Counting Graphlets of Size k under Local Differential Privacy

Vorapong Suppakitpaisarn[1] The University of Tokyo
Donlapark Ponnoprat[1] Chiang Mai University
Nicha Hirankarn Tsukuba University
Quentin Hillebrand The University of Tokyo

Abstract

The problem of counting subgraphs or graphlets under local differential privacy is an important challenge that has attracted significant attention from researchers. However, much of the existing work focuses on small graphlets like triangles or k-stars. In this paper, we propose a non-interactive, locally differentially private algorithm capable of counting graphlets of any size k. When n is the number of nodes in the input graph, we show that the expected ℓ_2 error of our algorithm is $O(n^{k-1})$. Additionally, we prove that there exists a class of input graphs and graphlets of size k for which any non-interactive counting algorithm incurs an expected ℓ_2 error of $\Omega(n^{k-1})$, demonstrating the optimality of our result. Furthermore, we establish that for certain input graphs and graphlets, any locally differentially private algorithm must have an expected ℓ_2 error of $\Omega(n^{k-1.5})$. Our experimental results show that our algorithm is more accurate than the classical randomized response method.

1 INTRODUCTION

In recent years, differential privacy [Dwork, 2006, Dwork et al., 2014] has gained recognition as the leading approach for ensuring strong privacy protections while still enabling effective data analysis. It achieves this by ensuring that the results of computations remain nearly the same, even if the data of a single individual is altered, thereby protecting personal information. While the initial focus of differential privacy was on tabular datasets [Dwork et al., 2006, McSherry and Talwar, 2007], there has been a growing interest in extending these privacy protections to graph data [Sajadmanesh and Gatica-Perez, 2021, Ye et al., 2020a], which presents a unique set of complexities and challenges.

[1]Equal contribution.

Proceedings of the 28th International Conference on Artificial Intelligence and Statistics (AISTATS) 2025, Mai Khao, Thailand. PMLR: Volume 258. Copyright 2025 by the author(s).

Differential privacy has diversified into several variants to address different use cases, as outlined in [Desfontaines and Pejó, 2020]. Of particular relevance to our work is local differential privacy [Cormode et al., 2018, Evfimievski et al., 2003], which differs from traditional approaches by eliminating the need for a trusted central server. Instead, users anonymize their own private data before transmitting it to an untrusted party. In the realm of graph data, the most widely used variant is edge local differential privacy [Qin et al., 2017], where each user's sensitive data relates to their relationships or connections with others in the graph.

A frequently used obfuscation method is the randomized response technique [Warner, 1965]. In this method, users randomly flip bits in their adjacency vector with a specified probability. The server then aggregates this modified data to create an obfuscated version of the graph. While various graph statistics can be derived from this obfuscated graph, the accuracy of the information is often limited. Algorithms tailored to publish specific statistics generally provide more precise and valuable insights into the graph.

A commonly studied graph statistic in the context of local differential privacy is the number of subgraphs or graphlets [Imola et al., 2021, Hillebrand et al., 2023]. Many studies have focused specifically on publishing triangle counts [Imola et al., 2022a, Hillebrand et al., 2023, Eden et al., 2023, Liu et al., 2024]. For example, [Imola et al., 2021] introduces a non-interactive, edge-local differentially private algorithm. When n is the number of nodes in the input graph, the authors show that the expected ℓ_2-error of the counting algorithm is $O(n^2)$. On the other hand, [Eden et al., 2023] proves that any non-interactive, edge-local differentially private algorithm will have an ℓ_2-error of $\Omega(n^2)$. Additionally, [Eden et al., 2023] further demonstrates that any edge-local differentially private algorithm would result in an ℓ_2-error of $\Omega(n^{1.5})$.

Counting graphlets beyond triangles plays a crucial role in various applications [Ahmed et al., 2015, Bressan et al., 2017, Marcus and Shavitt, 2010]. For example, in sociometric studies, the profile of each subgraph of size k is typically referred to as a k-subgraph census. This census is employed in various social network analyses [Holland and Leinhardt, 1976]. In computer networks, the number of bi-fan motifs can reflect the frequency of points of presence [Feldman and Shavitt, 2008]. Those applications often rely on social network data, where node connections may contain sensitive information. Users may prefer not to share their connection details with central servers, requiring local obfuscation of the data.

This motivates us to address this problem within the edge local differential privacy framework in this paper. However, to the best of our knowledge, only works address the counting of larger subgraphs are: [Hillebrand et al., 2025], which presents an algorithm for counting odd-length cycles, [He et al., 2024], which counts the number of cycles with length four in bipartite graphs, [Imola et al., 2022b], which counts the number of cycles with length four but under a weaker assumption called as shuffle model, and [Betzer et al., 2024], which counts walks of any length.

1.1 Our Contributions

Our contributions can be summarized as follows:

1. In Section 3, we give a non-interactive and edge-local differentially private algorithm for counting any graphlets of k nodes for any k.

2. Also, in Section 3, we demonstrate that our counting algorithm has an expected ℓ_2-error of $O(n^{k-1})$.

3. In Section 4, we demonstrate that there is a class of input graphs and a graphlet of k nodes such that *any* non-interactive and edge-local differentially private algorithm would have an expected ℓ_2-error of $\Theta(n^{k-1})$. Hence, our algorithm is asymptotically optimal in terms of the expected ℓ_2-error.

4. In Section 5, we demonstrate that there is a class of input graphs and a graphlet of k nodes such that *any* edge-local differentlly private algorithm would have an expected ℓ_2-error of $\Theta(n^{k-1.5})$.

We summarize our results in comparison to previous works in Table 1. For the case where the graphlet is a triangle, we set $k = 3$. Our findings are consistent with those of prior research [Imola et al., 2021, Eden et al., 2023], and can be viewed as a generalization of the results on triangle counting. While we draw on some of the algorithmic and proof techniques from these earlier studies, our results are significantly more general. Consequently, we had to introduce several additional proof methods, making our proofs more complex and involved.

Graphlet Type	**Results**
Triangle	$O(n^2)$ (non-interactive) [Imola et al., 2021] $\Omega(n^2)$ (non-interactive) [Eden et al., 2023] $\Omega(n^{1.5})$ (any algorithm) [Eden et al., 2023]
Graphlet with k nodes	$O(n^{k-1})$ (non-interactive) [Section 3] $\Omega(n^{k-1})$ (non-interactive) [Section 4] $\Omega(n^{k-1.5})$ (any algorithm) [Section 5]

Table 1: Our results in this paper compared with the previous results

While we believe that the algorithm proposed in Section 3 is valuable for local differential privacy research, the lower bound results presented in Sections 4 and 5 are equally significant. These results demonstrate that the expected ℓ_2-error can increase by a factor of n when the size of the graphlets increases by one. This suggests that obtaining precise results for large graphlet counting under local differential privacy on general graphs may not be possible. Therefore, developing algorithms tailored to specific types of graphs, as done in [Hillebrand et al., 2025, Betzer et al., 2024], is crucial.

In addition to the theoretical contributions, Section 6 presents experiments that validate the effectiveness of our algorithm. Since no existing algorithm is specifically designed for subgraph counting, we compare our approach with the classical randomized response technique. Our algorithm achieves an ℓ_2-error up to 36 times smaller than that of randomized response when $n = 100$. Moreover, we observe that the performance gap increases as n grows, suggesting that the improvement becomes significantly more pronounced for larger graphs.

1.2 Related Works

Graph data mining under local differential privacy is still an emerging area, whereas differential privacy has been explored for many years in various studies, including [Gupta et al., 2010, Olatunji et al., 2023]. According to [Imola et al., 2021], local differential privacy generally focuses on protecting the privacy of edges or relationships, with a few exceptions such as [Zhang et al., 2020]. In contrast, differential privacy at large can conceal whether a specific individual or node belongs to a social network, as seen in [Hay et al., 2009, Raskhodnikova and Smith, 2016]. While both edge and node privacy models exist, node-level privacy is not applicable within the scope of local differential privacy.

In addition to the subgraph counting, there are many graph algorithms proposed in the framework of local differential privacy. For examples, recent research has developed techniques for estimating the densest subgraph, k-core decomposition, and degeneracy within the local differential privacy framework [Dhulipala et al., 2022, Dhulipala et al., 2023, Dinitz et al., 2023, Henzinger et al., 2024].

2 PRELIMINARIES

2.1 Notations

Let the input graph be $G = (V, E)$, where $V = \{v_1, \ldots, v_n\}$ and $E \subseteq V^2$. Each node v_i represents a user in our social network, and the edges E represent the relationships between them. Now, consider a graphlet $\mathsf{G} = (\mathsf{V}, \mathsf{E})$ with $\mathsf{V} = \{u_1, \ldots, u_k\}$ and $\mathsf{E} \subseteq \mathsf{V}^2$. The number of graphlets G in G, denoted as $\mathsf{G}(G)$, is defined as the number of distinct subgraphs $(V', E') \subseteq G$ that are isomorphic to G.

For each $i \in [1, n]$, let $a_i = [a_{i,1}, \ldots, a_{i,n}]$ be the adjacency list of user v_i, where $a_{i,j} = 1$ indicates an edge between v_i and v_j (i.e., $(v_i, v_j) \in E$), and $a_{i,j} = 0$ otherwise. The degree of node v_i, denoted by d_i, represents the number of edges connected to v_i. At the start of any locally differentially private algorithm, we assume that each user v_i has knowledge of only their own adjacency vector a_i. The vector is a sensitive information of the user.

We deploy the expected ℓ_2-error to measure the precision of the graphlet counting algorithms in this work. Let $\tilde{\mathsf{G}}(G)$ be the estimation of the number of graphlets obtaind from the algorithm. The expected ℓ_2-error is defined as:

$$\sqrt{\int_{-\infty}^{\infty} \Pr[\tilde{\mathsf{G}}(G) = i](\mathsf{G}(G) - i)^2 \, di}.$$

This expected ℓ_2-error is widely recognized as a standard approach for evaluating the precision of various non-deterministic algorithms such as [Drews and Kohler, 2023, Huang et al., 2024].

