Counting Subgraphs under Shuffle Differential Privacy

Juanru Fang*
The University of Hong Kong
Hong Kong, Hong Kong
jrfang@cs.hku.hk

Ke Yi
Hong Kong University of Science and Technology
Hong Kong, Hong Kong
yike@cse.ust.hk

Abstract

To understand the complex structures and relationships in graph data while safeguarding personal privacy, subgraph counting under differential privacy (DP) has received a lot of attention recently. The problem is particularly important in a distributed setting, where each node holds only its local neighboring information and the analyst is untrusted. In the literature, two DP models are tailored for this scenario, known as local DP and shuffle DP, whereas the latter is equipped with a trusted shuffler that random shuffles the messages before handing them to the analyst. Since the shuffler introduces no additional privacy risk, any local DP protocol automatically satisfies shuffle DP, and the key question is whether shuffle DP can offer any improvement, especially for utility. While positive results have been obtained for a number of basic problems, such as basic counting, frequency estimation, and distinct count, it still remains elusive if this is the case for any graph problem. In this paper, we advance the understanding of this question by presenting new shuffle DP protocols for counting various subgraphs, including triangles, 4-cycles, and 3-hop paths, which improve upon the existing local DP and shuffle DP protocols, both asymptotically and concretely.

CCS Concepts

• **Security and privacy → Database and storage security**.

Keywords

Differential privacy; Subgraph counting

ACM Reference Format:
Juanru Fang and Ke Yi. 2025. Counting Subgraphs under Shuffle Differential Privacy. In *Proceedings of the 2025 ACM SIGSAC Conference on Computer and Communications Security (CCS '25), October 13–17, 2025, Taipei, Taiwan.* ACM, New York, NY, USA, 17 pages. https://doi.org/10.1145/3719027.3765047

1 Introduction

Subgraph counting is essential to understanding the structures and relationships in graph data. By counting the occurrence of specific subgraphs, researchers can analyze clustering tendencies [27], identify communities [28], predict links [11], etc. However, disclosing the exact counts of these subgraphs may reveal sensitive personal information.

Differential privacy (DP) [10] has become the benchmark for personal privacy. The central model of DP assumes a trusted data curator who possesses the entire graph and publishes privatized subgraph counts [2, 6–9, 20, 22, 32], which is a strong requirement that is not met in many distributed scenarios. The local and shuffle models of DP have thus attracted much attention recently. In these two models, each node only possesses its local neighbor information, i.e., its adjacency list, and privatizes this information on its own and sends privatized messages to an untrusted analyst. The difference is that in local DP, the analyst knows who sent which message, but in shuffle DP, they do not. The latter is equivalent to performing a random shuffle of the messages before handing them to the analyst, hence the name "shuffle DP".

Since the shuffler only applies a random permutation to the messages without modifying their contents, it introduces no additional privacy risk. The shuffle model mechanism thus inherits the privacy guarantees of the local randomizers, and any local DP protocol automatically satisfies shuffle DP [30]. Then an interesting question is if shuffle DP can offer better utility. Positive answers have been obtained for many fundamental problems. For example, the error for the bit counting problem (each user has a bit and the goal is to estimate the number of 1's) is[1] $\tilde{\Theta}(\sqrt{n})$ under local DP [3], while it is possible to achieve $\tilde{O}(1)$ error under shuffle DP [14]; similar improvements have also been obtained for problems like real summation [15], frequency estimation [13, 26], distinct count [5], etc. However, it still remains an open problem whether shuffle DP can do better for any graph problems. In this paper, we advance the understanding of this question by showing that for many subgraph counting problems, shuffle DP indeed allows us to obtain better error bounds than the best known results under local DP.

1.1 Our Results

We recognize a key construct in subgraph counting, which we call *k-stars* (see Section 3.3 for a formal definition). We then develop shuffle DP protocols to count such k-stars, and show how they can be used to count a variety of subgraphs with improved accuracy. Table 1 summarizes our results on some of the patterns that can be supported, in comparison with the best existing results in shuffle DP and local DP. The protocols are compared in terms of utility (i.e., the variance of the estimator, noting that the bias is 0 for all the estimators), the communication cost per node, and the analyst's running time, where n is the number of nodes of the graph and d is the degree upper bound.

In Figure 1, we further illustrate the asymptotic bounds on the variance as d varies from 1 (very sparse graphs) to n (very dense

*This work was completed while the author was at Hong Kong University of Science and Technology.

CCS '25, Taipei, Taiwan.

ACM ISBN 979-8-4007-1525-9/2025/10
https://doi.org/10.1145/3719027.3765047

[1]The $\tilde{O}(\cdot)$ or $\tilde{\Theta}(\cdot)$ notation hides logarithmic factors and the dependency on ε.

Table 1: Subgraph counting protocols under shuffle DP and local DP

Count	Our Results			Existing Results							
	Shuffle DP			Shuffle DP				Local DP			
	Variance	Comm.	Time	Variance	Comm.	Time	Ref	Variance	Comm.	Time	Ref
$C_{\triangle}$	$\tilde{O}(n^2d + nd^3)$	$\tilde{O}(n + d^{1.5})$	$\tilde{O}(n^2 + nd^{1.5})$	$\tilde{O}(n^3d^2)$	$\tilde{O}(n)$	$\tilde{O}(n^2)$	[19]	$O(n^3 + nd^3)$	$O(n)$	$O(n^3)$	[12]
$C_{\square}$	$\tilde{O}(n^2d^2 + nd^{4.5})$	$\tilde{O}(n + d^{1.5})$	$\tilde{O}(n^2 + nd^{1.5})$	$\tilde{O}(n^3d^2 + n^2d^6)$	$\tilde{O}(n)$	$\tilde{O}(n^2)$	[19]	$O(n^4 + nd^5)$	$O(n)$	$O(n^4)$	Appendix
$C_{\sqcup}$	$\tilde{O}(nd^{4.5})$	$\tilde{O}(d^{1.5})$	$\tilde{O}(nd^{1.5})$	-	-	-	-	$O(n^4 + n^2d^4)$	$O(n)$	$O(n^4)$	Appendix

graphs). Note that both axes are in logarithmic scale, so a polynomial function becomes a line. For example, our variance bound for triangle counting is $O(n^2d + nd^3)$. This becomes a piecewise-linear curve: it is $O(n^2d)$ when $d \le \sqrt{n}$ and $O(nd^3)$ when $d > \sqrt{n}$ otherwise.

Triangles are the most important subgraph pattern, and have received the most attention. However, the best utility so far is still achieved by a very simple local DP protocol [12], which just applies randomized response to every edge, although it requires $O(n^3)$ time to compute the estimated count. The shuffle DP protocol [19] reduces the computation time to $\tilde{O}(n^2)$, but the utility has been made worse for the entire range of d. On the other hand, our new shuffle-DP protocol improves upon the local DP protocol in terms of both utility and time when $d = o(n^{2/3})$; see the first row of Table 1 as well as Figure 1.

For 4-cycles, the local DP protocol using randomized response also works, though its variance has not been analyzed. We give an analysis in the appendix, and show the result in the second row of Table 1. The shuffle DP protocol [19] has a better utility when $d = o(n^{1/3})$ and worse otherwise, while our new protocol improves over the entire range of d (see Figure 1) with less analyst's time. Similar improvements have also been obtained for 3-hop paths. Our protocol can also handle many other subgraphs; please see Section 4.4 for details.

Besides asymptotic improvements, our protocols also have good concrete performance. In Section 5, we present the experimental results on counting triangles, 4-cycles, and 3-hop paths using real-world graphs. The results indicate that our protocols significantly outperform the existing shuffle DP and local DP protocols. We also show that one can adjust the trade-off between utility and communication/computation costs of our protocols by sampling.

Limitations. Note that, although we have obtained better shuffle-DP protocols for many subgraph counting problems than the best known results under local DP, whether there is a separation between shuffle-DP and local-DP for subgraph counting remains elusive, which would need stronger lower bounds under local-DP for small d (the lower bound in [12] only holds for $d = \Theta(n)$).

1.2 Related Work

In this paper, we focus on the standard, one-round model of local DP and shuffle DP. While one-round local DP mechanisms for counting k-stars and triangles are well-studied in [12, 17], with [12] achieving state-of-the-art accuracy for triangles and offering broader subgraph extensibility, the landscape for shuffle DP is less explored. To our knowledge, [19] is the sole prior work under one-round shuffle DP. The mechanism counts triangles and 4-cycles based on wedge counts estimated using randomized response. Although both [19] and our protocols share some underlying primitives, our approach improves in the following aspects. First, our protocol is built on k-star frequencies, which is more general than wedge counts in the sense that wedge is a special case of k-star with $k = 2$. This enables us to count additional subgraphs, such as 3-hop paths. Second, we have combined sampling with k-star counting, and managed to improve the variance bounds by carefully setting the sampling probability. Finally, for wedge counting itself, we achieve better accuracy. In [19], a logarithmic term in the variance arises from amplification by shuffling. We have removed this term by the negative binomial mechanism. Multi-round protocols have also been studied under local DP [16–18], and often achieve better utility. For example, the very recent two-round local DP protocol [16] achieves a variance of $O(nd^3)$ for triangle counting, which is better than our result when $d = o(\sqrt{n})$. However, multi-round protocols incur larger latency, higher communication costs, and are more complicated to implement (e.g., they require synchronization among the users). Besides, [25] assumes that the graphs satisfy a certain structure; thus, it only performs well for certain types of graphs. [24] and [29] instead count subgraphs based on the assumption that each user has an extended view, e.g., their 2-hop neighbors, which requires the stronger assumption that neighbors are trusted.

So far, all works under local and shuffle DP on graph problems adopt the edge-DP policy, i.e., the presence of any edge cannot be learned by the adversary. In central-DP, a stronger policy has also been studied, known as node-DP, which protects the presence of any node and all its incident edges [2, 6, 7, 22]. It is an interesting open problem if this stronger DP policy can be supported in local or shuffle DP.

Finally, it is worth pointing out that we work with the standard DP definition, which provides information-theoretical privacy guarantees, i.e., the adversary is allowed to have unlimited computing power. By weakening the guarantee to a computational one and combining with secure multi-party computation (MPC) techniques, it is possible to further reduce the variance of triangle counting at the expense of higher computational costs and more rounds of communication [23].

2 Preliminaries

Let $[n] := \{1, 2, \ldots, n\}$. Suppose there are n users $U = \{u_1, \ldots, u_n\}$, where user u_i holds data $x_i \in \mathcal{X}$ for $i \in [n]$, and they collectively constitute the *instance* $I = \{x_i\}_{i\in[n]}$. We use $I \sim I'$ to denote that the instances I and I' are neighbors, which shall be defined more precisely later. The distance between I and I', denoted by $\mathrm{dist}(I, I')$, is the length of the shortest sequence $(I_0 = I, I_1, \ldots, I_k = I')$ such that $I_{i-1} \sim I_i$ for all $i \in [k]$.

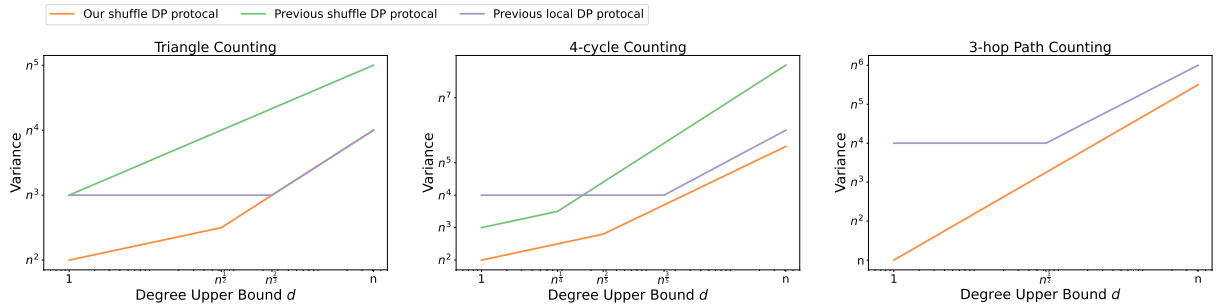

Figure 1: Asymptotic variance bounds of subgraph counting protocols under shuffle DP and local DP as d varies.

