 (2023), 1–25.
- [3] Dorwin Cartwright and Frank Harary. 1956. Structural balance: a generalization of Heider’s theory. *Psychological review* 63, 5 (1956), 277.
- [4] Wei-Yen Day, Ninghui Li, and Min Lyu. 2016. Publishing graph degree distribution with node differential privacy. In *SIGMOD*. 123–138.
- [5] Tyler Derr, Charu Aggarwal, and Jiliang Tang. 2018. Signed network modeling based on structural balance theory. In *CIKM*. 557–566.
- [6] Laxman Dhulipala, Quanquan C Liu, Sofya Raskhodnikova, Jessica Shi, Julian Shun, and Shangdi Yu. 2022. Differential privacy from locally adjustable graph algorithms: k -core decomposition, low out-degree ordering, and densest subgraphs. In *FOCS*. 754–765.
- [7] Xiaofeng Ding, Shujun Sheng, Huajian Zhou, Xiaodong Zhang, Zhifeng Bao, Pan Zhou, and Hai Jin. 2021. Differentially private triangle counting in large graphs. *TKDE* (2021).
- [8] John C Duchi, Michael I Jordan, and Martin J Wainwright. 2013. Local privacy and statistical minimax rates. In *FOCS*. 429–438.
- [9] Cynthia Dwork. 2008. Differential privacy: A survey of results. In *International conference on theory and applications of models of computation*. 1–19.
- [10] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. 2006. Calibrating noise to sensitivity in private data analysis. In *TCC*. 265–284.
- [11] Cynthia Dwork, Aaron Roth, et al. 2014. The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science* 9, 3–4 (2014), 211–407.
- [12] Victor AE Farias, Felipe T Brito, Cheryl Flynn, Javam C Machado, Subhabrata Majumdar, and Divesh Srivastava. 2023. Local dampening: Differential privacy for non-numeric queries via local sensitivity. *The VLDB Journal* (2023), 1–24.
- [13] Anupam Gupta, Katrina Ligett, Frank McSherry, Aaron Roth, and Kunal Talwar. 2010. Differentially private combinatorial optimization. In *SODA*. 1106–1125.
- [14] Jacob Imola, Takao Murakami, and Kamalika Chaudhuri. 2021. Locally differentially private analysis of graph statistics. In *USENIX Security*. 983–1000.

- [15] Jacob Imola, Takao Murakami, and Kamalika Chaudhuri. 2022. Communication-efficient triangle counting under local differential privacy. In *USENIX Security*. 537–554.
- [16] Jacob Imola, Takao Murakami, and Kamalika Chaudhuri. 2022. Differentially Private Triangle and 4-Cycle Counting in the Shuffle Model. In *CCS*. 1505–1519.
- [17] Zach Jorgensen, Ting Yu, and Graham Cormode. 2016. Publishing attributed social graphs with formal privacy guarantees. In *SIGMOD*. 107–122.
- [18] Peter Kairouz, Sewoong Oh, and Pramod Viswanath. 2014. Extremal mechanisms for local differential privacy. *NeurIPS* 27 (2014).
- [19] Yoonsuk Kang, Woncheol Lee, Yeon-Chang Lee, Kyungsik Han, and Sang-Wook Kim. 2021. Adversarial learning of balanced triangles for accurate community detection on signed networks. In *ICDM*. 1150–1155.
- [20] Vishesh Karwa, Sofya Raskhodnikova, Adam Smith, and Grigory Yaroslavtsev. 2014. Private analysis of graph structure. *TODS* 39, 3 (2014), 1–33.
- [21] Shiva Prasad Kasiviswanathan, Kobbi Nissim, Sofya Raskhodnikova, and Adam Smith. 2013. Analyzing graphs with node differential privacy. In *TCC*. 457–476.
- [22] Jure Leskovec, Daniel Huttenlocher, and Jon Kleinberg. 2010. Predicting positive and negative links in online social networks. In *WWW*. 641–650.