2.2 Edge Local Differential Privacy

We define two adjacency lists, a and a', as neighbors if they differ by exactly one bit, meaning that one can be transformed into the other by adding or removing a single edge involving node v_i. The concept of edge local differential privacy can be described as follows:

Definition 1 (Local differentially private query)**.** *Let $\epsilon > 0$. A randomized query $\mathcal{R}$ is said to be ϵ-edge locally differentially private for node v_i if, for any pair of neighboring adjacency lists a and a', and for any possible set of outcomes S, we have that $\mathbb{P}[\mathcal{R}(a) \in S] \leq e^{\epsilon}\mathbb{P}[\mathcal{R}(a') \in S]$.*

Definition 2 ([Qin et al., 2017])**.** *An algorithm $\mathcal{A}$ is defined as ϵ-edge locally differentially private if, for any user v_i and for any possible set of queries $\mathcal{R}_1, \ldots, \mathcal{R}_\kappa$ which $\mathcal{A}$ posed to user v_i, where each query $\mathcal{R}_j$ is ϵ_j-edge locally differentially private (for $1 \leq j \leq \kappa$), the total privacy loss satisfies $\epsilon_1 + \cdots + \epsilon_\kappa \leq \epsilon$.*

When an algorithm A is ϵ-edge locally differentially private, it is also said to have a privacy budget of ϵ. In the subsequent definition, we define the non-interactive algorithm.

Definition 3 (Non-interactive algorithm)**.** *An algorithm $\mathcal{A}$ is considered non-interactive if it issues a single query to all users, and the query is independent of the responses from other users.*

2.3 Unbiased Randomized Response

In this subsection, we examine the randomized response query, designed to release a privacy-preserving version of the adjacency vector.

Definition 4 ([Warner, 1965, Wang et al., 2016])**.** *For $\epsilon > 0$, the randomized response query $\mathcal{R}$ with privacy budget ϵ operates on an adjacency list $a = (a_1, \ldots, a_n)$, producing an obfuscated output $\tilde{a} = (\tilde{a}_1, \ldots, \tilde{a}_n)$ such that*

$$\mathbb{P}[\mathcal{R}(\tilde{a}_i) = 1] = \begin{cases} \frac{e^\epsilon}{1+e^\epsilon} & \text{if } a_i = 1, \\ \frac{1}{1+e^\epsilon} & \text{if } a_i = 0. \end{cases}$$

This query guarantees ϵ-edge local differential privacy.

A graph $\tilde{G}$ can be constructed using the collection of obfuscated adjacency vectors from all users. By analyzing the statistics of this obfuscated graph $\tilde{G}$, we can release various insights, such as the number of subgraphs [Ye et al., 2020b, Imola et al., 2021, Imola et al., 2022a, Hillebrand et al., 2023]. In Section 6, we will use the estimates derived from $\tilde{G}$ as our benchmark for comparison. Also, we denote the obfuscated version of variable $a_{i,j}$ as $\tilde{a}_{i,j}$, i.e. $\tilde{a}_{i,j} = \mathcal{R}(a_{i,j})$ when $\mathcal{R}$ is the randomized response query.

The variable $\tilde{a}_{i,j}$ provides a biased estimate of $a_{i,j}$, as $\mathbb{E}[\tilde{a}_{i,j}] \neq a_{i,j}$. In the triangle counting algorithm proposed by [Eden et al., 2023], the authors address this bias by using an adjusted estimator $\hat{a}_{i,j} := \frac{e^\epsilon+1}{e^\epsilon-1}\tilde{a}_{i,j} - \frac{1}{e^\epsilon-1}$ to estimate the number of triangles. They demonstrate that this adjustment ensures $\mathbb{E}[\hat{a}_{i,j}] = a_{i,j}$.

3 OUR ALGORITHM

Algorithm 1 presents the method for estimating the number of graphlets $\mathsf{G} = (\mathsf{V} = \{u_1, \ldots, u_k\}, \mathsf{E})$ in $G = (V = \{v_1, \ldots, v_n\}, E)$ under local differential privacy. A bijective function $\pi : \mathsf{V} \to \mathsf{V}$ is an automorphism of G if $(\pi(u_i), \pi(u_j)) \in \mathsf{E}$ if and only if $(u_i, u_j) \in \mathsf{E}$. In the algorithm, we use $A(\mathsf{G})$ to denote number of automorphisms of G. At Line 1, we apply the randomized response technique to collect edge information from all users. In Line 2, we adjust the results to remove bias introduced by the randomized response. Finally, in Lines 3-4, we compute the number of graphlets based on the unbiased results.

Algorithm 1: Estimate the number of graphlets under local differential privacy

```
Input: Graph $G = (V, E)$, privacy budget $\epsilon$,
       graphlet $\mathsf{G} = (\mathsf{V}, \mathsf{E})$, number of
       automorphisms of $\mathsf{G}$ denoted by $A(\mathsf{G})$
Output: Estimation of the number of graphlets
        $\mathsf{G}$ in $G$
1 [All users and server] Inquire the randomized
    response query with privacy budget $\epsilon$ to all
    users. Let the result be $\tilde{a}_{i,j}$ for all $i, j$.
2 [Server] For all $i, j$, calculate
    $\hat{a}_{i,j} := \frac{e^\epsilon+1}{e^\epsilon-1}\tilde{a}_{i,j} - \frac{1}{e^\epsilon-1}$.
3 [Server] Let $\mathcal{D}$ be the set of tuples
    $\mathcal{W} = (v_{\ell_1}, \ldots, v_{\ell_k}) \in V^k$ where $\ell_i \in [n]$ such that
    $v_{\ell_i} \neq v_{\ell_j}$ for all $i \neq j$. For all $\mathcal{W} \in \mathcal{D}$, calculate
    $\tilde{W}(\mathcal{W}, \mathsf{G}) = \prod_{\{u_i,u_j\}\in\mathsf{E}} \hat{a}_{\ell_i,\ell_j}$.
4 [Server] return
    $\tilde{\mathsf{G}}(G) = \left(\sum_{\mathcal{W}\in\mathcal{D}} \tilde{W}(\mathcal{W}, \mathsf{G})\right) / A(\mathsf{G})$.
```

It is clear that the algorithm is non-interactive and satisfies ϵ-edge local differential privacy, since the only query posed to users is the randomized response query in Line 1. In the remaining part of this section, we demonstrate that the ℓ_2-error of this algorithm is $O(n^{k-1})$.

Consider a tuple $\mathcal{W} = (v_{\ell_1}, \ldots, v_{\ell_k}) \in V^k$ where $\ell_i \in [n]$ such that $v_{\ell_i} \neq v_{\ell_j}$ for all $i \neq j$, and for all $\{u_i, u_j\} \in \mathsf{E}$, the edge $\{v_{\ell_i}, v_{\ell_j}\}$ is in E. Let $W(G, \mathsf{G})$ be the number of such tuples. We obtain the following lemma.

Lemma 3.1. $\mathsf{G}(G) = W(G, \mathsf{G})/A(\mathsf{G})$.

Proof. Let S be the set of subgraphs of G isomorphic to G, and let W be the set of tuples $\mathcal{W} = (v_{\ell_1}, \ldots, v_{\ell_k}) \in V^k$ such that $v_{\ell_i} \neq v_{\ell_j}$ for all $i \neq j$, and for all $\{u_i, u_j\} \in \mathsf{E}$, the edge $\{v_{\ell_i}, v_{\ell_j}\}$ is in E. To demonstrate this lemma, we provide a surjective function $g : \mathsf{W} \to \mathsf{S}$ such that $|g^{-1}(S)| = A(\mathsf{G})$ for all $S \in \mathsf{S}$.

For $\mathcal{W} = (v_{\ell_1}, \ldots, v_{\ell_k}) \in \mathsf{W}$, we define a subgraph $g(\mathcal{W}) = (V', E')$ such that $V' = \{v_{\ell_1}, \ldots, v_{\ell_k}\}$ and $E' = \{\{v_{\ell_i}, v_{\ell_j}\} : \{u_i, u_j\} \in \mathsf{E}\}$. By the definition of W, $g(\mathcal{W})$ is in S.

On the other hand, consider a subgraph $(V' = \{v_{\ell_1}, \ldots, v_{\ell_k}\}, E') \in \mathsf{S}$. We have that a tuple $\mathcal{W} = (v_{q_1}, \ldots, v_{q_k}) \in \mathsf{W}$ has $g(\mathcal{W}) = (V', E')$ if 1) $\{v_{q_1}, \ldots, v_{q_k}\} = \{v_{\ell_1}, \ldots, v_{\ell_k}\}$, and 2) $\{\{v_{q_i}, v_{q_j}\} : \{u_i, u_j\} \in \mathsf{E}\} = E'$.

Let $\mathsf{A}(\mathsf{G})$ be the set of automorphisms of G. For $\pi \in \mathsf{A}(\mathsf{G})$, define $\pi' : [n] \to [n]$ such that $\pi'(i) = j$ if $\pi(u_i) = u_j$. Because π is an automorphism, we have $\{u_{\pi'(i)}, u_{\pi'(j)}\} \in \mathsf{E}$ for $\{u_i, u_j\} \in \mathsf{E}$. Therefore, $g(\mathcal{W}) = (V', E')$ if and only if there exists $\pi \in \mathsf{A}(\mathsf{G})$ such that $v_{\ell_{\pi'(i)}} = v_{q_i}$ for all i. The number of distinct tuples $\mathcal{W}$ such that $g(\mathcal{W}) = (V', E')$ is then equal to the number of distinct functions π in $\mathsf{A}(\mathsf{G})$, which is $A(\mathsf{G})$. □

We show in the following lemma that the estimated value from our algorithm has no bias.

Lemma 3.2. *The estimator* $\tilde{\mathsf{G}}(G)$ *is an unbiased estimator of* $\mathsf{G}(G)$.

Proof. It follows from the independence of $\tilde{a}_{i,j}$'s and Lemma 3.1 that

$$\begin{aligned}
&\mathbb{E}\left[\tilde{\mathsf{G}}(G)\right] \\
&= \mathbb{E}\left[\sum_{(v_{\ell_1},\ldots,v_{\ell_k})\in\mathcal{D}} \prod_{\{u_i,u_j\}\in\mathsf{E}} \hat{a}_{\ell_i,\ell_j}\right] / A(\mathsf{G}) \\
&= \sum_{(v_{\ell_1},\ldots,v_{\ell_k})\in\mathcal{D}} \prod_{\{u_i,u_j\}\in\mathsf{E}} \mathbb{E}\left[\hat{a}_{\ell_i,\ell_j}\right] / A(\mathsf{G}) \\
&= \sum_{(v_{\ell_1},\ldots,v_{\ell_k})\in\mathcal{D}} \prod_{\{u_i,u_j\}\in\mathsf{E}} a_{\ell_i,\ell_j} / A(\mathsf{G}) \\
&= W(G, \mathsf{G})/A(\mathsf{G}) = \mathsf{G}(G).
\end{aligned}$$

□

In the following lemma, we consider the property of the random variable $\tilde{W}(\mathcal{W}, \mathsf{G})$. This property will be used later to determine the variance of our estimator.