Given an output range $\mathcal{Z}$, let $\mathbb{P}(\mathcal{Z})$ denote the set of probability distributions over $\mathcal{Z}$. For any distribution $\mathcal{P}$, we write $z \sim \mathcal{P}$ to denote a random variable z that is distributed as $\mathcal{P}$. A mechanism $\mathcal{M} : \mathcal{X}^n \to \mathbb{P}(\mathcal{Z})$ maps each instance $I \in \mathcal{X}^n$ to the distribution $\mathcal{M}(I)$, and outputs a random variable $z \sim \mathcal{M}(I)$.

Definition 2.1 (Differential Privacy (DP) [10])**.** A mechanism $\mathcal{M} : \mathcal{X}^n \to \mathbb{P}(\mathcal{Z})$ satisfies (ε, δ)-DP if for any neighboring instances $I \sim I'$ and any measurable subset $Z \subseteq \mathcal{Z}$,

$$\int_Z \mathcal{M}(I)(z)dz \le e^{\varepsilon} \cdot \int_Z \mathcal{M}(I')(z)dz + \delta.$$

We call ε the privacy budget. Moreover, δ should be negligibly small for the mechanism to protect privacy. When $\delta = 0$, we say that the mechanism $\mathcal{M}$ satisfies pure ε-DP.

Lemma 2.2 (The Basic Composition Theorem [10])**.** Given (ε, δ)-DP mechanisms $\mathcal{M}_1, ..., \mathcal{M}_m$, if for any neighboring instance $I \sim I'$, there exists at most k mechanisms $\mathcal{M}_i$ such that $\mathcal{M}_i(I) \neq \mathcal{M}_i(I')$, then the mechanism $\mathcal{M} = (\mathcal{M}_1, ..., \mathcal{M}_m)$ satisfies $(k\varepsilon, k\delta)$-DP.

Lemma 2.3 (The Advanced Composition Theorem [10])**.** Given (ε, δ)-DP mechanisms $\mathcal{M}_1, ..., \mathcal{M}_m$ and some δ_0, if for any neighboring instance $I \sim I'$, there exists at most k mechanisms $\mathcal{M}_i$ such that $\mathcal{M}_i(I) \neq \mathcal{M}_i(I')$, then the mechanism $\mathcal{M} = (\mathcal{M}_1, ..., \mathcal{M}_m)$ satisfies $(\varepsilon', k\delta + \delta_0)$-DP for

$$\varepsilon' = \sqrt{2k \ln(\frac{1}{\delta_0})}\varepsilon + k\varepsilon(e^{\varepsilon} - 1).$$

Lemma 2.4 (Post Processing [10])**.** Given a (probably randomized) function f, if $\mathcal{M}$ satisfies (ε, δ)-DP, then $f \circ \mathcal{M}$ also satisfies (ε, δ)-DP.

Lemma 2.5 (Group Privacy [10])**.** Given an (ε, δ)-DP mechanism $\mathcal{M}$, for any instances I and I' such that $\text{dist}(I, I') \le k$ and any measurable subset $Z \subseteq \mathcal{Z}$,

$$\int_Z \mathcal{M}(I)(z)dz \le e^{k\varepsilon} \cdot \int_Z \mathcal{M}(I')(z)dz + ke^{k\varepsilon}\delta.$$

2.1 Central DP, Local DP and Shuffle DP

There are three common DP models: central DP [10], local DP [21], and shuffle DP [30]. The mechanism $\mathcal{M}$ in the central DP model can be an arbitrary function from $\mathcal{X}^n$ to $\mathbb{P}(\mathcal{Z})$, modeling the scenario where a trusted data curator runs $\mathcal{M}$ on the entire instance I. In the local DP and shuffle DP model, each user $u_i \in U$ runs a local randomizer $\mathcal{R} : \mathcal{X} \to \mathbb{P}(\mathcal{Y})$ on their own data x_i and sends $\mathcal{R}(x_i)$ to the untrusted data analyst. More formally, the mechanism $\mathcal{M}$ in the local DP model must take the form

$$\mathcal{M}(I) := (\mathcal{R}(x_1), \ldots, \mathcal{R}(x_n)),$$

and the joint probability distribution of the n local randomizers should satisfy Definition 2.1. The mechanism in the shuffle DP model contains an additional shuffler $\mathcal{S}$ and

$$\mathcal{M}(I) := \mathcal{S}(\mathcal{R}(x_1), \ldots, \mathcal{R}(x_n)).$$

Here, the shuffler $\mathcal{S}$ performs a random shuffle of all the messages before passing them to the analyst, preventing the analyst from identifying the sender of each message. According to the post-processing property, if the local randomizers satisfy DP, then shuffling their outputs still satisfies DP, so any local-DP protocol is also a shuffle-DP protocol. Besides, in the local DP and shuffle DP model, the analyst often runs another function $\mathcal{A}$ to compute the final result $\mathcal{A}(\mathcal{M}(I))$.

2.2 DP in Graphs

While different DP models define the architectural roles by specifying what the analyst and users can observe, it remains to specify the *neighboring* relationship, i.e., what constitutes the sensitive information to be protected, which is specified by *DP policies*. For graph data, the most common DP policies are edge-DP and node-DP.

For graph problems, an instance I is an undirected graph $I = (U = V, E)$ on the n users (nodes), where the data held by u_i is their adjacency vector, i.e., a bit vector x_i such that $x_{i,j} = 1$ if there is an edge between u_i and u_j, and $x_{i,j} = 0$ otherwise. Moreover, we use d_i to denote the degree of node u_i, and assume that $d_i \le d$, where d is a given degree upper bound.

In this paper we focus on edge-DP, as with all prior work in the local and shuffle DP model. Under edge-DP, two instances I and I' are neighbors if they differ by one edge. More precisely, $I = (U, E)$ and $I' = (U, E')$ are on the same set of users, while there exist $i^*, j^* \in [n]$ such that

$$(E \setminus E') \cup (E' \setminus E) = \{\{u_{i^*}, u_{j^*}\}\}.$$

Note that this means the adjacency vectors of I and I' differ by exactly two bits.

Consequently, the *edge local DP* model [12] combines the local DP model with the edge DP policy:

Definition 2.6 (Edge Local DP)**.** Let $\mathcal{R} : \{0,1\}^n \to \mathcal{Y}$ be a local randomizer, and let

$$\mathcal{M}(I) = (\mathcal{R}(x_1), \ldots, \mathcal{R}(x_n))$$

on a graph instance I with adjacency vectors $x_i, i \in [n]$. The mechanism $\mathcal{M}$ satisfies (ϵ, δ)-edge local DP if for any neighboring instances I and I' that differ by one edge, and any measurable subset $Z \subseteq \mathcal{Z}$, the following inequality holds:

$$\int_Z \mathcal{M}(I)(z)dz \le e^{\varepsilon} \cdot \int_Z \mathcal{M}(I')(z)dz + \delta.$$

The edge local DP model in [17, 18] instead defines neighboring instances as adjacency matrices differing by one bit, and Definition 2.6 aligns with "relationship DP" in [17, 18]. These definitions differ by a factor of 2 in distance. By group privacy, any mechanism satisfying (ε, δ)-DP under edge local DP in [17, 18] satisfies $(2\varepsilon, 2e^{\varepsilon}\delta)$-DP under Definition 2.6.

Similarly, *edge shuffle DP* [19] combines the shuffle DP model with the same edge DP policy:

Definition 2.7 (Edge Shuffle DP)**.** Let $\mathcal{R} : \{0,1\}^n \to \mathcal{Y}$ be a local randomizer, and let

$$\mathcal{M}(I) = (\mathcal{R}(x_1), \ldots, \mathcal{R}(x_n))$$

on a graph instance I with adjacency vectors $x_i, i \in [n]$. The mechanism $\mathcal{M}$ satisfies (ϵ, δ)-edge shuffle DP if for any neighboring instances I and I' that differ by one edge, and any measurable subset $Z \subseteq \mathcal{Z}$, the following inequality holds:

$$\int_Z \mathcal{M}(I)(z)dz \le e^{\varepsilon} \cdot \int_Z \mathcal{M}(I')(z)dz + \delta.$$

2.3 DP Mechanisms

In this work, we use $\mathrm{Ber}(q)$ to denote the Bernoulli distribution with success probability q, and $\mathrm{DLap}(b)$ to denote the discrete Laplace distribution with scale b.[2] Given a discrete Laplace distribution $\mathrm{DLap}(b)$, the variance is $\mathrm{Var}[\mathrm{DLap}(b)] = O(b^2)$, and with probability at least $1-\beta$,

$$|\mathrm{DLap}(b)| \le b \ln\left(\frac{1}{\beta}\right).$$

The first local randomizer is the discrete Laplace mechanism $\mathcal{R}_{\mathrm{DL}}$ [4]. Given a function $f : \mathcal{X} \to \mathcal{Y}$, let

$$\mathrm{GS}_f = \max_{I \sim I'} \max_{i \in [n]} |f(x_i) - f(x_i')|$$

denote the global sensitivity of the function f, the discrete Laplace mechanism aims to estimate $f(x)$ under local DP. The local randomizer is shown in Algorithm 1, and the analyzer $\mathcal{A}_{\mathrm{DL}}$ computes $\tilde{f}(x) = y$ as a private estimate of $f(x)$.

Lemma 2.8. For any function f, the mechanism $\mathcal{R}_{\mathrm{DL}}$ satisfies pure 2ε-edge local DP. Besides, if for every pair of neighboring instances $I \sim I'$, the function f has the property that at most one i^* satisfies $f(x_{i^*}) \neq f(x'_{i^*})$, then $\mathcal{R}_{\mathrm{DL}}$ satisfies pure ε-local DP. For any data $x \in \mathcal{X}$, $\tilde{f}(x)$ is unbiased and has a variance of $O(\frac{\mathrm{GS}_f^2}{\varepsilon^2})$.

[2] The probability density functions of the discrete Laplace distribution is $\frac{e^{\frac{1}{b}}-1}{e^{\frac{1}{b}}+1} e^{-\frac{|x|}{b}}$.

Algorithm 1: The Local Randomizer $\mathcal{R}_{\mathrm{DL}}$

Input : The data $x \in \mathcal{X}$, the function $f : \mathcal{X} \to \mathcal{Y}$, the global sensitivity GS_f, the privacy budget ε

1 send the message $y = f(x) + \mathrm{DLap}(\frac{\mathrm{GS}_f}{\varepsilon})$;

It is noted that we analyze two distinct DP guarantees: a general privacy guarantee applicable to arbitrary functions under edge DP, and a tighter privacy guarantee achievable only in the special case where the computed function $f(x_i)$ differs in only one i, which enables enhanced privacy. Both guarantees will be formally established for all DP mechanisms in this subsection and leveraged accordingly in the subsequent sections.

The second local randomizer is Warner's randomized response $\mathcal{R}_{\mathrm{RR}}$ [31]. Given a function $f : \mathcal{X} \to \{0,1\}$, Warner's randomized response aims to estimate $f(x)$ under local DP. The local randomizer $\mathcal{R}_{\mathrm{RR}}$ is shown in Algorithm 2 and the analyzer $\mathcal{A}_{\mathrm{RR}}$ finally computes

$$\tilde{f}(x) = \frac{y \cdot (e^{\varepsilon}+1) - 1}{e^{\varepsilon} - 1}.$$

Algorithm 2: The Local Randomizer $\mathcal{R}_{\mathrm{RR}}$

Input : The data $x \in \mathcal{X}$, the function $f : \mathcal{X} \to \{0,1\}$, the privacy budget ε

1 sample $z \sim \mathrm{Ber}(\frac{e^{\varepsilon}}{e^{\varepsilon}+1})$;
2 **if** $z = 1$ **then** send the message $y = f(x)$;
3 **else** send the message $y = 1 - f(x)$;

Lemma 2.9. For any function f, the mechanism $\mathcal{R}_{\mathrm{RR}}$ satisfies pure 2ε-edge local DP. Besides, if for every pair of neighboring instances $I \sim I'$, the function f has the property that at most one i^* satisfies $f(x_{i^*}) \neq f(x'_{i^*})$, then $\mathcal{R}_{\mathrm{RR}}$ satisfies pure ε-local DP. For any data $x \in \mathcal{X}$, $\tilde{f}(x)$ is unbiased and has a variance of $\frac{e^{\varepsilon}}{(e^{\varepsilon}-1)^2} = O(\frac{1}{\varepsilon^2})$.