- [23] Rong-Hua Li, Qiangqiang Dai, Lu Qin, Guoren Wang, Xiaokui Xiao, Jeffrey Xu Yu, and Shaojie Qiao. 2019. Signed clique search in signed networks: concepts and algorithms. *TKDE* 33, 2 (2019), 710–727.
- [24] Xiaoming Li, Hui Fang, and Jie Zhang. 2017. Rethinking the link prediction problem in signed social networks. In *AAAI*, Vol. 31.
- [25] Dung Nguyen and Anil Vullikanti. 2021. Differentially private densest subgraph detection. In *ICML*. 8140–8151.
- [26] Kobbi Nissim, Sofya Raskhodnikova, and Adam Smith. 2007. Smooth sensitivity and sampling in private data analysis. In *STOC*. 75–84.
- [27] Pradumn Kumar Pandey, Bibhas Adhikari, Mainak Mazumdar, and Niloy Ganguly. 2020. Modeling signed networks as 2-layer growing networks. *TKDE* 34, 7 (2020), 3377–3390.
- [28] Zhan Qin, Ting Yu, Yin Yang, Issa Khalil, Xiaokui Xiao, and Kui Ren. 2017. Generating synthetic decentralized social graphs with local differential privacy. In *CCS*. 425–438.
- [29] Haipei Sun, Xiaokui Xiao, Issa Khalil, Yin Yang, Zhan Qin, Hui Wang, and Ting Yu. 2019. Analyzing subgraph statistics from extended local views with decentralized differential privacy. In *CCS*. 703–717.
- [30] Renjie Sun, Chen Chen, Xiaoyang Wang, Wenjie Zhang, Ying Zhang, and Xuemin Lin. 2023. Efficient maximum signed biclique identification. In *ICDE*. 1313–1325.
- [31] Renjie Sun, Chen Chen, Xiaoyang Wang, Ying Zhang, and Xun Wang. 2020. Stable community detection in signed social networks. *TKDE* 34, 10 (2020), 5051–5055.
- [32] Renjie Sun, Qiuyu Zhu, Chen Chen, Xiaoyang Wang, Ying Zhang, and Xun Wang. 2020. Discovering cliques in signed networks based on balance theory. In *DASFAA*. Springer, 666–674.
- [33] Jiliang Tang, Charu Aggarwal, and Huan Liu. 2016. Recommendations in signed social networks. In *WWW*. 31–40.
- [34] Jiliang Tang, Yi Chang, Charu Aggarwal, and Huan Liu. 2016. A survey of signed network mining in social media. *CSUR* 49, 3 (2016), 1–37.
- [35] Tianhao Wang, Jeremiah Blocki, Ninghui Li, and Somesh Jha. 2017. Locally differentially private protocols for frequency estimation. In *USENIX Security*. 729–745.
- [36] Qian Xiao, Rui Chen, and Kian-Lee Tan. 2014. Differentially private network data release via structural inference. In *KDD*. 911–920.
- [37] Jihang Ye, Hong Cheng, Zhe Zhu, and Minghua Chen. 2013. Predicting positive and negative links in signed social networks by transfer learning. In *WWW*. 1477–1488.
- [38] Qingqing Ye, Haibo Hu, Man Ho Au, Xiaofeng Meng, and Xiaokui Xiao. 2020. LF-GDPR: A framework for estimating graph metrics with local differential privacy. *TKDE* 34, 10 (2020), 4905–4920.
- [39] Qingqing Ye, Haibo Hu, Man Ho Au, Xiaofeng Meng, and Xiaokui Xiao. 2020. Towards locally differentially private generic graph metric estimation. In *ICDE*. 1922–1925.
- [40] Jun Zhang, Graham Cormode, Cecilia M Procopiuc, Divesh Srivastava, and Xiaokui Xiao. 2015. Private release of graph statistics using ladder functions. In *SIGMOD*. 731–745.
- [41] Jun Zhao, Renjie Sun, Qiuyu Zhu, Xiaoyang Wang, and Chen Chen. 2020. Community identification in signed networks: a k-truss based model. In *CIKM*. 2321–2324.