Lemma 3.3. *There are at most $O(n^{2k-2})$ pairs of $\mathcal{W}, \mathcal{W}' \in \mathcal{D}$ such that the covariance of $\tilde{W}(\mathcal{W}, \mathsf{G})$ and $\tilde{W}(\mathcal{W}', \mathsf{G})$ is more than 0.*

Proof. Let $\mathcal{W} = (v_{\ell_1}, \ldots, v_{\ell_k}) \in \mathcal{D}$ and $\mathcal{W}' = (v_{\ell'_1}, \ldots, v_{\ell'_k}) \in \mathcal{D}$. We have $\mathrm{Cov}(\tilde{W}(\mathcal{W}, \mathsf{G}), \tilde{W}(\mathcal{W}', \mathsf{G})) > 0$ only if the calculations of $\tilde{W}(\mathcal{W}, \mathsf{G})$ and $\tilde{W}(\mathcal{W}', \mathsf{G})$ both involve the same randomized data $\tilde{a}_{\ell_i, \ell_j}$ for some i, j. This occurs if $|\{v_{\ell_1}, \ldots, v_{\ell_k}\} \cap \{v_{\ell'_1}, \ldots, v_{\ell'_k}\}| \geq 2$. This results in:

$$k \leq |\{v_{\ell_1}, \ldots, v_{\ell_k}\} \cup \{v_{\ell'_1}, \ldots, v_{\ell'_k}\}| \leq 2k - 2. \quad (1)$$

Let us denote by $\mathsf{C}(\mathsf{k})$ the number of pairs $\mathcal{W}, \mathcal{W}'$ such that $|\{v_{\ell_1}, \ldots, v_{\ell_k}\} \cup \{v_{\ell'_1}, \ldots, v_{\ell'_k}\}| = \mathsf{k}$. We have from (1) that $\mathsf{C}(\mathsf{k}) > 0$ only if $k \leq \mathsf{k} \leq 2k - 2$. There are at most n^{k} combinations of $\{v_{\ell_1}, \ldots, v_{\ell_k}\} \cup \{v_{\ell'_1}, \ldots, v_{\ell'_k}\}$. From the n^{k} possible combinations, there are $\binom{\mathsf{k}}{k}$ distinct sets of the form $\{v_{\ell_1}, \ldots, v_{\ell_k}\}$. For any fixed set $\{v_{\ell_1}, \ldots, v_{\ell_k}\}$, there are $\binom{k}{2k-\mathsf{k}}$ ways to determine the intersection $\{v_{\ell_1}, \ldots, v_{\ell_k}\} \cap \{v_{\ell'_1}, \ldots, v_{\ell'_k}\}$, which uniquely defines the second set $\{v_{\ell'_1}, \ldots, v_{\ell'_k}\}$. Therefore, there are $\binom{\mathsf{k}}{k} \cdot \binom{k}{2k-\mathsf{k}}$ distinct ways to partition the union $\{v_{\ell_1}, \ldots, v_{\ell_k}\} \cup \{v_{\ell'_1}, \ldots, v_{\ell'_k}\}$ into the two sets.

There are $k!$ tuples for each of the set $\{v_{\ell_1}, \ldots, v_{\ell_k}\}$ and $k!$ tuples for each of the set $\{v_{\ell'_1}, \ldots, v_{\ell'_k}\}$. Hence, the number of such pairs $\mathcal{W}, \mathcal{W}'$ such that $|\{v_{\ell_1}, \ldots, v_{\ell_k}\} \cap \{v_{\ell'_1}, \ldots, v_{\ell'_k}\}| \geq 2$ is no more than:

$$\sum_{\mathsf{k}=k}^{2k-2} n^{\mathsf{k}} \cdot \binom{\mathsf{k}}{k} \cdot \binom{k}{2k-\mathsf{k}} \cdot (k!)^2 = O(n^{2k-2}).$$

□

We are now ready to give the ℓ_2-error of our estimator.

Theorem 3.4. *The ℓ_2-error of the estimator $\tilde{\mathsf{G}}_k(G)$ is $O(n^{k-1})$.*

Proof. The squared ℓ_2-error is the sum of the variance and the squared bias. From Lemma 3.2, we have established that the bias is zero. Therefore, the remainder of this proof will focus on analyzing the variance of our estimator.

We first examine the variance of $\tilde{W}(\mathcal{W}, \mathsf{G})$. This random variable is a product of a linear function involving $O(k^2) = O(1)$ Bernoulli variables. Consequently, we can deduce that the variance of $\tilde{W}(\mathcal{W}, \mathsf{G})$ is bounded by some constant C. Applying the Cauchy-Schwarz inequality, we obtain that $\mathrm{Cov}(\tilde{W}(\mathcal{W}, \mathsf{G}), \tilde{W}(\mathcal{W}', \mathsf{G})) \leq C$.

From Lemma 3.3, we obtain that:

$$\begin{aligned}
&\mathrm{Var}[\tilde{\mathsf{G}}_k(G)] \\
= \;& \frac{1}{(A(\mathsf{G}))^2}\Big[\sum_{\mathcal{W} \in \mathcal{D}} \mathrm{Var}[\tilde{W}(\mathcal{W}, \mathsf{G})] + \\
& \qquad \sum_{\mathcal{W} \in \mathcal{D}} \mathrm{Cov}[\tilde{W}(\mathcal{W}, \mathsf{G}), \tilde{W}(\mathcal{W}', \mathsf{G})]\Big] \\
\leq \;& \frac{1}{(A(\mathsf{G}))^2}\left[C \cdot n^k + C \cdot O(n^{2k-2})\right] = O(n^{2k-2}).
\end{aligned}$$

□

4 LOWER BOUND FOR NON-INTERACTIVE ALGORITHMS

In this section, we aim to demonstrate that there exists a class of graphs for which any non-interactive local differentially private algorithm that attempts to count cliques of size k incurs an ℓ_2-error of $\Omega(n^{k-1})$. Due to the technical complexity of the proof and space constraints, we outline the key proof ideas here and provide the full proof in the Appendix.

In this section, we assume, without loss of generality, that n is an integer divisible by three. The class of graphs under consideration is defined as follows.

For each $\mathbf{X} \in \{0,1\}^{n/3 \times n/3}$, $\mu = \{\mu_1, \ldots, \mu_{n/3}\} \in \{0,1\}^{n/3}$, and $\upsilon = \{\upsilon_1, \ldots, \upsilon_{n/3}\} \in \{0,1\}^{n/3}$, we construct the graph $\mathsf{G}_k^{\mu,\upsilon}(\mathbf{X})$ by the following steps:

1. We have k set of nodes $\mathsf{U}, \mathsf{Y}, \mathsf{W}_1, \ldots, \mathsf{W}_{k-2}$. All of the sets have size $n/3$. Let $\mathsf{U} = \{\mathsf{u}_1, \ldots, \mathsf{u}_{n/3}\}$, $\mathsf{Y} = \{\mathsf{y}_1, \ldots, \mathsf{y}_{n/3}\}$, and $\mathsf{W}_p = \{\mathsf{w}_{p,1}, \ldots, \mathsf{w}_{p,n/3}\}$ for all $1 \leq p \leq k-2$. The number of nodes in the graph is then equal to $k \cdot n/3$.
2. We have $\{\mathsf{u}_i, \mathsf{y}_j\} \in E$ if $\mathbf{X}_{i,j} = 1$.
3. We have $\{\mathsf{u}_i, \mathsf{w}_{p,j}\} \in E$ for all p, j if $\mu_i = 1$.
4. We have $\{\mathsf{w}_{p,i}, \mathsf{w}_{q,j}\} \in E$ for all i, j, and $p \neq q$.
5. We have $\{\mathsf{y}_i, \mathsf{w}_{p,j}\} \in E$ for all p, j if $\upsilon_i = 1$.

Our gadget is shown as in Figure 6.

The main result of this section is as follows. The proof of this theorem is provided in the Appendix.

Theorem 4.1. *Let $\mathsf{n} = \frac{k \cdot n}{3}$ be the number of nodes in the graph $\mathsf{G}_k^{\mu,\upsilon}(\mathbf{X})$. There exists no non-interactive ϵ-differentially private algorithm which can estimate the K_k count in the graph $\mathsf{G}_k^{\mu,\upsilon}(\mathbf{X})$ with expected ℓ_2-error in $o(\mathsf{n}^{k-1})$.*

Informally, it is shown in [Eden et al., 2023] that for any differentially private algorithm $\mathcal{A}$, there

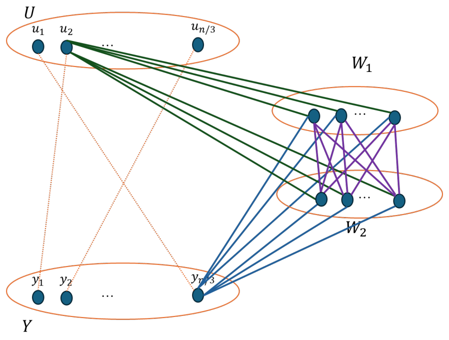


Figure 1: The gadget we use to show the lower bound of ℓ_2-error when counting K_4.

exists $\mathsf{N} = \Theta(n^2)$, and a set of parameters $\mathbf{X}, \mu_1, \ldots, \mu_\mathsf{N}, \upsilon_1, \ldots, \upsilon_\mathsf{N}$, such that the error in estimating the number of triangles for *many* graphs in the set $\{\mathsf{G}_3^{\mu_1,\upsilon_1}(\mathbf{X}), \ldots, \mathsf{G}_3^{\mu_\mathsf{N},\upsilon_\mathsf{N}}(\mathbf{X})\}$ is $\Omega(n^2)$.

We show that the number of K_k cliques in $\mathsf{G}_k^{\mu,\upsilon}(\mathbf{X})$ is $(n/3)^{k-3}$ times the number of triangles in $\mathsf{G}_3^{\mu,\upsilon}(\mathbf{X})$. If there existed a local differentially private algorithm that could estimate the number of K_k cliques in $\mathsf{G}_k^{\mu,\upsilon}(\mathbf{X})$ with ℓ_2-error of $o(n^{k-1})$, we could divide the result by $(n/3)^{k-3}$ to obtain an estimator for the number of triangles in $\mathsf{G}_3^{\mu,\upsilon}(\mathbf{X})$ with an ℓ_2-error of $o(n^2)$. This contradicts the result in [Eden et al., 2023].