The last local randomizer is the negative binomial mechanism $\mathcal{R}_{\mathrm{NB}}$ [15]. Given a function $f : \mathcal{X} \to \{0\} \cup [\Delta]$ for some positive integer Δ, the mechanism aims to estimate the sum

$$f(I) = \sum_{i \in [n]} f(x_i)$$

under shuffle DP. The general framework of the local randomizer $\mathcal{R}_{\mathrm{NB}}$ is shown in Algorithm 3. Let Y denote the multiset of all messages sent by the users (after shuffling), the analyzer $\mathcal{A}_{\mathrm{NB}}$ then computes

$$\tilde{f}(I) = \sum_{y \in Y} y$$

as a private estimate of the sum $f(I)$.

We see that the negative binomial mechanism sends three types of messages. First, the message $f(x)$ is sent directly in Line 2. Second, a series of messages of 1 or -1 are sent in Line 5. The numbers of such messages are drawn from a distribution $\mathcal{P}^{\mathrm{cent}}$, which shall be specified later, such that $\sum_{i \in [n]} \left(z_{+1}^{(i)} - z_{-1}^{(i)}\right)$ follows $\mathrm{DLap}(\frac{\Delta}{\varepsilon})$, thereby obscuring the total sum. Finally, a series of messages are sent in Line 9, where each message takes a value $t \in [-\Delta, \Delta]$. The

Algorithm 3: The Local Randomizer $\mathcal{R}_{\mathrm{NB}}$

Input : The data $x \in \mathcal{X}$, the function $f : \mathcal{X} \to \{0\} \cup [\Delta]$, the number of user n, the parameter Δ, the privacy budget ε and δ
if $f(x) \neq 0$ **then** send a message $f(x)$;
compute the collection of sets $\mathcal{T}$ and distributions $\mathcal{P}^{\mathrm{cent}}$ and $\{\mathcal{P}^T\}_{T\in\mathcal{T}}$ using ε, δ, Δ and n;
sample $z_{+1} \sim \mathcal{P}^{\mathrm{cent}}, z_{-1} \sim \mathcal{P}^{\mathrm{cent}}$;
send z_{+1} messages, where each message is 1, and send z_{-1} messages, where each message is -1;
for $T \in \mathcal{T}$ **do**
  sample $z_T \sim \mathcal{P}^T$;
  for $t \in T, t \neq 0$ **do**
    send z_T messages, where each message is t;
  end
end

numbers of such messages are drawn from a collection of distributions $\mathcal{P}^T, T \in \mathcal{T}$, which will also be specified later. These messages mask the difference in the count of messages for each value in $[-\Delta, \Delta]$, while ensuring that for any $T \in \mathcal{T}$, $\sum_{t\in T} t = 0$. Combining these, we can demonstrate that the sum of all the messages follows

$$\tilde{f}(I) = f(I) + \mathrm{DLap}(\frac{\Delta}{\varepsilon}).$$

It is important to emphasize that users do not add direct noise to $f(x)$. Instead, they send additional messages with varying values to conceal both the total sum and the count of messages for each distinct value, thereby achieving differential privacy. In the following, we present only the results. Interested readers can refer to [15] for the proofs.

Lemma 2.10 ([15]). For any given privacy budget $\varepsilon < 4$, if for every pair of neighboring instances $I \sim I'$, the function f has the property that at most one node v_{i^*} satisfies $f(x_{i^*}) \neq f(x'_{i^*})$, then the local randomizer $\mathcal{R}_{\mathrm{NB}}$ satisfies (ε, δ)-shuffle DP. Moreover, for any given privacy budget $\varepsilon < 4$ and any function f, $\mathcal{R}_{\mathrm{NB}}$ satisfies $(2\varepsilon, 2e^{2\varepsilon}\delta)$-edge shuffle DP according to the group privacy property. For any instance $I \in \mathcal{I}$, $\tilde{f}(I)$ is unbiased and has a variance of $\mathrm{Var}[\mathrm{DLap}(\frac{\Delta}{\varepsilon})] = O(\frac{\Delta^2}{\varepsilon^2})$. The communication cost is $\tilde{O}(\mathbb{I}[f(x_i) \neq 0] + \frac{\Delta}{n\varepsilon})$ bits for user u_i in expectation.[3]

Finally, we describe the distributions used in the negative binomial mechanism. Let $\mathrm{NB}(r, p)$ denote the negative binomial distribution (extended to real $r > 0$ via the gamma function) with number of successes r and success probability p. Moreover, let $\mathrm{NB}_{/n}(r, p) := \mathrm{NB}(r/n, p)$, so that

$$\sum_{i\in[n]} \mathrm{NB}_{/n}(r, p) = \mathrm{NB}(r, p).$$

We will only describe the distributions for the case $\Delta = 1$, which is the case used in this work; the general case $\Delta > 1$ is more complicated and interested readers are referred to [15] for details. Given privacy parameters ε and δ, we set $\varepsilon_1 = \frac{3\varepsilon}{4}$, $\varepsilon_2 = \frac{\varepsilon}{20}$, and have

$$\mathcal{T} = \{\{+1, -1\}\},$$

$$\mathcal{P}^{\mathrm{cent}} = \mathrm{NB}_{/n}(1, 1 - e^{-\varepsilon_1}),$$

$$\mathcal{P}^{\{+1,-1\}} = \mathrm{NB}_{/n}(3(1 + \log(\frac{1}{\delta})), 1 - e^{-\varepsilon_2}).$$

It is noted that

$$\mathrm{NB}(1, 1 - e^{-\varepsilon_1}) - \mathrm{NB}(1, 1 - e^{-\varepsilon_1}) = \mathrm{DLap}\left(\frac{1}{\varepsilon_1}\right),$$

thus, we can conclude that a sequence of messages sampled from $\mathcal{P}^{\mathrm{cent}}$ collectively generates the discrete Laplace noise $\mathrm{DLap}(1/\varepsilon_1)$ to obscure the total sum, while messages sampled from $\mathcal{P}^{+1,-1}$ mask counts and correlations between +1 and −1 messages.

Table 2: Summary of notations

ε, δ	privacy parameters in (ε, δ)-DP
U, u_i, n	U: user set; u_i: i-th user in U; n: number of users
$\mathcal{X}, x_i$	$\mathcal{X}$: universe of data; x_i: u_i's data
$\mathcal{X}^n, I = (U = V, E)$	$\mathcal{X}^n$: universe of instances, I: instance in $\mathcal{X}^n$, an undirected graph with set of nodes $U = V$ and set of edges E
$V, \tilde{V}, v_i$	V: set of nodes; $\tilde{V}$: subset of nodes; v_i: i-th node in V
d, d_i	d: given degree upper bound; d_i: degree of v_i
Y, y	Y: multiset of messages; y: a message
$\mathcal{Z}, z, \mathbb{P}(\mathcal{Z})$	$\mathcal{Z}$: output range of the mechanisms; z: an output; $\mathbb{P}(\mathcal{Z})$: set of probability distributions over $\mathcal{Z}$
$\mathcal{M}, \mathcal{R}, \mathcal{S}, \mathcal{A}$	$\mathcal{M}$: mechanism; $\mathcal{R}$: local randomizer; $\mathcal{S}$: shuffler; $\mathcal{A}$: analyzer
$\mathcal{Q}, q$	$\mathcal{Q}$: sampling procedure; q: sampling probability
$f, f_{\mathbb{I}}, f_{\mathrm{deg}}, f_j, f_{T,k}$	f: generic function; $f_{\mathbb{I}}$: identity function; f_{deg}: degree function; f_j: bit-extraction function; $f_{T,k}$: k-star counting function
$C_\triangle, C_\square, C_\sqcup$, etc.	count of triangles, 4-cycles, 3-hop paths, etc.
Δ	parameter in the negative binomial mechanism

2.4 Notation

Table 2 summarizes key symbols used throughout this work. Moreover, in the following sections, there will be a slight notational shift: when describing a graph instance $I = (U, E)$ in Section 2.2, we simply set $U = V$, treating the set of users as equivalent to the set of nodes, thus, we may refer to "v_i holding its data x_i", meaning that node v_i holds the adjacency vector x_i, instead of using the user u_i representation.

Throughout the following sections, we use i and j as indices iterating over nodes, k for the star size in k-star counting, $m < n$ as the number of groups to be analyzed (formally defined in Section 4.1.3), and ℓ as an index iterating over groups.

3 Main Building Blocks

In this section, we introduce the main building blocks of our mechanism. For completeness, we first review the local randomizers that estimate each node's degree and adjacency list. Following this, we introduce our new mechanism that estimates the frequency of specific k-stars in the graph.

[3] The parameter δ appears only in logarithmic factors of the communication cost and is therefore hidden in the $\tilde{O}(\cdot)$ notation.

3.1 Node Degree Estimation

Let f_{deg} denote the function that computes the node degree, i.e.,

$$f_{\text{deg}}(x_i) = d_i = \sum_{j\in[n], i\neq j} x_{i,j}.$$

The first local randomizer $\mathcal{R}_{\text{deg}}$, as shown in Algorithm 4, obfuscates node degrees by invoking the discrete Laplace mechanism. For any node $v_i \in V$, the analyzer finally computes $\tilde{d}_i = y_i$.

Algorithm 4: The Local Randomizer $\mathcal{R}_{\text{deg}}$

Input : The data $x_i \in \mathcal{X}$ and the privacy budget ε

invoke the local randomizer $y_i = \mathcal{R}_{\text{DL}}(x_i, f_{\text{deg}}, 1, \frac{\varepsilon}{2})$;

Lemma 3.1. For any given privacy budget ε, the local randomizer $\mathcal{R}_{\text{deg}}$ satisfies ε-edge local DP. For any node $v_i \in V$,

$$\mathbb{E}[\tilde{d}_i] = d_i, \quad \text{Var}[\tilde{d}_i] = O\left(\frac{1}{\varepsilon^2}\right).$$

Proof. Consider any neighboring instances $I \sim I'$ that differ by the edge (v_{i^*}, v_{j^*}). In this case, the degrees of nodes v_{i^*} and v_{j^*} differ by at most 1. The proof then follows the basic composition theorem and Lemma 2.8. □

3.2 Adjacent Matrix Estimation

Given the data $x_i \in \mathcal{X} = \{0,1\}^n$, for any $j \in [n]$, let $f_j : \mathcal{X} \to \{0,1\}$ be the function that returns the j-th bit of x_i, i.e.,

$$f_j(x_i) = x_{i,j}.$$

The local randomizer $\mathcal{R}_{\text{adj}}$, as shown in Algorithm 5, then obfuscates the adjacent matrix by invoking Warner's Randomized Response. For any pair of nodes $v_i, v_j \in V, i < j$, the analyzer $\mathcal{A}_{\text{adj}}$ computes

$$\tilde{x}_{i,j} = \frac{y_{i,j}\cdot(e^\varepsilon+1)-1}{e^\varepsilon-1}.$$

Here, only the upper triangle of the adjacency matrix is obfuscated, and the lower part can be reconstructed using symmetry.

Algorithm 5: The Local Randomizer $\mathcal{R}_{\text{adj}}$

Input : The data $x_i \in \mathcal{X}$ and the privacy budget ε

for $j \in [n], j > i$ **do**

  invoke the local randomizer $y_{i,j} = \mathcal{R}_{\text{RR}}(x_i, f_j, \varepsilon)$;

end

Lemma 3.2. For any given privacy budget ε, the local randomizer $\mathcal{R}_{\text{adj}}$ satisfies ε-edge local DP. For any pair of nodes $v_i, v_j \in V, i < j$,

$$\mathbb{E}[\tilde{x}_{i,j}] = x_{i,j}, \quad \text{Var}[\tilde{x}_{i,j}] = O\left(\frac{1}{\varepsilon^2}\right).$$

The communication cost is $O(n)$ bits.

Proof. Consider any neighboring instances $I \sim I'$ that differ by the edge (v_{i^*}, v_{j^*}) such that $i^* < j^*$. In this case, only the output distribution of $\mathcal{R}_{\text{RR}}(x_{i^*}, f_{j^*}, \varepsilon)$ changes. The proof then follows the basic composition theorem and Lemma 2.9. □

The local DP triangle counting mechanism in [12] directly utilizes $\mathcal{R}_{\text{adj}}$. The following summarizes the result. For additional details, please refer to [12].