5 LOWER BOUND FOR ANY ALGORITHM

In this section, we explore the lower bound of estimating the number of graphlets with size k under any local differentially private algorithm. Specifically, we give the lower bounds for the estimation of the number of cycles with length k in a graph G, denoted by $C_k(G)$.

Suppose the number of nodes n is even. Let $\mathbf{x} = (x_1, \ldots, x_{n/2})$ be a bit vector of length $n/2$. We define a graph $G^\mathbf{x} = (V, E^\mathbf{x})$ as follows:

1. Set $V = \{v_1, \ldots, v_n\}$.
2. For $u, v \in V$ where $u \neq v$ and $\{u, v\} \neq \{v_{2i-1}, v_{2i}\}$ for all $1 \leq i \leq n/2$, we have $\{u, v\} \in E^\mathbf{x}$.
3. For $1 \leq i \leq n/2$, we include $\{v_{2i-1}, v_{2i}\} \in E^\mathbf{x}$ if $x_i = 1$. Let $E_0^\mathbf{x} = \{\{v_{2i-1}, v_{2i}\} : 1 \leq i \leq n/2\} \cap E^\mathbf{x}$. That is $|E_0^\mathbf{x}| = |\mathbf{x}|$.

To establish a lower bound for the expected error, let us assume that all edges in $G^\mathbf{x}$, except those in $E_0^\mathbf{x}$, are non-sensitive and publicly known. It is understood that the expected error is lower when some edges are publicly known compared to when all edges are sensitive [Eden et al., 2023]. Hence, this assumption can be utilized to derive a lower bound for the expected error. By the assumption, the set of possible graphs is then $\mathcal{G} := \{G^\mathbf{x} : \mathbf{x} \in \{0,1\}^{n/2}\}$.

For a subset $E_p \subseteq E_0^\mathbf{x}$ of size p, let C_p denote the number of k-cycles in $G^\mathbf{x}$ that include all edges in E_p. Since all edges in $E_0^\mathbf{x}$ are identical, the number of k-cycles that use p edges from $E_0^\mathbf{x}$ is $\binom{|\mathbf{x}|}{p} \cdot \mathsf{C}_p$. Therefore, the total number of k-cycles in $G^\mathbf{x}$ is given by:

$$f(|\mathbf{x}|) = C_k(G^\mathbf{0}) + \sum_{p=1}^{|\mathbf{x}|} \binom{|\mathbf{x}|}{p} \cdot \mathsf{C}_p. \qquad (2)$$

Lemma 5.1. *For any $\mathbf{x}, \mathbf{x}'$ and $D \geq 0$ such that $|\mathbf{x}| - |\mathbf{x}'| = D$, we have*

$$C_k\left(G^\mathbf{x}\right) - C_k\left(G^{\mathbf{x}'}\right) = \Omega\left(n^{k-2} \cdot D\right).$$

Proof. We have:

$$\begin{aligned}
& C_k\left(G^\mathbf{x}\right) - C_k\left(G^{\mathbf{x}'}\right) \\
= & \left(C_k(G^\mathbf{0}) + \sum_{p=1}^{|\mathbf{x}|} \binom{|\mathbf{x}|}{p} \cdot \mathsf{C}_p\right) \\
& \quad - \left(C_k(G^\mathbf{0}) + \sum_{p=1}^{|\mathbf{x}'|} \binom{|\mathbf{x}'|}{p} \cdot \mathsf{C}_p\right) \\
= & \left(|\mathbf{x}| - |\mathbf{x}'|\right) \cdot \mathsf{C}_1 \\
& \quad + \left(\sum_{p=2}^{|\mathbf{x}|} \binom{|\mathbf{x}|}{p} \cdot \mathsf{C}_p - \sum_{p=2}^{|\mathbf{x}'|} \binom{|\mathbf{x}'|}{p} \cdot \mathsf{C}_p\right) \\
\geq & \left(|\mathbf{x}| - |\mathbf{x}'|\right) \cdot \mathsf{C}_1.
\end{aligned}$$

Because the number of cycles that use one edge in $E_0^\mathbf{x}$ is $\Theta(n^{k-2})$, we obtain the lemma statement. □

The primary theorem in this section relies on the following lower bound result from [Joseph et al., 2019].

Theorem 5.2 (Theorem 5.3 from version 2 of the ArXiv preprint [Joseph et al., 2019])**.** *Let the sensitive information of users $v_1, \ldots, v_n$ be represented by $x_1, \ldots, x_n \in \{0,1\}$. No locally differentially private algorithm can compute $\sum_{i=1}^n x_i$ with an ℓ_2-error of $o(\sqrt{n})$.*

Now, we are ready to present the main theorem of this section.

Theorem 5.3. *There is no local ϵ-edge differential privacy algorithm that can estimate the number of C_k with an expected ℓ_2-loss of $o(n^{k-1.5})$.*

Proof. Assume, for contradiction, that such an algorithm exists. We call this algorithm Algorithm $\mathcal{A}$. Suppose we give a graph $G^{\mathbf{x}} \in \mathcal{G}$ to Algorithm $\mathcal{A}$ to count the number of C_k. Next, we introduce an algorithm $\mathcal{B}$ designed to estimate $|\mathbf{x}|$. Recall the function f defined in (2). Algorithm $\mathcal{B}$ outputs q such that $f(q)$ is closest to the output of Algorithm $\mathcal{A}$ on $G^{\mathbf{x}}$.

If the expected ℓ_2-loss of Algorithm $\mathcal{A}$, which is a local ϵ-edge differential privacy algorithm, is $o(n^{k-1.5})$, then, by Lemma 5.1, Algorithm $\mathcal{B}$, also a local ϵ-edge differential privacy algorithm, must have an estimation error of $o(\sqrt{n})$. However, this contradicts with Theorem 5.2. □

6 EXPERIMENTAL RESULTS

6.1 Settings

In this section, we discuss our experimental results, which are used to confirm the validity of our theoretical results. Our experiment settings are as follows:

Evaluation Method For each set of parameters, we generate a graph G where the actual number of subgraphs is denoted as $\mathsf{S}(G)$. We then execute our algorithm ten times, obtaining estimates $i_1, \dots, i_{10}$. Then we evaluate the precision using the root mean square error, which can be computed as:

$$\sqrt{\sum_{t=1}^{10} (i_t - \mathsf{S}(G))^2}.$$

We also evaluate our algorithm using the relative root mean square error, which can be computed as:

$$\frac{\sqrt{\sum_{t=1}^{10} (i_t - \mathsf{S}(G))^2}}{\mathsf{S}(G)}.$$

Input graphs Recall that n is the size of the input graph and k is the size of graphlets. The computation complexity of our algorithm is $O(n^k)$. Hence, we cannot conduct experiments on a large input graphs at this state. We therefore choose to conduct experiments on synthetic graphs with size 10, 20, ..., 100 in this paper.

The input graphs are derived from the Barabási–Albert model [Albert and Barabási, 2002] and the stochastic block model [Holland et al., 1983], selected to represent two distinct types of social networks. The stochastic block model captures networks with multiple clusters, while the Barabási–Albert model illustrates networks characterized by a small number of central nodes.

Given the relatively small number of nodes, we configured the stochastic block model with two blocks, each containing an equal number of nodes, $n/2$. The probability of a link between two nodes within the same block is set to 0.25, while the probability of a link between nodes in different blocks is 0.05. In the Barabási–Albert model, the degree of each newly added node is set to $n/5$.

Graphlet In this experiment, we focus on counting four-node cycles. Based on our algorithm and the corresponding theoretical results, we expect that the experimental outcomes would not significantly vary for different types of graphlets. Therefore, we selected the smallest graphlet to minimize the computation time of our algorithm.

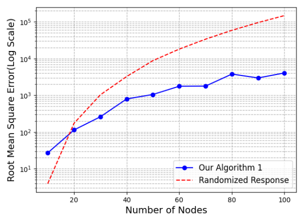
Benchmark Algorithm Apart from the work by Imola et al. [Imola et al., 2022b], which reports the number of four-length cycles under an assumption weaker than local differential privacy, and the work by He et al. [He et al., 2024], which assumes that the input graph is bipartite, we are not aware of any algorithm specifically designed for counting four-length cycles. Therefore, in this experiment, we choose to compare our algorithm with the classical randomized response technique.

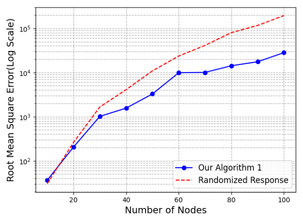
Privacy Budget We set the privacy budget to 1 and 5 in our experiment, which may seem relatively high. However, due to the small number of nodes in this experiment, a higher privacy budget is necessary to ensure meaningful results. We believe that for larger graphs, a significantly lower privacy budget would suffice without compromising the quality of the publication.

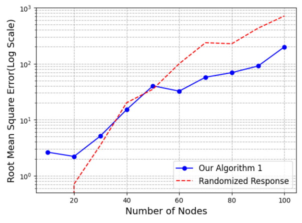
The code for our experiments are available at `https://github.com/gcwsilver/Counting-Graphlets-of-Size-k-under-Local-Differential-Privacy`.

6.2 Results

We give the root mean square error of our Algorithm 1 compared with the classical randomized response in Figure 2. We found that our algorithm significantly better than the randomized rounding in almost all of the settings. The figures indicate that our algorithm shows greater improvement in the stochastic block model, demonstrating its clear advantage for clustered social networks. Additionally, the results reveal that the improvement increases as the number of nodes grows and the privacy budget ϵ decreases. This suggests that in practical scenarios with a much larger number of nodes and a smaller privacy budget, our algorithm's improvement will be even more significant than what is shown in the plots. Specifically, in the stochastic block model, our algorithm achieves an improvement of approximately 36 times when the number of nodes is 100 and the privacy budget is one.