Theorem 3.3. Assume $\varepsilon = \Theta(1)$, there exists an ε-local DP triangle counting mechanism that obtains an unbiased estimate with a variance of $O(n^3 + nd^3)$, while the analysis time is $O(n^3)$.

3.3 k-star Frequency Estimation

The last local randomizer is our key building block. To begin with, we first define the k-stars in a graph as follows: Given an undirected graph instance $I = (V, E)$, a k-star is a subgraph consisting of a central node $v_0 \in V$ and a set of k distinct nodes $v_1, \ldots, v_k \subseteq V \backslash \{v_0\}$ such that

$$\{(v_0, v_i) | i \in [k]\} \subseteq E,$$

i.e., there is an edge between v_0 and each node v_i for $i \in [k]$. Notably, when $k = 1$, this reduces to a single edge, and when $k = 2$, it forms a wedge.

3.3.1 Mechanism. Estimating the frequency of specific k-stars is exactly a sum estimation problem. In this section, we integrate the state-of-the-art negative binomial mechanism with sampling to develop a new sampling-based approach for counting k-stars with different nodes serving as the leaves.

Given a subset of nodes $T \subseteq V$ and some positive integer $k \geq |T|$, for any node $v_i \in V$, recall that d_i is the node degree of v_i, we then formally define

$$f_{T,k}(x_i) = \prod_{v_j\in T} x_{i,j} \cdot \binom{d_i - |T|}{k - |T|}$$

if $v_i \notin T$, and $f_{T,k}(x_i) = 0$ otherwise.[4] If $v_i \notin T$, then the product $\prod_{v_j\in T} x_{i,j}$, which is a 0/1 bit, indicates whether v_i is connected to all nodes in T. If it is, then there may exist k-stars with v_i being the central node and all the nodes in T being (part of) the leaves, and the number of such k-stars corresponds to the combinations of $k - |T|$ edges that can be chosen from the remaining $d_i - |T|$ edges. Otherwise, when $v_i \in T$, there cannot exist any k-star with v_i being the central node. Therefore, $f_{T,k}(x_i)$ is exactly the number of k-stars with v_i being the central node and all the nodes in T being (part of) the leaves. Finally, the number of k-stars with all the nodes in T being (part of) the leaves can be represented as

$$f_{T,k}(I) = \sum_{i\in[n]} f_{T,k}(x_i).$$

Example 3.1. Consider the graph instance I that consists of a 5-hop path $v_1 - v_2 - v_3 - v_4 - v_5$. Given $T = \{v_3\}$ and $k = 2$, then the function $f_{T,k}(x_i)$ counts the number of 2-stars with v_i being the central node and v_3 being one of the leaves. The function values for each node are computed as follows:

$$f_{T,k}(x_2) = f_{T,k}(x_4) = 1, \quad f_{T,k}(x_1) = f_{T,k}(x_3) = f_{T,k}(x_5) = 0,$$

Specifically, $f_{T,k}(x_2) = 1$ as there are a 2-star centered at v_2 with v_3 as a leaf: $v_1 - v_2 - v_3$. Similarly, $f_{T,k}(x_4) = 1$ corresponds to the 2-star $v_3 - v_4 - v_5$. Summing over all nodes yields $f_{T,k}(I) = 2$, indicating that there are two 2-stars in instance I with v_3 being one of the leaves.

[4] We define $\binom{a}{b} = 0$ for any $a < b$.

Our local randomizer $\mathcal{R}_{\text{star}}$, which obfuscates $f_{T,k}(I)$ under shuffle DP, is shown in Algorithm 6, where $f_{\mathbf{I}}$ is the identity function. More specifically, the mechanism works as follows: Each node v_i first computes $f_{T,k}(x_i)$, samples it with probability q, and then invokes the negative binomial mechanism. Let Y denote the multiset of all messages sent by the users (after shuffling), the analyzer $\mathcal{A}_{\text{star}}$ then computes $\tilde{f}_{T,k}(I) = \frac{1}{q} \cdot \sum_{y \in Y} y$.

Algorithm 6: The Local Randomizer $\mathcal{R}_{\text{star}}$

Input : The data $x_i \in \mathcal{X}$, the function $f_{T,k}$, the number of node n, the sampling probability q, the privacy budget ε and δ

$\varepsilon_q \leftarrow \ln(1 + \frac{1}{q} \cdot (e^{\varepsilon} - 1))$, $\delta_q \leftarrow \frac{1}{q} \cdot \delta$, $\Delta \leftarrow \binom{d-|T|}{k-|T|}$;

compute $f_{T,k}(x_i)$;

sample $z_i \sim \text{Ber}(q)$ and invoke the local randomizer $\mathcal{R}_{\text{NB}}(z_i \cdot f_{T,k}(x_i), f_{\mathbf{I}}, n, \Delta, \varepsilon_q, \delta_q)$;

3.3.2 Analysis. To analyze the privacy guarantees of $\mathcal{R}_{\text{star}}$, we first establish the framework of privacy amplification by sampling. Consider a data universe $\mathcal{X} = \mathbb{N}$ where datasets $I, I' \in \mathbb{N}^n$ are neighbors if they differ in exactly one data point. Let $\mathcal{Q} : \mathbb{N}^n \to \mathbb{N}^n$ denote the Poisson sampling procedure where each user's data is sampled independently with probability q, i.e., for each user u_i,

$$\mathcal{Q}(x_i) = \begin{cases} x_i & \text{with probability } q \\ 0 & \text{otherwise} \end{cases}$$

The privacy amplification result looks as follows.[5]

Lemma 3.4 (Privacy Amplification by Sampling [1])**.** Given an (ε', δ')-DP mechanism $\mathcal{M} : \mathbb{N}^n \to \mathbb{P}(\mathcal{Z})$, the mechanism $\mathcal{M} \circ \mathcal{Q} : \mathbb{N}^n \to \mathbb{P}(\mathcal{Z})$ satisfies (ε, δ)-DP for

$$\varepsilon = \ln(1 + q \cdot (e^{\varepsilon'} - 1)), \quad \delta = q\delta'.$$

Especially when $\varepsilon' < O(1)$, we have $\varepsilon = \Theta(q\varepsilon')$.

Applying this to the negative binomial mechanism then yields:

Lemma 3.5. For any given privacy budget ε and sampling probability q such that $\varepsilon_q < 4$, the local randomizer $\mathcal{R}_{\text{star}}$ satisfies $(2\varepsilon, 2e^{2\varepsilon}\delta)$-shuffle DP. Especially when $k = |T|$, the local randomzier $\mathcal{R}_{\text{star}}$ satisfies (ε, δ)-shuffle DP.

Proof. We first consider the simple case where $k = |T|$. For any neighboring instances $I \sim I'$ under edge DP, at most one $f_{T,k}(x_i)$ differs. Therefore, according to Lemma 3.4, the local randomizer $\mathcal{R}_{\text{star}}$ satisfies (ε, δ)-shuffle DP. Instead, when $k > |T|$, then at most two $f_{T,k}(x_i)$'s differs, and the proof follows Lemma 3.4 and the group privacy property. □

Lemma 3.6. For any given privacy budget ε such that $\varepsilon_q < O(1)$ and any subset of nodes $T \subseteq V$,

$$\mathbb{E}[\tilde{f}_{T,k}(I)] = f_{T,k}(I), \quad \text{Var}[\tilde{f}_{T,k}(I)] = \frac{1-q}{q} \cdot \sum_{i \in [n]} \left(f_{T,k}(x_i)\right)^2 + O\left(\frac{\Delta^2}{\varepsilon^2}\right).$$

[5] Balle et al. [1] states the results specifically for the central DP model. However, we observe that the framework and the results are also applicable to the local/shuffle DP model, given that the data not sampled is set to 0.

Proof. The expectation is

$$\mathbb{E}[\tilde{f}_{T,k}(I)] = \frac{1}{q} \cdot \left(q \cdot f_{T,k}(I) + \mathbb{E}\left[\text{DLap}\left(\frac{\Delta}{\varepsilon_q}\right)\right]\right) = f_{T,k}(I),$$

and the variance is

$$\begin{aligned}\text{Var}[\tilde{f}_{T,k}(I)] =& \frac{1}{q^2} \cdot \left(q \cdot (1-q) \cdot \sum_{i \in [n]} \left(f_{T,k}(x_i)\right)^2 + \text{Var}\left[\text{DLap}\left(\frac{\Delta}{\varepsilon_q}\right)\right]\right) \\ =& \frac{1-q}{q} \cdot \sum_{i \in [n]} \left(f_{T,k}(x_i)\right)^2 + \frac{1}{q^2} \cdot \text{Var}\left[\text{DLap}\left(\frac{\Delta}{\varepsilon_q}\right)\right]\end{aligned}$$

The proof follows that $\text{Var}[\text{DLap}(b)] = O(b^2)$ and $\varepsilon = \Theta(q\varepsilon_q)$ when $\varepsilon_q < O(1)$. □

Lemma 3.7. For any given privacy budget ε such that $\varepsilon_q < O(1)$ and any subset of nodes $T \subseteq V$, the communication cost is $\tilde{O}(q \cdot \mathbb{I}[f_{T,k}(x_i) \neq 0] + q \cdot \frac{\Delta}{n\varepsilon})$ bits for node v_i in expectation.

Proof. The proof follows Lemma 2.10 and $\varepsilon = \Theta(q\varepsilon_q)$ when $\varepsilon_q < O(1)$. □

Example 3.2. Consider the case where $T = \{v_1, v_2\}$ and $k = 2$, then the mechanism counts the number of 2-stars in the whole graph such that v_1 and v_2 are the leaves. We can compute $\Delta = 1$, i.e., $f_{T,k}(x_i)$ is either 0 or 1 for any $i \in [n]$. Thus,

$$\sum_{i \in [n]} \left(f_{T,k}(x_i)\right)^2 = \sum_{i \in [n]} f_{T,k}(x_i) = f_{T,k}(I).$$

Then we have

$$\mathbb{E}[\tilde{f}_{T,k}(I)] = f_{T,k}(I), \quad \text{Var}[\tilde{f}_{T,k}(I)] = \frac{1-q}{q} \cdot f_{T,k}(I) + O\left(\frac{1}{\varepsilon^2}\right).$$

The communication cost is $\tilde{O}(q \cdot f_{T,k}(x_i) + \frac{q}{n\varepsilon})$ bits for node v_i in expectation.

Example 3.3. Consider another case where $T = \{v_1\}$ and $k = 2$, then the mechanism counts the number of 2-stars in the whole graph such that v_1 is one of the leaves. We can compute $\Delta = d$ and set $q = 1$, i.e., there is no sampling, then

$$\mathbb{E}[\tilde{f}_{T,k}(I)] = f_{T,k}(I), \quad \text{Var}[\tilde{f}_{T,k}(I)] = O\left(\frac{d^2}{\varepsilon^2}\right).$$

The communication cost is $\tilde{O}(d + \frac{d}{n\varepsilon})$ bits for each node in expectation.

4 Subgraph Counting

We then present the explicit mechanism for counting various subgraphs. In this work, we mainly focus on subgraphs that cannot be entirely observed by individual nodes: For k-star counting, a straightforward shuffle DP mechanism can be designed by combining the k-star counting mechanism under local DP [17] with privacy amplification by shuffling.

4.1 Triangle Counting

In this section, we study the triangle counting problem. Given a graph instance I, we focus on estimating the number of triangles in I under shuffle DP. More specifically, we set $k = 2$ and the number of triangles in I is

$$C_\triangle(I) = \sum_{\bar{V}=\{v_i,v_j\},v_i,v_j\in V} f_{\bar{V},2}(I)\cdot x_{i,j}.$$

It is noted that each triangle is counted three times in $C_\triangle(I)$. However, we can scale down the count during post-processing, which does not affect privacy.