(a)


(b)


(c)

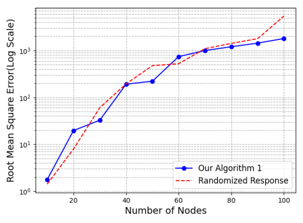
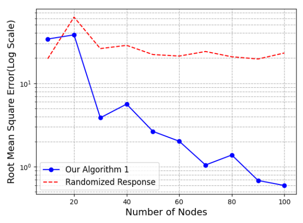

(d)

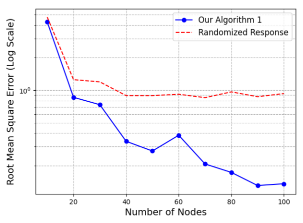
Figure 2: Root mean square error of our Algorithm 1 compared with the randomized response mechanism (a) in the stochastic block model when the privacy budget $\epsilon = 1$, (b) in the Barabási–Albert model when $\epsilon = 1$, (c) in the stochastic block model when $\epsilon = 5$, and (d) in the Barabási–Albert model when $\epsilon = 5$. We omit the result for the randomized response in (c) when $n = 10$ as the error is zero, making it impossible to display on a log scale plot.

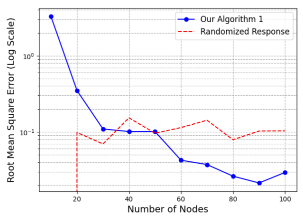
The results, showing that our improvements are greater for a larger number of nodes, align with our theoretical findings. Specifically, the expected ℓ_2-error of our algorithm is $O(n^{k-1})$, while the expected ℓ_2-error for the randomized response is $\Theta(n^k)$.

In Figure 3, we present the relative root mean square error, calculated as the relative mean square error divided by the average value. The plots clearly demonstrate that the relative error decreases as the number of nodes increases. Even with as few as 100 nodes, the plot confirms that our algorithm produces meaningful results. Specifically, the relative error is below 0.6 when the privacy budget $\epsilon = 1$, and it drops to less than 0.03 when $\epsilon = 5$. This decline in relative error with increasing nodes aligns with our theoretical predictions. Since the probability that any tuple of length four forms a cycle in the stochastic block model is $\Theta(1)$, the expected number of 4-cycles in graphs generated by the model increases as $\Theta(n^4)$. Meanwhile, the ℓ_2-error of our algorithm scales as $\Theta(n^3)$.

Conversely, the plot shows that the relative error from the randomized response mechanism remains constant


(a)


(b)


(c)

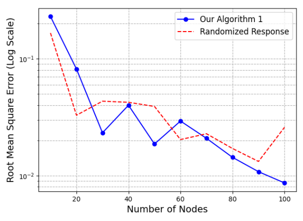
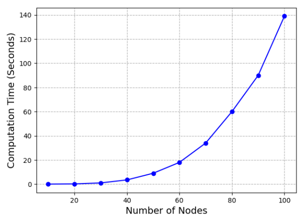

(d)

Figure 3: Relative root mean square error of our Algorithm 1 compared with the randomized response mechanism in the stochastic block model (a) in the stochastic block model when the privacy budget $\epsilon = 1$, (b) in the Barabási–Albert model when $\epsilon = 1$, (c) in the stochastic block model when $\epsilon = 5$, and (d) in the Barabási–Albert model when $\epsilon = 5$. We omit the result for the randomized response in (c) when $n = 10$ as the error is zero, making it impossible to display on a log scale plot.

as the number of nodes increases. This outcome is also theoretically anticipated, as the ℓ_2-error of the classical mechanism is $\Theta(n^4)$, matching the order of the expected number of 4-cycles.


(a)

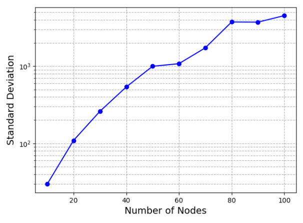

(b)

Figure 4: (a) Computation time of our Algorithm 1 as a function of the number of nodes (b) Standard deviation of our results for graphs derived from the stochastic block model when the privacy budget is set to one.

Next, we address the computation time of our algorithm, which represents its main drawback. In Figure 4a, we display the computation time for Algorithm 1 executed on Google Colab. The figure shows that the computation time increases rapidly, following an $O(n^4)$ growth pattern. For instance, the algorithm takes 139 seconds to compute the number of

4-cycles in a graph with 100 nodes. This suggests that the computation time may be too lengthy for practical applications.

On the other hands, we can optimize the counting of some specific subgraph based on the randomized response mechanism. As the results of the mechanism is a network, we can utilize subgraph counting algorithms (such as [Ahmed et al., 2015, Rahman et al., 2014]) to speed up the count. The results based on the mechanism is hence can be calculated in a much more efficient way.

In Figure 4b, we present the standard deviation for graphs generated under the stochastic block model when the privacy budget is set to one. The standard deviation closely matches the ℓ_2-error depicted in Figure 2a, confirming that the error in our algorithm is solely due to variance, with no indication of bias.

7 CONCLUSION AND FUTURE WORKS

In this paper, we establish the lower bound for the ℓ_2-error in the task of counting subgraphs of size k under local differential privacy. We show that for a certain class of input graphs of size n, any locally differentially private algorithm must incur an ℓ_2-error of at least $\Omega(n^{k-1.5})$, and for any non-interactive locally differentially private algorithm, the ℓ_2-error is at least $\Omega(n^{k-1})$. Moreover, we match this lower bound by proposing a non-interactive, locally differentially private algorithm that achieves an ℓ_2-error of $O(n^{k-1})$ for counting any subgraph of size k. Our algorithm leverages the randomized response mechanism and employs a similar unbiased technique as the one used for triangle counting in [Eden et al., 2023]. We generalize this approach by using subgraph automorphisms to extend the triangle counting method to any subgraphs.

The primary limitation of our algorithm is its computational complexity. Currently, the algorithm has a computation time of $O(n^k)$, which is impractically high for real-world applications. Nonetheless, we believe it provides a foundation for scalable and locally private counting. We believe that incorporating the sampling technique proposed for non-private subgraph counting [Ribeiro et al., 2021] could enhance our algorithm's speed. We anticipate that, with these techniques, our algorithm will achieve computation times comparable to state-of-the-art non-private counting methods.

For the theoretical results in this work, we derive upper and lower bounds for the ℓ_2-error in terms of the number of users n. Here, we treat ϵ as a constant, since in practical scenarios, $1/\epsilon$ is typically much smaller than n and rarely exceeds 100. However, expressing both upper and lower bounds in terms of n and ϵ would offer deeper insights for future research. Therefore, we plan to derive such bounds in future work.

Acknowledgments

The authors sincerely thank the anonymous reviewers for their valuable comments, which have greatly contributed to improving this paper. Vorapong Suppakitpaisarn receives partial support from KAKENHI Grants 21H05845 and 23H04377, as well as JST NEXUS Grant Y2024L0906031. Quentin Hillebrand is partially supported by KAKENHI Grant 20H05965, and by JST SPRING Grant Number JPMJSP2108. This research was partially conducted during Donlapark Ponnoprat's stay at The University of Tokyo, hosted by Masaaki Imaizumi. Nicha Hirankarn joined this project upon the introduction of Sira Srisawat (Chulalongkorn University). The authors would also like to take this opportunity to thank Masaaki Imaizumi and Sira Srisawat.

References

[Ahmed et al., 2015] Ahmed, N. K., Neville, J., Rossi, R. A., and Duffield, N. (2015). Efficient graphlet counting for large networks. In *ICDM 2015*, pages 1–10.

[Albert and Barabási, 2002] Albert, R. and Barabási, A.-L. (2002). Statistical mechanics of complex networks. *Reviews of modern physics*, 74(1):47.

[Betzer et al., 2024] Betzer, L., Suppakitpaisarn, V., and Hillebrand, Q. (2024). Publishing number of walks and Katz centrality under local differential privacy. In *UAI 2024*, pages 377–393.

[Bressan et al., 2017] Bressan, M., Chierichetti, F., Kumar, R., Leucci, S., and Panconesi, A. (2017). Counting graphlets: Space vs time. In *WWW 2017*, pages 557–566.

[Cormode et al., 2018] Cormode, G., Jha, S., Kulkarni, T., Li, N., Srivastava, D., and Wang, T. (2018). Privacy at scale: Local differential privacy in practice. In *SIGMOD 2018*, pages 1655–1658.

[Desfontaines and Pejó, 2020] Desfontaines, D. and Pejó, B. (2020). SoK: Differential privacies. *Proceedings on Privacy Enhancing Technologies*, 2020(2):288–313.

[Dhulipala et al., 2023] Dhulipala, L., Li, G. Z., and Liu, Q. C. (2023). Near-optimal differentially private k-core decomposition. *arXiv preprint arXiv:2312.07706*.

[Dhulipala et al., 2022] Dhulipala, L., Liu, Q. C., Raskhodnikova, S., Shi, J., Shun, J., and Yu, S. (2022). Differential privacy from locally adjustable graph algorithms: k-core decomposition, low out-degree ordering, and densest subgraphs. In *FOCS 2022*, pages 754–765.

[Dinitz et al., 2023] Dinitz, M., Kale, S., Lattanzi, S., and Vassilvitskii, S. (2023). Improved differentially private densest subgraph: Local and purely additive. *arXiv preprint arXiv:2308.10316*.

[Drews and Kohler, 2023] Drews, S. and Kohler, M. (2023). Analysis of the expected L_2 error of an over-parametrized deep neural network estimate learned by gradient descent without regularization. *arXiv preprint arXiv:2311.14609*.

[Dwork, 2006] Dwork, C. (2006). Differential privacy. In *ICALP 2006*, pages 1–12.

[Dwork et al., 2006] Dwork, C., McSherry, F., Nissim, K., and Smith, A. (2006). Calibrating noise to sensitivity in private data analysis. In *TCC 2006*, pages 265–284.

[Dwork et al., 2014] Dwork, C., Roth, A., et al. (2014). The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science*, 9(3–4):211–407.

[Eden et al., 2023] Eden, T., Liu, Q. C., Raskhodnikova, S., and Smith, A. (2023). Triangle counting with local edge differential privacy. In *ICALP 2023*, pages 52:1–52:21.

[Evfimievski et al., 2003] Evfimievski, A., Gehrke, J., and Srikant, R. (2003). Limiting privacy breaches in privacy preserving data mining. In *PODS 2003*, pages 211–222.

[Feldman and Shavitt, 2008] Feldman, D. and Shavitt, Y. (2008). Automatic large scale generation of internet pop level maps. In *GLOBECOM 2008*, pages 1–6.

[Gupta et al., 2010] Gupta, A., Ligett, K., McSherry, F., Roth, A., and Talwar, K. (2010). Differentially private combinatorial optimization. In *SODA 2010*, pages 1106–1125.

[Hay et al., 2009] Hay, M., Li, C., Miklau, G., and Jensen, D. (2009). Accurate estimation of the degree distribution of private networks. In *ICDM 2009*, pages 169–178.