4.1.1 Mechanism. We first split the privacy budget. Let $\varepsilon' = \frac{\varepsilon}{2}$,

$$\varepsilon'' = \text{argmax}\{\varepsilon : \sqrt{4d\ln(\frac{2}{\delta})}\varepsilon + 2d\varepsilon(e^\varepsilon - 1) \le \varepsilon'\},$$

and $\delta'' = \frac{\delta}{4d}$. The local randomizer $\mathcal{R}_\triangle$ looks as follows: Each node v first invokes the local randomizer $\mathcal{R}_{\text{adj}}(x,\varepsilon')$ for analyzing the adjacent matrix. Then for any set of nodes $\bar{V} = \{v_i, v_j\}, v_i, v_j \in V$, each node v invokes the local randomizer $\mathcal{R}_{\text{star}}(x, f_{\bar{V},2}, n, q, \varepsilon'', \delta'')$ to count the number of 2-stars such that $\bar{V} = \{v_i, v_j\}$ are the leaves, where q is a given sampling probability. The shuffler then shuffles all the messages, and the analyzer $\mathcal{A}_\triangle$ computes

$$\tilde{C}_\triangle(I) = \sum_{\bar{V}=\{v_i,v_j\},v_i,v_j\in V} \tilde{f}_{\bar{V},2}(I)\cdot\tilde{x}_{i,j},$$

where all the $\tilde{f}_{\bar{V},2}(I)$'s and $\tilde{x}_{i,j}$'s are computed as described in the building blocks in Section 3.

4.1.2 Analysis.

Theorem 4.1. For any given privacy budget ε, the sampling probability q and the maximum degree d such that $\varepsilon_q'' = \ln(1+\frac{1}{q}\cdot(e^{\varepsilon''}-1)) < 4$, the local randomizer $\mathcal{R}_\triangle$ satisfies (ε,δ)-shuffle DP.

Proof. First, invoking the local randomizer $\mathcal{R}_{\text{adj}}$ satisfies pure ε'-edge local (and thus shuffle) DP. Next, we notice that adding an edge to the graph instance can change at most $2d$ output distributions across all local randomizers $\mathcal{R}_{\text{star}}$'s. According to the composition theorems, invoking all the local randomizers $\mathcal{R}_{\text{star}}$'s satisfies (ε',δ)-shuffle DP. Therefore, the local randomizer $\mathcal{R}_\triangle$ satisfies (ε,δ)-shuffle DP. □

We then state the accuracy and efficiency guarantees, for which we assume $\varepsilon < 2$ so that $\varepsilon' < 1$ and

$$\varepsilon'' = \Theta\left(\frac{\varepsilon}{\sqrt{d\log(\frac{1}{\delta})}}\right).$$

Theorem 4.2. For any given privacy budget $\varepsilon < 2$ and sampling probability q such that $\varepsilon_q'' = \ln(1+\frac{1}{q}\cdot(e^{\varepsilon''}-1)) < O(1)$, and any instance $I \in \mathcal{I}$, we have

$$\mathbb{E}[\tilde{C}_\triangle(I)] = C_\triangle(I),$$

and

$$\text{Var}[\tilde{C}_\triangle(I)] = O\left(\frac{nd^3}{\varepsilon^2}\right) + \frac{1-q}{q}\cdot O\left(\frac{nd^2}{\varepsilon^2}\right) + \tilde{O}\left(\frac{n^2d}{\varepsilon^4}\right).$$

Proof. The expectation is

$$\begin{aligned}\mathbb{E}[\tilde{C}_\triangle(I)] &= \sum_{\bar{V}=\{v_i,v_j\},v_i,v_j\in V}\mathbb{E}[\tilde{f}_{\bar{V},2}(I)]\cdot\mathbb{E}[\tilde{x}_{i,j}]\\ &= \sum_{\bar{V}=\{v_i,v_j\},v_i,v_j\in V} f_{\bar{V},2}(I)\cdot x_{i,j}\\ &= C_\triangle(I)\end{aligned}$$

The estimate for any two sets of nodes is independent, thus, the variance is

$$\text{Var}[\tilde{C}_\triangle(I)] = \sum_{\bar{V}=\{v_i,v_j\},v_i,v_j\in V}\text{Var}[\tilde{f}_{\bar{V},2}(I)\cdot\tilde{x}_{i,j}].$$

We first note that for any set of nodes $\bar{V} = \{v_i, v_j\}, v_i, v_j \in V$,

$$[\mathbb{E}[\tilde{x}_{i,j}]]^2 = (x_{i,j})^2 = x_{i,j}.$$

Combining this with the results from Example 3.2, we can compute that for any set of nodes $\bar{V} = \{v_i, v_j\}, v_i, v_j \in V$,

$$\begin{aligned}&\text{Var}[\tilde{f}_{\bar{V},2}(I)\cdot\tilde{x}_{i,j}]\\ =&\text{Var}[\tilde{x}_{i,j}]\cdot[\mathbb{E}[\tilde{f}_{\bar{V},2}(I)]]^2 + \left([\mathbb{E}[\tilde{x}_{i,j}]]^2 + \text{Var}[\tilde{x}_{i,j}]\right)\cdot\text{Var}[\tilde{f}_{\bar{V},2}(I)]\\ =&O\left(\frac{1}{\varepsilon^2}\right)\cdot\left(f_{\bar{V},2}(I)\right)^2 + \left(x_{i,j} + O\left(\frac{1}{\varepsilon^2}\right)\right)\cdot\left(\frac{1-q}{q}\cdot f_{\bar{V},2}(I) + \tilde{O}\left(\frac{d}{\varepsilon^2}\right)\right)\\ =&O\left(\frac{1}{\varepsilon^2}\right)\cdot\left(\left(f_{\bar{V},2}(I)\right)^2 + \frac{1-q}{q}\cdot f_{\bar{V},2}(I) + \tilde{O}\left(\frac{d}{\varepsilon^2}\right)\right)\end{aligned}$$

Moreover, given the maximum degree is upper bounded by d, we have

$$\sum_{\bar{V}=\{v_i,v_j\},v_i,v_j\in V} f_{\bar{V},2}(I) = O(nd^2),$$

$$\sum_{\bar{V}=\{v_i,v_j\},v_i,v_j\in V}\left(f_{\bar{V},2}(I)\right)^2 = O(nd^3).$$

Therefore, the overall variance is

$$\begin{aligned}&\text{Var}[\tilde{C}_\triangle(I)]\\ =&O\left(\frac{1}{\varepsilon^2}\right)\cdot\sum_{\bar{V}=\{v_i,v_j\},v_i,v_j\in V}\left(\left(f_{\bar{V},2}(I)\right)^2 + \frac{1-q}{q}\cdot f_{\bar{V},2}(I) + \tilde{O}\left(\frac{d}{\varepsilon^2}\right)\right)\\ =&O\left(\frac{1}{\varepsilon^2}\right)\cdot\left(O(nd^3) + \frac{1-q}{q}\cdot O(nd^2) + \tilde{O}\left(\frac{n^2d}{\varepsilon^2}\right)\right)\\ =&O\left(\frac{nd^3}{\varepsilon^2}\right) + \frac{1-q}{q}\cdot O\left(\frac{nd^2}{\varepsilon^2}\right) + \tilde{O}\left(\frac{n^2d}{\varepsilon^4}\right)\end{aligned}$$

□

Theorem 4.3. For any given privacy budget $\varepsilon < 2$ and sampling probability q such that $\varepsilon_q'' = \ln(1+\frac{1}{q}\cdot(e^{\varepsilon''}-1)) < O(1)$, the communication cost for each node is $\tilde{O}(q\cdot d^2 + \frac{qn\sqrt{d}}{\varepsilon})$ bits in expectation.

Proof. Consider any node $v \in V$, $O(n)$ bits are sent to analyze the adjacent matrix. Moreover, for any node v_i, it is noted that

$$\sum_{\bar{V}=\{v_i,v_j\},v_i,v_j\in V} f_{\bar{V}}(x_i) = O(d^2),$$

thus, according to Lemma 3.7, $\tilde{O}(q\cdot d^2 + q\cdot\frac{n\sqrt{d}}{\varepsilon})$ bits are sent in expectation for counting the 2-stars. It is noted that each message requires $O(\log(n))$ additional bits to identify the specific set of

nodes $\bar{V} = \{v_i, v_j\}$ we are working with, ensuring that the counts for different k-stars remain distinct during the shuffling process. □

Example 4.1. Consider the case where the constant privacy budget $\varepsilon < 2$ and the graph is sparse in the sense that $d^{1.5} \le n$. If we set $q = \frac{1}{\sqrt{d}}$, then

$$\frac{1-q}{q} \cdot O\left(\frac{nd^2}{\varepsilon^2}\right) = O\left(\frac{nd^{2.5}}{\varepsilon^2}\right),$$

i.e., the sample variance is dominated by other terms. The overall variance is $\tilde{O}(nd^3 + n^2 d)$ and the communication cost is $\tilde{O}(n)$ bits per node in expectation. We can see that sampling does not increase the variance much while saving the communication cost.

4.1.3 Improvement of Communication Cost. Moreover, we can reduce the communication cost by estimating the frequency for only a few k-stars. More specifically, we divide the sets of nodes into n different groups: For $\ell \in [n]$, let

$$\Gamma_\ell = \{\{v_i, v_j\} : v_i, v_j \in V, i + j \equiv \ell \pmod n\}$$

denote the ℓ-th group of the sets of nodes. Let m be a given parameter denoting the number of groups that we need to analyze. We set $\check{d} = \min(d, m)$ and split the privacy budget as follows. Let

$$\varepsilon^{\text{bsc}} = \frac{\varepsilon'}{2\check{d}}, \quad \varepsilon^{\text{adv}} = \operatorname{argmax}\{\varepsilon : \sqrt{4\check{d}\ln(\frac{2}{\delta})}\varepsilon + 2\check{d}\varepsilon(e^\varepsilon - 1) \le \varepsilon'\},$$

and $\varepsilon'' = \max(\varepsilon^{\text{bsc}}, \varepsilon^{\text{adv}})$. We then set $\delta'' = \frac{\delta}{2\check{d}}$ if $\varepsilon'' = \varepsilon^{\text{bsc}}$ and $\delta'' = \frac{\delta}{4\check{d}}$ otherwise. The overall flow of the mechanism is modified as well. The analyzer $\mathcal{A}_\triangle$ first chooses m random numbers $L \subseteq [n]$ and sends the numbers to each node. Then the local randomizer $\mathcal{R}^m_\triangle$ looks as follows: Each node v first invokes the local randomizer $\mathcal{R}_{\text{adj}}(x, \varepsilon')$ for analyzing the adjacent matrix (but only sends the necessary $y_{i,j}$'s for sets of nodes $\{v_i, v_j\} \in \Gamma_\ell, \ell \in L$). Next, for any set of nodes $\bar{V} = \{v_i, v_j\} \in \Gamma_\ell, \ell \in L$, each node v invokes the local randomizer $\mathcal{R}_{\text{star}}(x, f_{\bar{V},2}, n, q, \varepsilon'', \delta'')$. The shuffler shuffles all the messages and the analyzer $\mathcal{A}_\triangle$ computes

$$\tilde{C}_\triangle(I) = \frac{n}{m} \cdot \sum_{\ell \in L} \sum_{\bar{V} = \{v_i, v_j\} \in \Gamma_\ell} \tilde{f}_{\bar{V},2}(I) \cdot \tilde{x}_{i,j}.$$

The privacy guarantee remains the same. Specifically, the local randomizer $\mathcal{R}^m_\triangle$ satisfies (ε, δ)-shuffle DP as long as $\varepsilon''_q = \ln(1 + \frac{1}{q} \cdot (e^{\varepsilon''} - 1)) < 4$. Furthermore, the number of noisy messages can be reduced from $\tilde{O}(\frac{qn\sqrt{d}}{\varepsilon})$ to $\tilde{O}(\frac{qm\sqrt{d}}{\varepsilon})$.

4.2 4-cycle Counting

Our triangle-counting mechanism can be easily extended to count 4-cycles. More specifically, we set $k = 2$ and the number of 4-cycles in I is

$$C_\square(I) = \sum_{\bar{V} = \{v_i, v_j\}, v_i, v_j \in V} f_{\bar{V},2}(I) \cdot \left(f_{\bar{V},2}(I) - 1\right).$$

Each 4-cycle is counted four times in $C_\square(I)$ and we can also scale the count down during post-processing.