[He et al., 2024] He, Y., Wang, K., Zhang, W., Lin, X., Ni, W., and Zhang, Y. (2024). Butterfly counting over bipartite graphs with local differential privacy. In *ICDE 2024*, pages 2351–2364.

[Henzinger et al., 2024] Henzinger, M., Sricharan, A., and Zhu, L. (2024). Tighter bounds for local differentially private core decomposition and densest subgraph. *arXiv preprint arXiv:2402.18020*.

[Hillebrand et al., 2023] Hillebrand, Q., Suppakitpaisarn, V., and Shibuya, T. (2023). Communication cost reduction for subgraph counting under local differential privacy via hash functions. *arXiv preprint arXiv:2312.07055*.

[Hillebrand et al., 2025] Hillebrand, Q., Suppakitpaisarn, V., and Shibuya, T. (2025). Cycle counting under local differential privacy for degeneracy-bounded graphs. *STACS 2025*.

[Holland et al., 1983] Holland, P. W., Laskey, K. B., and Leinhardt, S. (1983). Stochastic blockmodels: First steps. *Social networks*, 5(2):109–137.

[Holland and Leinhardt, 1976] Holland, P. W. and Leinhardt, S. (1976). Local structure in social networks. *Sociological methodology*, 7:1–45.

[Huang et al., 2024] Huang, Z., Tang, T., Chen, S., Lin, S., Jie, Z., Ma, L., Wang, G., and Liang, X. (2024). Making large language models better planners with reasoning-decision alignment. In *ECCV 2024*, pages 73–90.

[Imola et al., 2021] Imola, J., Murakami, T., and Chaudhuri, K. (2021). Locally differentially private analysis of graph statistics. In *USENIX 2021*, pages 983–1000.

[Imola et al., 2022a] Imola, J., Murakami, T., and Chaudhuri, K. (2022a). Communication-efficient triangle counting under local differential privacy. In *USENIX 2022*, pages 537–554.

[Imola et al., 2022b] Imola, J., Murakami, T., and Chaudhuri, K. (2022b). Differentially private triangle and 4-cycle counting in the shuffle model. In *CCS 2022*, pages 1505–1519.

[Joseph et al., 2019] Joseph, M., Mao, J., Neel, S., and Roth, A. (2019). The role of interactivity in local differential privacy. In *FOCS 2019*, pages 94–105.

[Liu et al., 2024] Liu, Y., Wang, T., Liu, Y., Chen, H., and Li, C. (2024). Edge-protected triangle count estimation under relationship local differential privacy. *IEEE Transactions on Knowledge and Data Engineering*, 36(10):5138–5152.

[Marcus and Shavitt, 2010] Marcus, D. and Shavitt, Y. (2010). Efficient counting of network motifs. In *ICDCS 2010*, pages 92–98.

[McSherry and Talwar, 2007] McSherry, F. and Talwar, K. (2007). Mechanism design via differential privacy. In *FOCS 2007*, pages 94–103.

[Olatunji et al., 2023] Olatunji, I. E., Funke, T., and Khosla, M. (2023). Releasing graph neural networks with differential privacy guarantees. *Transactions on Machine Learning Research*.

[Qin et al., 2017] Qin, Z., Yu, T., Yang, Y., Khalil, I., Xiao, X., and Ren, K. (2017). Generating synthetic decentralized social graphs with local differential privacy. In *CCS 2017*, pages 425–438.

[Rahman et al., 2014] Rahman, M., Bhuiyan, M. A., and Al Hasan, M. (2014). Graft: An efficient graphlet counting method for large graph analysis. *IEEE Transactions on Knowledge and Data Engineering*, 26(10):2466–2478.

[Raskhodnikova and Smith, 2016] Raskhodnikova, S. and Smith, A. (2016). Differentially private analysis of graphs. *Encyclopedia of Algorithms*.

[Ribeiro et al., 2021] Ribeiro, P., Paredes, P., Silva, M. E., Aparicio, D., and Silva, F. (2021). A survey on subgraph counting: Concepts, algorithms, and applications to network motifs and graphlets. *ACM Computing Surveys (CSUR)*, 54(2):1–36.

[Sajadmanesh and Gatica-Perez, 2021] Sajadmanesh, S. and Gatica-Perez, D. (2021). Locally private graph neural networks. In *CCS 2021*, pages 2130–2145.

[Wang et al., 2016] Wang, Y., Wu, X., and Hu, D. (2016). Using randomized response for differential privacy preserving data collection. In *EDBT/ICDT Workshops*.

[Warner, 1965] Warner, S. L. (1965). Randomized response: A survey technique for eliminating evasive answer bias. *Journal of the American Statistical Association*, 60(309):63–69.

[Ye et al., 2020a] Ye, Q., Hu, H., Au, M. H., Meng, X., and Xiao, X. (2020a). LF-GDPR: A framework for estimating graph metrics with local differential privacy. *IEEE Transactions on Knowledge and Data Engineering*, 34(10):4905–4920.

[Ye et al., 2020b] Ye, Q., Hu, H., Au, M. H., Meng, X., and Xiao, X. (2020b). Towards locally differentially private generic graph metric estimation. In *ICDE 2020*, pages 1922–1925.

[Zhang et al., 2020] Zhang, H., Latif, S., Bassily, R., and Rountev, A. (2020). Differentially-private control-flow node coverage for software usage analysis. In *USENIX Security 2020*, pages 1021–1038.

Checklist

1. For all models and algorithms presented, check if you include:
   (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. [Yes, we outline the setting and the algorithm in Sections 2 and 3.]
   (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. [Yes, we analyze the privacy of our algorithm in Section 3. However, we do not provide an explicit analysis of the time and space complexity, as they are straightforward.]
   (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. [Not Applicable]

2. For any theoretical claim, check if you include:
   (a) Statements of the full set of assumptions of all theoretical results. [Not applicable. Our algorithm and lower bounds apply to the general case, so no specific assumptions are required.]
   (b) Complete proofs of all theoretical results. [Yes. We provide the proofs for all theoretical results in Sections 3-5, and the supplementary material.]
   (c) Clear explanations of any assumptions. [Not applicable. Our algorithm and lower bounds apply to the general case, so no specific assumptions are required.]

3. For all figures and tables that present empirical results, check if you include:
   (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). [Yes. We provide the code and the instructions as a supplemental material. The data is synthesized from the code.]
   (b) All the training details (e.g., data splits, hyperparameters, how they were chosen). [Not Applicable. We do not train any machine learning model in this work.]
   (c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). [Yes. We describe that in Section 6.]
   (d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). [Yes. We describe that in Section 6.]

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:
   (a) Citations of the creator If your work uses existing assets. [Not Applicable. We do not use any existing asset in this work.]
   (b) The license information of the assets, if applicable. [Not Applicable. We do not use any existing asset in this work.]
   (c) New assets either in the supplemental material or as a URL, if applicable. [Yes. We include the code in the supplemental material.]
   (d) Information about consent from data providers/curators. [Not Applicable. We do not use any existing asset in this work.]
   (e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. [Not Applicable. We do not collect any information in this work.]

5. If you used crowdsourcing or conducted research with human subjects, check if you include:
   (a) The full text of instructions given to participants and screenshots. [Not Applicable. We do not conduct any experiment with human subjects.]
   (b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. [Not Applicable. We do not conduct any experiment with human subjects.]
   (c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. [Not Applicable. We do not conduct any experiment with human subjects.]

Supplemental Materials for the Manuscript "Counting Graphlets of Size k under Local Differential Privacy"

8 MISSING PROOFS FOR SECTION 4

We give details for our lower bound proof in Section 4. In the section, we discuss that there is a class of input graphs such that any non-interactive locally differentially private algorithm which counts the number of cliques with size k must have an ℓ_2 error of $\Omega(n^{k-1})$. Let the input graph be G, and let the number of cliques in the graph be $K_k(G)$.

8.1 Result of [Eden et al., 2023]

Our proof will rely on the lower bound of error in counting triangles as established in [Eden et al., 2023]. In the paper, the authors demonstrate that no non-interactive ϵ-edge local differential privacy mechanism can count triangles with an expected ℓ_2-error less than $o(n^2)$. However, we cannot directly utilize this result as a black box for our proof. Instead, we derive our lower bound by leveraging the gadget described in the paper.

Gadget $G^{\mu,\upsilon}(\mathbf{X})$

We begin by describing the gadget used in this paper. Without loss of generality, we assume that $3|n$. Let $\mu, \upsilon \in \{0,1\}^{n/3}$. Also, let $\mathbf{X}$ be a 0-1 matrix of size $\frac{n}{3} \times \frac{n}{3}$. We define the gadget, denoted by $G^{\mu,\upsilon}(\mathbf{X})$, as follows:

1. The graph is a tripartite graph. Let the sets of nodes in each partition be $U = \{u_1, \ldots, u_{n/3}\}$, $Y = \{y_1, \ldots, y_{n/3}\}$, and $W = \{w_1, \ldots, w_{n/3}\}$.
2. For each pair i, j, there is an edge between u_i and y_j if $\mathbf{X}_{i,j} = 1$; otherwise, there is no edge between them.
3. For each i, there is an edge between u_i and w_j *for all* j if $\mu_i = 1$; otherwise, there is no edge between them.
4. For each i, there is an edge between y_i and w_j *for all* j if $\upsilon_i = 1$; otherwise, there is no edge between them.

We illustrate the gadget in Figure 5.

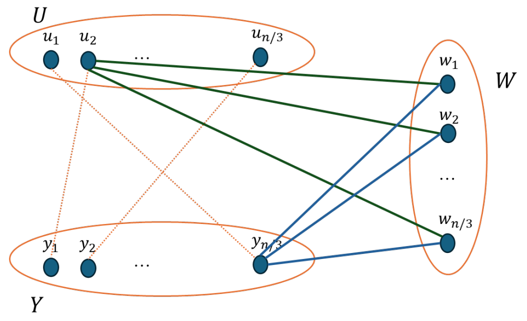


Figure 5: Gadget used in [Eden et al., 2023] for showing the lower bound in ℓ_2-error of estimating the number of triangles.

Edge Differentially Private Algorithm $\mathcal{C}$

For each node v, let $\mathbf{a}_v$ represent the adjacency vector of v in the graph $G^{\mu,\upsilon}(\mathbf{X})$. A non-interactive ϵ-edge differential privacy algorithm, denoted by $\mathcal{C}$, can be described as follows:

1. Each node v applies an ϵ-differential privacy algorithm, denoted by $\mathcal{L}_v$ (referred to as a randomizer), to its adjacency vector $\mathbf{a}_v$. The node then sends the result $\mathcal{L}_v(\mathbf{a}_v)$ to a central server.
2. The central server applies an aggregator function $\mathcal{A}$ to the collected results from all randomizers and outputs $\mathcal{A}\left(\langle \mathcal{L}_v(\mathbf{a}_v)\rangle_{v\in V}\right)$ as the final result.

Let us now recall the graph $G^{\mu,\upsilon}(\mathbf{X})$. For a node u_i, we denote its adjacency vector in the graph as $\mathbf{a}_{u_i}(\mu,\upsilon,\mathbf{X})$. Given any vectors $\mu,\mu',\upsilon,\upsilon'$ where $\mu_i=\mu'_i$, we have $\mathbf{a}_{u_i}(\mu,\upsilon,\mathbf{X})=\mathbf{a}_{u_i}(\mu',\upsilon',\mathbf{X})$. Therefore, we can simplify the notation for $\mathbf{a}_{u_i}(\mu,\upsilon,\mathbf{X})$ to $\mathbf{a}_{u_i}(\mu_i,\mathbf{X})$. Similarly, $\mathbf{a}_{v_i}(\mu,\upsilon,\mathbf{X})$ can be abbreviated as $\mathbf{a}_{v_i}(\upsilon_i,\mathbf{X})$, and, by the definition, $\mathbf{a}_{w_i}(\mu,\upsilon,\mathbf{X})$ can be abbreviated as $\mathbf{a}_{w_i}(\mu,\upsilon)$ for all $w_i\in W$. Since the computation of w_i do not involve any private information, we can assume that those computations are done at the central server.

Edge Differentially Private Algorithm $\mathcal{D}_{\mathcal{C}}$

Let us assume that the algorithm $\mathcal{C}$ is designed to estimate the number of triangles in a given graph. Additionally, suppose that for any graph $G^{\mu,\upsilon}(\mathbf{X})$, the edges between the node sets $U\cup Y$ and W (described by μ,υ) are not private, while the edges between the node sets U and Y (described by $\mathbf{X}$) are private. For a fixed private matrix $\mathbf{X}$, the authors of [Eden et al., 2023] have designed a 2ϵ-differentially private algorithm, denoted as Algorithm $\mathcal{D}_{\mathcal{C}}$, which can estimate the number of triangles in $\mathsf{N}=\Theta(n^2)$ graphs in the set $\{G^{\mu,\upsilon}(\mathbf{X}):\mu,\upsilon\in\{0,1\}^{n/3}\}$ based on the functions $\mathcal{L}_v$ and $\mathcal{A}$ within the algorithm $\mathcal{C}$. Let the set of these $\Theta(n^2)$ graphs be $\mathcal{S}_{\mathbf{X}}$. The algorithm $\mathcal{D}_{\mathcal{C}}$ can be described as follows:

1. For each $u_i\in U$, the node u_i applies the algorithm $\mathcal{L}_{u_i}$ to the adjacency vectors $\mathbf{a}_{u_i}(0,\mathbf{X})$ and $\mathbf{a}_{u_i}(1,\mathbf{X})$. The node then sends the results to the central server. Let these results be $\mathcal{L}_{u_i}(0)$ and $\mathcal{L}_{u_i}(1)$. Since the node publishes two results, this process is 2ϵ-differentially private.
2. For each $y_i\in Y$, the node y_i applies the algorithm $\mathcal{L}_{y_i}$ to the adjacency vectors $\mathbf{a}_{y_i}(0,\mathbf{X})$ and $\mathbf{a}_{y_i}(1,\mathbf{X})$. The node then sends the results to the central server. Let these results be $\mathcal{L}_{y_i}(0)$ and $\mathcal{L}_{y_i}(1)$. Since the node publishes two results, this process is 2ϵ-differentially private.
3. Then, for all graphs in $\mathcal{S}_{\mathbf{X}}=\{G^{\mu^{(1)},\upsilon^{(1)}}(\mathbf{X}),\ldots,G^{\mu^{(\mathsf{N})},\upsilon^{(\mathsf{N})}}(\mathbf{X})\}$:
   (a) Each user w_i first applies the randomized algorithm $\mathcal{L}_{w_i}$ to its adjacency vector $\mathbf{a}_{w_i}(\mu^{(1)},\upsilon^{(1)},\mathbf{X}),\ldots,\mathbf{a}_{w_i}(\mu^{(\mathsf{N})},\upsilon^{(\mathsf{N})},\mathbf{X})$. Then, the user sends the result to the central server. Note that all edges incident to w_i are unrelated to the private matrix $\mathbf{X}$, ensuring that this process does not leak any privacy.
   (b) The central server then applies the algorithm $\mathcal{A}$ to $\langle\langle\mathcal{L}_{u_i}(\mu_i^{(t)})\rangle_{i=1}^{n/3},\ \langle\mathcal{L}_{y_i}(\upsilon_i^{(t)})\rangle_{i=1}^{n/3},\langle\mathbf{a}_{w_i}(\mu^{(t)},\upsilon^{(t)})\rangle_{i=1}^{n/3}\rangle$ for each $t\in[\mathsf{N}]$. The result obtained from this step is denoted as $\mathcal{D}_{\mathcal{C}}(\mu^{(1)},\upsilon^{(1)},\mathbf{X}),\ldots,\mathcal{D}_{\mathcal{C}}(\mu^{(\mathsf{N})},\upsilon^{(\mathsf{N})},\mathbf{X})$.

The authors of [Eden et al., 2023] show the following result in their paper.

Theorem 8.1 ([Eden et al., 2023]). *Let n be the number of nodes in the graph $G^{\mu^{(t)},\nu^{(t)}}(\mathbf{X})$. There exists $\mathsf{c}>0$, $\mathsf{N}=\mathsf{c}\cdot n^2$, such that no non-interactive ϵ-differentially private algorithms $\mathcal{C}$ where, for all $\mathbf{X}$, $\mu^{(1)},\ldots,\mu^{(\mathsf{N})}$, $\upsilon^{(1)},\ldots,\upsilon^{(\mathsf{N})}$,*

$$\Pr\left[\left|\left\{t\in\{1,\ldots,\mathsf{N}\}:\left|\mathcal{D}_{\mathcal{C}}(\mu^{(t)},\nu^{(t)},\mathbf{X})-K_3\left(G^{\mu^{(t)},\nu^{(t)}}(\mathbf{X})\right)\right|\le\frac{n^2}{12}\right\}\right|>\frac{\mathsf{N}}{5184}\right]>\frac{1}{6}. \tag{3}$$

8.2 Our Construction

We define our construction in this subsection.

Graph $\mathsf{G}_k^{\mu,\upsilon}(\mathbf{X})$

For each $G^{\mu,\upsilon}(\mathbf{X})$, we construct the graph $\mathsf{G}_k^{\mu,\upsilon}(\mathbf{X})$ by the following steps:

1. We have k set of nodes $\mathsf{U}, \mathsf{Y}, \mathsf{W}_1, \dots, \mathsf{W}_{k-2}$. All of the sets have size $n/3$. Let $\mathsf{U} = \{\mathsf{u}_1, \dots, \mathsf{u}_{n/3}\}$, $\mathsf{Y} = \{\mathsf{y}_1, \dots, \mathsf{y}_{n/3}\}$, and $\mathsf{W}_p = \{\mathsf{w}_{p,1}, \dots, \mathsf{w}_{p,n/3}\}$ for all $1 \leq p \leq k-2$. The number of nodes in the graph, denoted by n, is then equal to $k \cdot n/3$.

2. We have $\{\mathsf{u}_i, \mathsf{y}_j\} \in E$ if $\mathbf{X}_{i,j} = 1$.

3. We have $\{\mathsf{u}_i, \mathsf{w}_{p,j}\} \in E$ for all p, j if $\mu_i = 1$.

4. We have $\{\mathsf{w}_{p,i}, \mathsf{w}_{q,j}\} \in E$ for all i, j, and $p \neq q$.

5. We have $\{\mathsf{y}_i, \mathsf{w}_{p,j}\} \in E$ for all p, j if $\upsilon_i = 1$.

Our gadget is shown as in Figure 6.

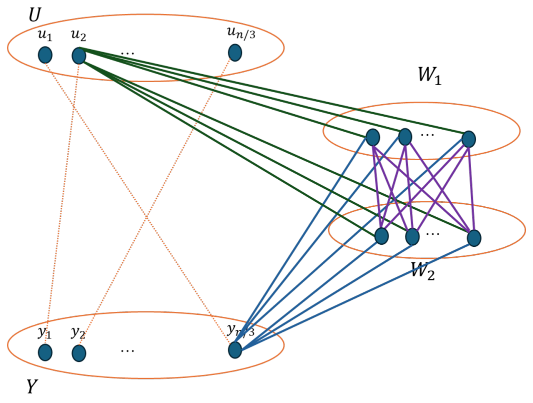


Figure 6: The gadget we use to show the lower bound of ℓ_2-error when counting K_4.

By the construction, we obtain the following lemma.

Lemma 8.2. *Let $K_3(G^{\mu,\upsilon}(\mathbf{X}))$ be the number of triangles in the graph $G^{\mu,\upsilon}(\mathbf{X})$. The number of k-clique in the graph $\mathsf{G}^{\mu,\upsilon}(\mathbf{X})$ is $K_3(G^{\mu,\upsilon}(\mathbf{X})) \cdot (n/3)^{k-3}$.*

Proof. Since there is no edge between nodes in the same set, a set S is a k-clique in $\mathsf{G}^{\mu,\upsilon}(\mathbf{X})$ only if it contains one node in U, one node in Y, and one node in each of the set $\mathsf{W}_1, \dots, \mathsf{W}_{k-2}$. Let $S \cap \mathsf{U} = \{\mathsf{u}_{i_S}\}$ and $S \cap \mathsf{Y} = \{\mathsf{y}_{j_S}\}$. We know that S is a clique if and only if $\mu_{i_S} = 1$, $\upsilon_{j_S} = 1$, and $\mathbf{X}_{i_S,j_S} = 1$. Let $\mathcal{S}$ be the set of such (i_S, j_S). We notice that the number of triangles in the graph $G^{\mu,\upsilon}(\mathbf{X})$ is equal to $|\mathcal{S}| \cdot n/3$. On the other hand, the number of k-cliques in the graph $\mathsf{G}^{\mu,\upsilon}(\mathbf{X})$ is equal to $|\mathcal{S}| \cdot (n/3)^{k-2}$. We hence obtain the lemma statement. □

Edge Differentially Private Algorithm C

We then introduce a non-interactive ϵ-edge differentially private algorithm C which estimates the number of K_k in the graph $\mathsf{G}^{\mu,\upsilon}(\mathbf{X})$. For each node v, let $\mathbf{a}_\mathsf{v}$ represent the adjacency vector of v in the graph $\mathsf{G}^{\mu,\upsilon}(\mathbf{X})$. A non-interactive ϵ-edge differential privacy algorithm, denoted by C, can be described as follows:

1. Each node v applies an ϵ-differential privacy algorithm, denoted by L_v (referred to as a randomizer), to its adjacency vector $\mathbf{a}_\mathsf{v}$. The node then sends the result $\mathsf{L}_\mathsf{v}(\mathbf{a}_\mathsf{v})$ to a central server.