4.2.1 Mechanism. The privacy budgets ε'' and δ'' are the same as the ones in Section 4.1.1. The local randomizer $\mathcal{R}_\square$ looks as follows: For any set of nodes $\bar{V} = \{v_i, v_j\}, v_i, v_j \in V$, each node v invokes the local randomizer $\mathcal{R}_{\text{star}}(x, f_{\bar{V},2}, n, q, \varepsilon'', \delta'')$ twice. The shuffler shuffles all the messages and the analyzer $\mathcal{A}_\square$ can compute

$$\tilde{C}_\square(I) = \sum_{\bar{V} = \{v_i, v_j\}, v_i, v_j \in V} \tilde{f}^1_{\bar{V},2}(I) \cdot \left(\tilde{f}^2_{\bar{V},2}(I) - 1\right),$$

where $\tilde{f}^1_{\bar{V},2}(I)$ and $\tilde{f}^2_{\bar{V},2}(I)$ are obtained from two separate invocations of the local randomizer $\mathcal{R}_{\text{star}}$.

4.2.2 Analysis. We then present the analysis. Notably, the proofs for the privacy and efficiency analysis are similar to those in triangle counting and are therefore omitted.

Theorem 4.4. For any given privacy budget ε, the sampling probability q and the maximum degree d such that $\varepsilon''_q = \ln(1 + \frac{1}{q} \cdot (e^{\varepsilon''} - 1)) < 4$, the local randomizer $\mathcal{R}_\square$ satisfies (ε, δ)-shuffle DP.

Theorem 4.5. For any instance $I \in \mathcal{I}$,

$$\mathbb{E}[\tilde{C}_\square(I)] = C_\square(I).$$

When $q = 1$, for any given privacy budget $\varepsilon < 2$ and any instance $I \in \mathcal{I}$,

$$\text{Var}[\tilde{C}_\square(I)] = \tilde{O}\left(\frac{nd^4}{\varepsilon^2}\right) + \tilde{O}\left(\frac{n^2d^2}{\varepsilon^4}\right).$$

Instead, when $q = \frac{1}{\sqrt{d}}$, for any given privacy budget $\varepsilon < 2$ such that $\varepsilon''_q = \ln(1 + \frac{1}{q} \cdot (e^{\varepsilon''} - 1)) < O(1)$ and any instance $I \in \mathcal{I}$,

$$\text{Var}[\tilde{C}_\square(I)] = O(nd^{4.5}) + \tilde{O}\left(\frac{nd^4}{\varepsilon^2}\right) + \tilde{O}\left(\frac{n^2d^2}{\varepsilon^4}\right).$$

Proof. The proof is similar to the one of triangle counting, and we reuse some results from that proof. For any set of nodes $\bar{V} = \{v_i, v_j\}, v_i, v_j \in V$, we have

$$\mathbb{E}[\tilde{f}^1_{\bar{V},2}(I)] = \mathbb{E}[\tilde{f}^2_{\bar{V},2}(I)] = \mathbb{E}[\tilde{f}_{\bar{V},2}(I)] = f_{\bar{V},2}(I),$$

and

$$\text{Var}[\tilde{f}^1_{\bar{V},2}(I)] = \text{Var}[\tilde{f}^2_{\bar{V},2}(I)] = \text{Var}[\tilde{f}_{\bar{V},2}(I)],$$

where $\tilde{f}_{\bar{V},2}(I)$ is the estimate in triangle counting. Thus, the expectation is

$$\begin{aligned}\mathbb{E}[\tilde{C}_\square(I)] &= \sum_{\bar{V} = \{v_i, v_j\}, v_i, v_j \in V} \mathbb{E}[\tilde{f}^1_{\bar{V},2}(I)] \cdot (\mathbb{E}[\tilde{f}^2_{\bar{V},2}(I)] - 1) \\ &= \sum_{\bar{V} = \{v_i, v_j\}, v_i, v_j \in V} f_{\bar{V},2}(I) \cdot (f_{\bar{V},2}(I) - 1) \\ &= C_\square(I)\end{aligned}$$

The estimate for any two sets of nodes is independent, therefore, the variance is

$$\text{Var}[\tilde{C}_\square(I)] = \sum_{\bar{V} = \{v_i, v_j\}, v_i, v_j \in V} \text{Var}[\tilde{f}^1_{\bar{V},2}(I) \cdot \left(\tilde{f}^2_{\bar{V},2}(I) - 1\right)].$$

We first analyze the variance $\mathrm{Var}[\tilde{f}^1_{\bar{V},2}(I) \cdot (\tilde{f}^2_{\bar{V},2}(I) - 1)]$. For any set of nodes $\bar{V} = \{v_i, v_j\}, v_i, v_j \in V$,

$$\begin{aligned}
&\mathrm{Var}[\tilde{f}^1_{\bar{V},2}(I) \cdot (\tilde{f}^2_{\bar{V},2}(I) - 1)] \\
=&\mathrm{Var}[\tilde{f}_{\bar{V},2}(I)] \cdot \left([\mathbb{E}[\tilde{f}_{\bar{V},2}(I) - 1]]^2 + [\mathbb{E}[\tilde{f}_{\bar{V},2}(I)]]^2 + \mathrm{Var}[\tilde{f}_{\bar{V},2}(I)]\right) \\
=&\left(\frac{1-q}{q} \cdot f_{\bar{V},2}(I) + \tilde{O}\left(\frac{d}{\varepsilon^2}\right)\right) \cdot \left(2(f_{\bar{V},2}(I))^2 + \frac{1-3q}{q} \cdot f_{\bar{V},2}(I) + \tilde{O}\left(\frac{d}{\varepsilon^2}\right)\right)
\end{aligned}$$

Given the maximum degree is upper bounded by d, we have

$$\sum_{\bar{V}=\{v_i,v_j\}, v_i, v_j \in V} (f_{\bar{V},2}(I))^3 = O(nd^4).$$

Therefore, the overall variance is

$$\begin{aligned}
\mathrm{Var}[\tilde{C}_\square(I)] =& \sum_{\bar{V}=\{v_i,v_j\}, v_i, v_j \in V} \mathrm{Var}[\tilde{f}^1_{\bar{V},2}(I) \cdot (\tilde{f}^2_{\bar{V},2}(I) - 1)] \\
=& \frac{1-q}{q} \cdot O(nd^4) + \frac{(1-q)(1-3q)}{q^2} \cdot O(nd^3) \\
&+ \frac{2-4q}{q} \cdot \tilde{O}\left(\frac{nd^3}{\varepsilon^2}\right) + \tilde{O}\left(\frac{nd^4}{\varepsilon^2}\right) + \tilde{O}\left(\frac{n^2d^2}{\varepsilon^4}\right)
\end{aligned}$$

When $q = 1$, the variance is

$$\mathrm{Var}[\tilde{C}_\square(I)] = \tilde{O}\left(\frac{nd^4}{\varepsilon^2}\right) + \tilde{O}\left(\frac{n^2d^2}{\varepsilon^4}\right).$$

Instead, when $q = \frac{1}{\sqrt{d}}$, the variance is

$$\mathrm{Var}[\tilde{C}_\square(I)] = O(nd^{4.5}) + \tilde{O}\left(\frac{nd^4}{\varepsilon^2}\right) + \tilde{O}\left(\frac{n^2d^2}{\varepsilon^4}\right).$$

□

Theorem 4.6. For any given privacy budget $\varepsilon < 2$ and sampling probability q such that $\varepsilon''_q = \ln(1 + \frac{1}{q} \cdot (e^{\varepsilon''} - 1)) < O(1)$, the communication cost is $\tilde{O}(q \cdot d^2 + \frac{qn\sqrt{d}}{\varepsilon})$ bits per node in expectation.

4.2.3 Improvement of Communication Cost. Similarly, we can reduce the communication cost by estimating the frequency for only a few k-stars. We divide the sets of nodes into n groups as described in Section 4.1.3. Given the parameter m, the number of groups that we need to analyze, the privacy budgets ε'' and δ'' are the same as the ones in Section 4.1.3. The overall flow of the mechanism is modified similarly. The analyzer $\mathcal{A}_\square$ first chooses m random numbers $L \subseteq [n]$ and sends the numbers to each node. Then in the local randomizer $\mathcal{R}^m_\square$, each node v invokes the local randomizer $\mathcal{R}_{\mathrm{star}}(x, f_{\bar{V},2}, n, q, \varepsilon'', \delta'')$ twice for any set of nodes $\bar{V} = \{v_i, v_j\} \in \Gamma_\ell, \ell \in L$. The shuffler shuffles all the messages and the analyzer $\mathcal{A}_\square$ can compute

$$\tilde{f}_\square(I) = \frac{n}{m} \cdot \sum_{\ell \in L} \sum_{\bar{V} \in \Gamma_\ell} \tilde{f}^1_{\bar{V},2}(I) \cdot (\tilde{f}^2_{\bar{V},2}(I) - 1).$$

The local randomizer $\mathcal{R}^m_\square$ satisfies (ε, δ)-shuffle DP as long as $\varepsilon''_q = \ln(1 + \frac{1}{q} \cdot (e^{\varepsilon''} - 1)) < 4$. The number of noisy messages can also be reduced by a factor of $\frac{m}{n}$.

4.3 3-hop Path Counting

We then focus on the 3-hop path counting problem. Let $k = 2$, and the number of 3-hop paths in the instance I is

$$C_\sqcup(I) = \sum_{i \in [n]} f_{\{v_i\},2}(I) \cdot (d_i - 1).$$

Each 3-hop path is counted twice in $C_\sqcup(I)$, and triangles are counted as 3-hop paths as well. Since we can remove the triangles by counting them with part of the privacy budget, we only focus on the 3-hop path counting function $C_\sqcup(I)$ in this section.

A straightforward solution is to estimate $f_{\{v_i\},2}(I)$ directly using the local randomizer $\mathcal{R}_{\mathrm{star}}$ for any node $v_i \in V$, where we add noise proportional to $\Delta = O(d)$ for all the frequencies. However, many frequencies may not vary so much between neighboring instances, as demonstrated in Example 4.2.

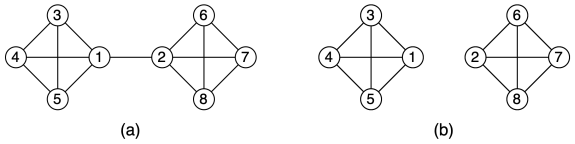

Figure 2: Neighboring instances in Example 4.2.

Example 4.2. Consider the neighboring instances as shown in Figure 2. Assume the given degree upper bound d is 4. For instance I in (a), we have $f_{\{v_1\},2}(I) = f_{\{v_2\},2}(I) = 9$ and $f_{\{v_i\},2}(I) = 7$ for any $i \geq 3$. In contrast, for the neighboring instance I' in (b), which removes the edge (v_1, v_2), we instead have $f_{\{v_i\},2}(I') = 6$ for any i. Therefore, the difference between $f_{\{v_i\},2}(I)$ and $f_{\{v_i\},2}(I')$ is 3 for $i = 1, 2$ and 1 for any other i.

We can verify that given the different edge (v_{i^*}, v_{j^*}) between neighboring instances $I \sim I'$, $f_{\{v_i\},2}(I)$ and $f_{\{v_i\},2}(I')$ may have a difference proportional to $O(d)$ if and only if $i = i^*$ or $i = j^*$. For any other i, $f_{\{v_i\},2}(I)$ and $f_{\{v_i\},2}(I')$ will differ by at most $O(1)$.

Therefore, we turn to the following solution and construct a new graph $\bar{I}$. We add $d - 1$ additional nodes $W = \{w_1, \ldots, w_{d-1}\}$ to the graph. For any node $v_i \in V$ and any $j \in [d_i - 1]$,[6] we also add an edge between v_i and w_j. Then for any node v_i, we have

$$\begin{aligned}
f_{\{v_i\},2}(I) =& \sum_{i' \in [n]} x_{i,i'} \cdot (d_{i'} - 1) = \sum_{i' \in [n]} x_{i,i'} \cdot \sum_{j \in [d-1]} \mathbb{I}[d_{i'} - 1 \geq j] \\
=& \sum_{j \in [d-1]} \sum_{i' \in [n]} x_{i,i'} \cdot \mathbb{I}[d_{i'} - 1 \geq j] = \sum_{j \in [d-1]} f_{\{v_i, w_j\},2}(\bar{I})
\end{aligned}$$

Therefore, for any $i \in [n]$, we can estimate $f_{\{v_i, w_j\},2}(\bar{I})$'s and sum them to estimate $f_{\{v_i\},2}(I)$'s. In this scenario, for any neighboring instances, the difference between $f_{\{v_i, w_j\},2}(\bar{I})$ and $f_{\{v_i, w_j\},2}(\bar{I}')$ is at most $O(1)$. In contrast, the difference between $f_{\{v_i\},2}(I)$ and $f_{\{v_i\},2}(I')$ can be $O(d)$. Therefore, our new method reduces the variance of each $f_{\{v_i\},2}(I)$ by a factor of d.