2. The central server applies an aggregator function $\mathcal{A}$ to the collected results from all randomizers and outputs $\mathsf{A}\left(\langle \mathsf{L}_\mathsf{v}(\mathbf{a}_\mathsf{v})\rangle_\mathsf{v}\right)$ as the final result.

Let us now recall the graph $\mathsf{G}^{\mu,\upsilon}(\mathbf{X})$. For a node $\mathsf{u}_i \in \mathsf{U}$, we denote its adjacency vector in the graph as $\mathbf{a}_{\mathsf{u}_i}(\mu,\upsilon,\mathbf{X})$. Given any vectors $\mu,\mu',\upsilon,\upsilon'$ where $\mu_i = \mu'_i$, we have $\mathbf{a}_{\mathsf{u}_i}(\mu,\upsilon,\mathbf{X}) = \mathbf{a}_{\mathsf{u}_i}(\mu',\upsilon',\mathbf{X})$. Therefore, we can simplify the notation for $\mathbf{a}_{\mathsf{u}_i}(\mu,\upsilon,\mathbf{X})$ to $\mathbf{a}_{\mathsf{u}_i}(\mu_i,\mathbf{X})$. Similarly, for each node $\mathsf{y}_i \in \mathsf{Y}$, $\mathbf{a}_{\mathsf{y}_i}(\mu,\upsilon,\mathbf{X})$ can be abbreviated as $\mathbf{a}_{\mathsf{y}_i}(\upsilon_i,\mathbf{X})$, and $\mathbf{a}_{\mathsf{w}_{i,j}}(\mu,\upsilon,\mathbf{X})$ can be abbreviated as $\mathbf{a}_{\mathsf{w}_{i,j}}(\mu,\upsilon)$. Since the computation of $\mathsf{w}_{i,j}$ do not involve any private information, we can assume that those computations are done at the central server.

Constructing an Estimator of Triangle Count $h(\mathsf{C})$ from C

In the next step, we define an estimator of triangle count $h(\mathsf{C})$ for the graph $G^{\mu,\upsilon}(\mathbf{X})$ from the estimator C. Recall that the set of nodes of the graph $G^{\mu,\upsilon}(\mathbf{X})$ are U, Y, and W. The construction can be done as follows:

1. For each $u_i \in U$, given μ_i, the user possesses sufficient information to construct the adjacency vector of the node $\mathsf{u}_i \in \mathsf{U}$. It can then compute $\mathsf{L}_{\mathsf{u}_i}(\mu_i,\mathbf{X})$ according to the construction C. In the construction $h(\mathsf{C})$, we have $\mathcal{L}_{u_i}(\mu_i,\mathbf{X}) = \mathsf{L}_{\mathsf{u}_i}(\mu_i,\mathbf{X})$. Similarly, the node $y_i \in Y$ returns $\mathcal{L}_{y_i}(\upsilon_i,\mathbf{X}) = \mathsf{L}_{\mathsf{y}_i}(\upsilon_i,\mathbf{X})$ to the central server.

2. The central server calculate $\mathsf{L}_{\mathsf{w}_{i,j}}(\mu,\upsilon)$ for all i and j. The server then estimates the number of triangles as $\mathcal{A}(\langle \mathcal{L}_v(\mathbf{a})\rangle_v) = \mathsf{A}\left(\langle \mathsf{L}_\mathsf{v}(\mathbf{a}_\mathsf{v})\rangle_\mathsf{v}\right)/(n/3)^{k-3}$.

By the construction, we obtain the following lemma.

Lemma 8.3. *Given μ,υ, and $\mathbf{X}$. Let C be a non-interactive ϵ-edge differentially private K_k count of $\mathsf{G}_k^{\mu,\upsilon}(\mathbf{X})$ with error E. Then, $h(\mathsf{C})$ is a non-interactive ϵ-edge differentially private triangle count of $G^{\mu,\upsilon}(\mathbf{X})$ with error $\mathsf{E}/(n/3)^{k-3}$.*

Proof. The only information which $h(\mathsf{C})$ has from a user v is $\mathsf{L}_\mathsf{v}(\mathbf{a}_\mathsf{v})$. Since the publication of $\mathsf{L}_\mathsf{v}(\mathbf{a}_\mathsf{v})$ is ϵ-edge differentially private, we have that $h(\mathsf{C})$ is a non-interactive ϵ-edge differentially private protocol.

By Lemma 8.2 and the definition of $h(\mathsf{C})$, we obtain that:

$$\begin{aligned}
|\mathsf{C}(\mathsf{G}_k^{\mu,\upsilon}(\mathbf{X})) - K_k(\mathsf{G}_k^{\mu,\upsilon}(\mathbf{X}))| &= \mathsf{E},\\
|(n/3)^{k-3}\cdot h(\mathsf{C})(G^{\mu,\upsilon}(\mathbf{X})) - (n/3)^{k-3}\cdot K_3(\mathsf{G}_k^{\mu,\upsilon}(\mathbf{X}))| &= \mathsf{E},\\
|h(\mathsf{C})(G^{\mu,\upsilon}(\mathbf{X})) - K_3(\mathsf{G}_k^{\mu,\upsilon}(\mathbf{X}))| &= \mathsf{E}/(n/3)^{k-3}.
\end{aligned}$$

□

Main Result

We are now ready to prove our main result.

Theorem 8.4. *Let $\mathsf{n} = \frac{k\cdot n}{3}$ be the number of nodes in the graph $\mathsf{G}_k^{\mu,\upsilon}(\mathbf{X})$. There exists no non-interactive ϵ-differentially private algorithm which can estimate the K_k count in the graph $\mathsf{G}_k^{\mu,\upsilon}(\mathbf{X})$ with expected ℓ_2-error in $o(\mathsf{n}^{k-1})$.*

Proof. Consider the contradictory statement that there exists a non-interactive ϵ-differentially private algorithm capable of estimating the K_k count in the graph $\mathsf{G}_k^{\mu,\upsilon}(\mathbf{X})$ with an expected ℓ_2-error of $o(\mathsf{n}^{k-1})$. According to Lemma 8.3, we can construct $h(\mathsf{C})$, which represents the triangle count in the graph $\mathsf{G}_k^{\mu,\upsilon}(\mathbf{X})$, with an expected ℓ_2-error of $o(\mathsf{n}^{k-1}/(n/3)^{k-3}) = o((k\cdot n/3)^{k-1}/(n/3)^{k-3}) = o(n^2)$. This implies that

$$\mathbb{E}\left[|h(\mathsf{C})(G^{\mu,\upsilon}(\mathbf{X})) - K_3(\mathsf{G}_k^{\mu,\upsilon}(\mathbf{X}))|^2\right] = o(n^4).$$

We can use the Jensen's inequality to show that

$$\mathbb{E}\left[|h(\mathsf{C})(G^{\mu,\upsilon}(\mathbf{X})) - K_3(\mathsf{G}_k^{\mu,\upsilon}(\mathbf{X}))|\right] = o(n^2).$$

Let $\gamma = 10^{-7}$. We know that, for large n,

$$\mathbb{E}\left[|h(\mathsf{C})(G^{\mu,\upsilon}(\mathbf{X})) - K_3(\mathsf{G}_k^{\mu,\upsilon}(\mathbf{X}))|\right] < \gamma\cdot n^2.$$

Using Markov's inequality, we obtain that

$$\Pr\left[|h(\mathsf{C})(G^{\mu,\upsilon}(\mathbf{X})) - K_3(\mathsf{G}_k^{\mu,\upsilon}(\mathbf{X}))| \geq \frac{n^2}{12}\right] < 12 \cdot \gamma.$$

Hence,

$$\Pr\left[|h(\mathsf{C})(G^{\mu,\upsilon}(\mathbf{X})) - K_3(\mathsf{G}_k^{\mu,\upsilon}(\mathbf{X}))| \leq \frac{n^2}{12}\right] > 1 - 12 \cdot \gamma,$$

$$\mathbb{E}\left[\left|\left\{t \in \{1,\ldots,\mathsf{N}\} : \left|h(\mathsf{C})(\mu^{(t)}, \nu^{(t)}, \mathbf{X}) - K_3\left(G^{\mu^{(t)},\nu^{(t)}}(\mathbf{X})\right)\right| \leq \frac{n^2}{12}\right\}\right|\right] > (1 - 12 \cdot \gamma)\mathsf{N}.$$

By the Chernoff's bound, we obtain that:

$$\Pr\left[\left|\left\{t \in \{1,\ldots,\mathsf{N}\} : \left|h(\mathsf{C})(\mu^{(t)}, \nu^{(t)}, \mathbf{X}) - K_3\left(G^{\mu^{(t)},\nu^{(t)}}(\mathbf{X})\right)\right| \leq \frac{n^2}{12}\right\}\right| \leq \frac{\mathsf{N}}{5184}\right] < e^{-2((1-12\gamma-\frac{1}{5184})\mathsf{N})^2/\mathsf{N}} < 0.36.$$

Thus,

$$\Pr\left[\left|\left\{t \in \{1,\ldots,\mathsf{N}\} : \left|h(\mathsf{C})(\mu^{(t)}, \nu^{(t)}, \mathbf{X}) - K_3\left(G^{\mu^{(t)},\nu^{(t)}}(\mathbf{X})\right)\right| \leq \frac{n^2}{12}\right\}\right| \geq \frac{\mathsf{N}}{5184}\right] > 0.64.$$

Since $h(\mathsf{C})$ can be regarded as $\mathcal{D}_{\mathsf{C}}$, this leads to a contradiction with Theorem 8.1. $\square$