Example 4.3. Given $d = 3$, consider the instance I in Figure 3 (a). We construct the instance $\bar{I}$, as shown in Figure 3 (b), as follows: We first add two additional nodes w_1 and w_2. Then for v_1, since

[6] We set $[n] = \emptyset$ when $n \leq 0$.

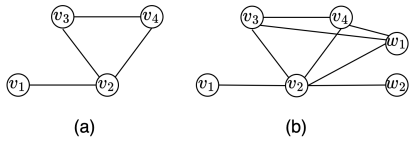

Figure 3: Instances in Example 4.3.

$d_1 = 1$, we do not add any edge between v_1 and $w_i, i = 1, 2$. For v_2, since $d_2 = 3$, we add edges (v_2, w_1) and (v_2, w_2). Similarly, we add edges (v_3, w_1) and (v_4, w_1) for the nodes v_3 and v_4. We can then verify that

$$f_{\{v_2\},2}(I) = f_{\{v_2,w_1\},2}(\bar{I}) + f_{\{v_2,w_2\},2}(\bar{I}) = 2 + 0 = 2,$$

and

$$f_{\{v_3\},2}(I) = f_{\{v_3,w_1\},2}(\bar{I}) + f_{\{v_3,w_2\},2}(\bar{I}) = 2 + 1 = 3.$$

4.3.1 Mechanism. We first split the privacy budget as follows. Let $\varepsilon' = \frac{\varepsilon}{2}$, $\varepsilon^{\text{grp}} = \frac{\varepsilon'}{3}$, $\delta^{\text{grp}} = \frac{\delta}{3e^{\varepsilon^{\text{grp}}}}$,

$$\varepsilon'' = \text{argmax}\{\varepsilon : \sqrt{4d\ln(\frac{2}{\delta})}\varepsilon + 2d\varepsilon(e^{\varepsilon} - 1) \le \varepsilon^{\text{grp}}\},$$

and $\delta'' = \frac{\delta}{4d}$. The local randomizer $\mathcal{R}_{\sqcup}$ looks as follows: Each node v first invokes the local randomizer $\mathcal{R}_{\text{deg}}(x, \varepsilon')$ to analyze the node degrees. Then, for any $i \in [n]$, let $\bar{x}_i \in \{0,1\}^{n+d-1}$ denote the data that the node v_i holds in the instance $\bar{I}$, i.e.,

$$\bar{x}_{i,j} = \begin{cases} x_{i,j} & \text{if } j \le n \\ \mathbb{I}[d_i - 1 \ge j - n] & \text{otherwise} \end{cases}$$

For any set of nodes $\{v_i, w_j\}, i \in [n], j \in [d-1]$, each node v then invokes the local randomizer $\mathcal{R}_{\text{star}}(\bar{x}, f_{\{v_i,w_j\},2}, n, q, \varepsilon'', \delta'')$. The shuffler shuffles the messages and the analyzer $\mathcal{A}_{\sqcup}$ can estimate $\tilde{d}_i$ and $\tilde{f}_{\{v_i,w_j\},2}(\bar{I})$ for all sets of nodes $\{v_i, w_j\}, i \in [n], j \in [d-1]$. Finally, the analyzer computes

$$\tilde{C}_{\sqcup}(I) = \sum_{i\in[n]} \sum_{j\in[d-1]} \tilde{f}_{\{v_i,w_j\},2}(\bar{I}) \cdot (\tilde{d}_i - 1).$$

4.3.2 Analysis.

Theorem 4.7. For any given privacy budget ε, the sampling probability q and the maximum degree d such that $\varepsilon_q'' = \ln(1 + \frac{1}{q} \cdot (e^{\varepsilon''} - 1)) < 4$, the local randomizer $\mathcal{R}_{\sqcup}$ satisfies (ε, δ)-shuffle DP.

Proof. First, invoking the local randomizer $\mathcal{R}_{\text{deg}}$ satisfies pure ε'-shuffle DP. We then observe that for any neighboring instances $I \sim I'$, the corresponding graphs $\bar{I}$ and $\bar{I}'$ can differ by at most 3 edges. Furthermore, any of these differing edges will change at most $2d$ output distributions of the local randomizer $\mathcal{R}_{\text{star}}$. Therefore, according to group privacy and the composition theorems, invoking all local randomizer $\mathcal{R}_{\text{star}}$'s satisfies (ε', δ)-shuffle DP and the local randomizer $\mathcal{R}_{\sqcup}$ satisfies (ε, δ)-shuffle DP. □

Theorem 4.8. For any given privacy budget $\varepsilon < 2$ such that $\varepsilon_q'' = \ln(1 + \frac{1}{q} \cdot (e^{\varepsilon''} - 1)) < O(1)$ and any instance $I \in \mathcal{I}$, we have

$$\mathbb{E}[\tilde{C}_{\sqcup}(I)] = C_{\sqcup}(I),$$

and

$$\text{Var}[\tilde{C}_{\sqcup}(I)] = \frac{1-q}{q} \cdot O(nd^4) + \frac{1-q}{q} \cdot O\left(\frac{nd^2}{\varepsilon^2}\right) + \tilde{O}\left(\frac{nd^4}{\varepsilon^2}\right) + \tilde{O}\left(\frac{nd^2}{\varepsilon^4}\right).$$

Proof. For any $i \in [n]$, the expectation is

$$\begin{aligned}\mathbb{E}[\tilde{f}_{\{v_i\},2}(I)] &= \sum_{\bar{V}=\{v_i,w_j\},j\in[d-1]} \mathbb{E}[\tilde{f}_{\bar{V},2}(\bar{I})] \\ &= \sum_{\bar{V}=\{v_i,w_j\},j\in[d-1]} f_{\bar{V},2}(\bar{I}) = f_{\{v_i\},2}(I)\end{aligned}$$

and thus,

$$\begin{aligned}\mathbb{E}[\tilde{C}_{\sqcup}(I)] &= \sum_{i\in[n]} \mathbb{E}[\tilde{f}_{\{v_i\},2}(I)] \cdot (\mathbb{E}[\tilde{d}_i] - 1) \\ &= \sum_{i\in[n]} f_{\{v_i\},2}(I) \cdot (d_i - 1) = C_{\sqcup}(I)\end{aligned}$$

For any node $v_i \in V$,

$$\begin{aligned}\text{Var}[\tilde{f}_{\{v_i\},2}(I)] &= \sum_{\bar{V}=\{v_i,w_j\},j\in[d-1]} \text{Var}[\tilde{f}_{\bar{V},2}(\bar{I})] \\ &= \frac{1-q}{q} \cdot O(d^2) + \tilde{O}\left(\frac{d^2}{\varepsilon^2}\right).\end{aligned}$$

and

$$\begin{aligned}&\text{Var}[\tilde{f}_{\{v_i\},2}(I) \cdot (\tilde{d}_i - 1)] \\ =&\text{Var}[\tilde{f}_{\{v_i\},2}(I)] \cdot ([\mathbb{E}[\tilde{d}_i - 1]]^2 + \text{Var}[\tilde{d}_i]) + [\mathbb{E}[\tilde{f}_{\{v_i\},2}(I)]]^2 \cdot \text{Var}[\tilde{d}_i] \\ =&\left(\frac{1-q}{q} \cdot O(d^2) + \tilde{O}\left(\frac{d^2}{\varepsilon^2}\right)\right) \cdot \left(O(d^2) + O\left(\frac{1}{\varepsilon^2}\right)\right) + O(d^4) \cdot O\left(\frac{1}{\varepsilon^2}\right) \\ =&\frac{1-q}{q} \cdot \left(O(d^4) + O\left(\frac{d^2}{\varepsilon^2}\right)\right) + \tilde{O}\left(\frac{d^4}{\varepsilon^2}\right) + \tilde{O}\left(\frac{d^2}{\varepsilon^4}\right)\end{aligned}$$

Therefore, the overall variance is

$$\begin{aligned}\text{Var}[\tilde{C}_{\sqcup}(I)] &= \sum_{i\in[n]} \text{Var}[\tilde{f}_{\{v_i\},2}(I) \cdot (\tilde{d}_i - 1)] \\ &= \frac{1-q}{q} \cdot O(nd^4) + \frac{1-q}{q} \cdot O\left(\frac{nd^2}{\varepsilon^2}\right) + \tilde{O}\left(\frac{nd^4}{\varepsilon^2}\right) + \tilde{O}\left(\frac{nd^2}{\varepsilon^4}\right)\end{aligned}$$

□

Theorem 4.9. For any given privacy budget $\varepsilon < 2$ such that $\varepsilon_q'' = \ln(1 + \frac{1}{q} \cdot (e^{\varepsilon''} - 1)) < O(1)$, the communication cost for each node is $\tilde{O}(q \cdot d^2 + \frac{qd^{1.5}}{\varepsilon})$ bits in expectation.

Proof. Consider any node $v \in V$, $O(\log(n))$ bits are sent for estimating the node degrees (assuming we clip the message to $[0, d]$), and according to Lemma 3.7, $\tilde{O}(q \cdot d^2 + \frac{qd^{1.5}}{\varepsilon})$ messages are sent in expectation for counting the 2-stars. □

Example 4.4. Consider the case where the constant privacy budget $\varepsilon < 2$. We can set $q = 1$, i.e., there is no sampling, then the variance is $\tilde{O}(nd^4)$ and the communication cost is $\tilde{O}(d^2)$ messages per node in expectation. Instead, if we set $q = \frac{1}{\sqrt{d}}$, the variance increases to $\tilde{O}(nd^{4.5})$ while the communication cost decreases to $\tilde{O}(d^{1.5})$ messages per node in expectation.

4.3.3 Improvement of Communication Cost. Similarly, we can reduce the communication cost by estimating the frequency for only a few k-stars. Given the parameter m, the number of k-star frequencies that we need to analyze, the privacy budgets ε'' and δ'' are the same as the ones in Section 4.3.1. The overall flow of the mechanism is modified similarly. The analyzer $\mathcal{A}_{\sqcup}$ first chooses m random numbers $L \subseteq [n]$ and sends the numbers to each node. Then the local randomizer $\mathcal{R}^m_{\sqcup}$ looks as follows: Each node v first invokes the local randomizer $\mathcal{R}_{\deg}(x, \varepsilon')$ to analyze the node degrees. For any sets of nodes $\{v_\ell, w_j\}, \ell \in L, j \in [d-1]$, each node v then invokes the local randomizer $\mathcal{R}_{\text{star}}(\bar{x}, f_{\{v_\ell, w_j\},2}, n, q, \varepsilon'', \delta'')$. The shuffler shuffles all the messages and the analyzer $\mathcal{A}_{\sqcup}$ can compute

$$\tilde{C}_{\sqcup}(I) = \frac{n}{m} \cdot \sum_{\ell \in L} \sum_{j \in [d-1]} \tilde{f}_{\{v_\ell, w_j\},2}(\bar{I}) \cdot (\tilde{d}_\ell - 1).$$

Clearly, the privacy guarantee still holds, and the number of noisy messages can also be reduced by a factor of $\frac{m}{n}$.

4.4 General Subgraph Counting

Our mechanism can be extended to count subgraphs beyond the ones discussed earlier. In the following paragraphs, we outline the approach for counting the other four-node subgraphs, while omitting the detailed analysis for brevity.

For G_6 in Figure 4, we set $k = 3$, and the number of G_6 in the instance I is

$$C_{G_6}(I) = \sum_{\bar{V}=\{v_i, v_j\}, v_i, v_j \in V} f_{\bar{V},3}(I) \cdot x_{i,j}.$$

We can estimate all $f_{\bar{V},3}(I)$'s and $x_{i,j}$'s using the local randomizer $\mathcal{R}_{\text{star}}$ and $\mathcal{R}_{\text{adj}}$ respectively. It is noted that in this particular case, we have a scenario where $k > |T|$, distinguishing it from the examples mentioned earlier.

For G_7 in Figure 4, i.e. the 2-triangle, we set $k = 2$, and the number of G_7 in the instance I is

$$C_{G_7}(I) = \sum_{\bar{V}=\{v_i, v_j\}, v_i, v_j \in V} f_{\bar{V},2}(I) \cdot (f_{\bar{V},2}(I) - 1) \cdot x_{i,j}.$$

We can estimate all $f_{\bar{V},2}(I)$'s and $x_{i,j}$'s using the local randomizer $\mathcal{R}_{\text{star}}$ and $\mathcal{R}_{\text{adj}}$ respectively. Each 2-triangle is counted twice, and we can also scale down the count during post-processing.

The solution for G_8 in Figure 4, i.e., the complete graph on 4 nodes, is more complicated. We set $k = 3$ and the number of G_8 in the graph is

$$C_{G_8}(I) = \sum_{\bar{V}=\{v_i, v_j, v_k\}, v_i, v_j, v_k \in V} f_{\bar{V},3}(I) \cdot x_{i,j} \cdot x_{i,k} \cdot x_{j,k}.$$

We can estimate these values using the local randomizer $\mathcal{R}_{\text{star}}$ and $\mathcal{R}_{\text{adj}}$. One potential problem is that the communication cost is quite high, as we need to count the k-stars for $O(n^3)$ sets of nodes. A sparse frequency estimation protocol under shuffle DP could potentially solve this problem. However, tackling this task is beyond the scope of the current work, and we defer it to the future.

The examples show that any subgraph counting problem can be viewed as dot products involving graph statistics, such as the adjacency matrix, degree distributions, and k-star distribution. This allows us to generalize our mechanism to count any complicated subgraphs as long as the privacy budget is well separated.

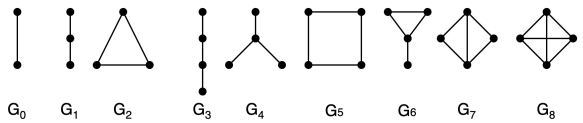

Figure 4: Subgraphs with at most four nodes.

4.5 Summary

We assume $\varepsilon = \Theta(1)$ and summarize the results in Table 1. For our mechanism, we set the sampling probability $q = \frac{1}{\sqrt{d}}$. The existing results for shuffle DP come from `WedgeShuffle` [19], and the result for triangle counting under local DP comes from `GenLocal` [12]. To our knowledge, this is the one-round mechanism that achieves the best accuracy guarantee, as `RR` [17] can only achieve a variance of $O(n^4)$. Furthermore, `GenLocal` can be extended to count any subgraph under local DP, with the accuracy analysis in the appendix. We report the variance, the communication cost, and the analysis time. For both our mechanism and `WedgeShuffle`, the analysis time is proportional to the total number of messages received from the shuffler. For `GenLocal`, it is instead $O(n^k)$, where k is the number of nodes in the target subgraph.

We begin by focusing on the shuffle DP mechanisms. Although our mechanism incurs a slightly higher communication cost, it is more general and exhibits a significantly lower variance across all subgraph counting tasks. Moreover, while not stated in the table, we can reduce the communication cost by adjusting the number of k-stars for analysis and the sampling probability, thereby achieving a balance between accuracy and communication cost. We then compare our results with those under local DP. Our variance is smaller, and a notable distinction lies in the analysis time: the local DP mechanism requires an analysis time proportional to $O(n^k)$ to count subgraphs with k nodes, which hinders its practicality.

Table 3: Basic information of graph datasets.

Dataset	Nodes	Edges	Max degree	Degree upper bound d
AstroPh	18,771	198,050	504	505
Enron	36,692	183,381	1,383	1,385
Facebook	4,039	88,234	1,045	1,050
GrQc	5,241	14,484	81	85

5 Experiments

We conducted extensive experiments with various subgraph counting queries. For comparison, we tested the following one-round shuffle-DP mechanisms:

- `GenShuffle`$_m$: Our general mechanism given the parameter m. When $m > 1$, we select the sampling probability q so that the number of noisy messages sent by each node is $\tilde{O}(n)$ in expectation; when $m = 1$, we simply set $q = 1$.
- `WedgeShuffle` [19]: The mechanism estimates wedge counts and count triangles and 4-cycles under shuffle DP.

We do not compare our approach with other one-round local DP mechanisms, as they are impractical or exhibit poor performance, as demonstrated in [19].

Table 4: Experimental Results for Subgraph Counting.

Dataset	AstroPh		Enron		Facebook		GrQc	
Triangle Count	1,351,441		727,044		1,612,010		48,260	
Result	Error (%)	Cost	Error (%)	Cost	Error (%)	Cost	Error (%)	Cost
$\texttt{GenShuffle}_n$	18.03	38.88 MB	109.21	78.60 MB	5.15	8.59 MB	54.83	10.17 MB
$\texttt{GenShuffle}_5$	37.10	22.12 KB	120.42	22.12 KB	10.20	22.13 KB	112.01	22.12 KB
$\texttt{GenShuffle}_1$	55.76	0.69 KB	99.36	0.69 KB	21.63	0.69 KB	83.08	0.69 KB
WedgeShuffle	74.55	0.10 KB	397.73	0.10 KB	20.46	0.10 KB	249.85	0.10 KB
4-cycle Count	44,916,549		36,262,229		144,023,053		1,054,723	
Result	Error (%)	Cost	Error (%)	Cost	Error (%)	Cost	Error (%)	Cost
$\texttt{GenShuffle}_n$	155.80	77.60 MB	1315.65	156.89 MB	25.96	17.15 MB	337.44	20.29 MB
$\texttt{GenShuffle}_{20}$	34.29	797.07 KB	130.75	797.12 KB	4.28	797.10 KB	234.13	796.93 KB
$\texttt{GenShuffle}_5$	39.72	44.20 KB	64.64	44.20 KB	9.49	44.23 KB	101.61	44.19 KB
$\texttt{GenShuffle}_1$	63.09	1.33 KB	89.52	1.33 KB	23.38	1.33 KB	97.23	1.33 KB
WedgeShuffle	42.42	0.10 KB	99.82	0.10 KB	24.19	0.10 KB	279.81	0.10 KB
3-hop Path Count	990,797,443		2,315,397,774		1,060,162,219		6,305,160	
Result	Error (%)	Cost	Error (%)	Cost	Error (%)	Cost	Error (%)	Cost
$\texttt{GenShuffle}_n$	0.97	6.22 MB	2.24	15.67 MB	4.00	1.37 MB	3.92	1.33 MB
$\texttt{GenShuffle}_{400}$	17.58	2.71 MB	39.65	6.60 MB	11.64	0.68 MB	23.00	0.62 MB
$\texttt{GenShuffle}_{100}$	32.95	2.38 MB	68.21	5.79 MB	22.44	0.61 MB	38.13	0.54 MB
$\texttt{GenShuffle}_1$	106.30	25.34 KB	98.48	61.47 KB	99.03	0.36 MB	105.04	5.79 KB

The experiments were conducted on a machine with a 2.2GHz Intel Xeon CPU and 256GB memory. We repeat each experiment 50 times, remove the best 10 and the worst 10 runs, and report the average error of the remaining runs. The default privacy budget is $\varepsilon = 4$ and $\delta = 10^{-5}$. The effect of ε is shown in the latter sections.

5.1 Datasets

We used the following real-world graph datasets: **AstroPh**, **Enron**, **Facebook**, and **GrQc**. Among the graph datasets, **AstroPh** and **GrQc** are collaboration networks, **Enron** is a communication network, and **Facebook** is a social network. The basic information of the graphs is given in Table 3: **AstroPh** and **Facebook** are comparatively dense, whereas **Enron** and **GrQc** are very sparse.

For the graph datasets, our mechanisms need a degree upper bound d. Similar to previous work, we choose d to be higher than the actual maximum degree, as shown in Table 3.

5.2 Experimental Results

The experimental results are shown in Table 4. We report the relative errors and the communication cost for all the mechanisms.

Specifically, for WedgeShuffle, we assume that the analyst sends a seed for each node to pair the nodes, and nodes only send messages when their values are non-zero. This considerably reduces the communication cost. In contrast, if this approach is not implemented, the communication costs of WedgeShuffle range from 32.82 KB to 298.12 KB. Furthermore, the analysis time is directly proportional to the communication cost for both shuffle-DP mechanisms; thus, we omit the timing details in this context.

For triangle counting, we set $q = \frac{1}{5\sqrt{d}}$ for $\texttt{GenShuffle}_n$. We can see that WedgeShuffle achieves an error of less than 100% in only two of the four instances. In contrast, our approach can provide reasonable answers across all four instances. Moreover, we observe that, for the dense graphs, i.e., **AstrPh** and **Facebook**, the error decreases as m increases. However, for the sparse ones, i.e., **Enron** and **GrQc**, the error may increase. This occurs because, with an increase in m, the sampling error decreases, but the DP noise increases as we allocate the privacy budget ε. Moreover, the communication cost of GenShuffle also rises with an increase of m. When $m = n$, the communication cost may seem excessively high, which appears to challenge our analysis of $\tilde{O}(n^2)$. However, this is attributed to the presence of several logarithmic terms within our communication cost calculation.

For 4-cycle counting, we also set $q = \frac{1}{5\sqrt{d}}$ for $\texttt{GenShuffle}_n$. We can see that $\texttt{GenShuffle}_1$ outperforms WedgeShuffle in three instances, and $\texttt{GenShuffle}_5$ performs better in all four instances. As m increases from 1 to 20, the error decreases for dense graphs, while it may increase for sparse graphs. Moreover, we notice that the error increases significantly when we set $m = n$. After careful investigation, we note that this is because when m is large, we need to split the privacy budget ε into very small portions. This leads to the variance term

$$\sum_{\tilde{V}=\{v_i,v_j\},v_i,v_j\in V} \mathrm{Var}[\tilde{f}^1_{\tilde{V},2}(I)] \cdot \mathrm{Var}[\tilde{f}^2_{\tilde{V},2}(I)] = \tilde{O}(\frac{n^2d^2}{\varepsilon^4})$$

dominating all other errors, resulting in poor performance of the mechanism. As a result, we recommend keeping the value of m relatively low when counting 4-cycles to enhance performance.

Finally, for 3-hop path counting, we set the sampling probability q so that the number of messages is about $20n$ for $\texttt{GenShuffle}_{100}$ and $\texttt{GenShuffle}_{400}$, and about $40n$ for $\texttt{GenShuffle}_n$. The amplified privacy budget ε_q is carefully verified to ensure that the mechanism satisfies DP. The results indicate that the accuracy improves as m increases, even when the sampling probability q is progressively reduced to satisfy the communication cost constraints. Ultimately, when m reaches n, the mechanism obtains the lowest error. Thus, for counting 3-hop paths, we recommend setting $m = n$ to optimize performance, while managing communication costs through the sampling probability q.

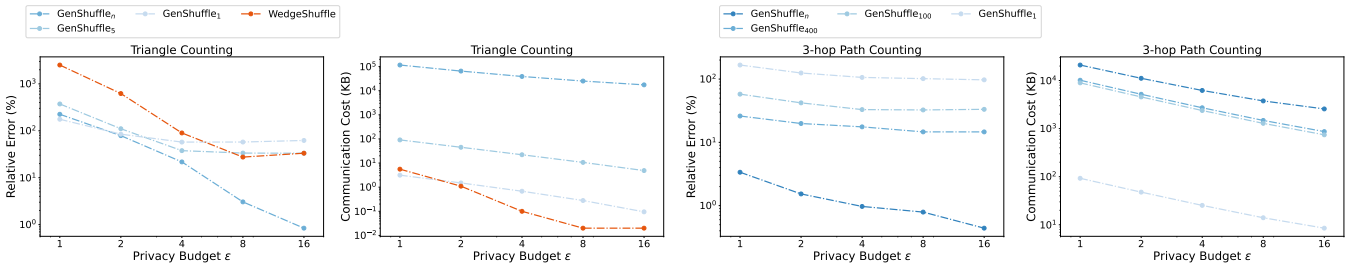

Figure 5: Experimental Results for Subgraph Counting on AstroPh with Various ε.

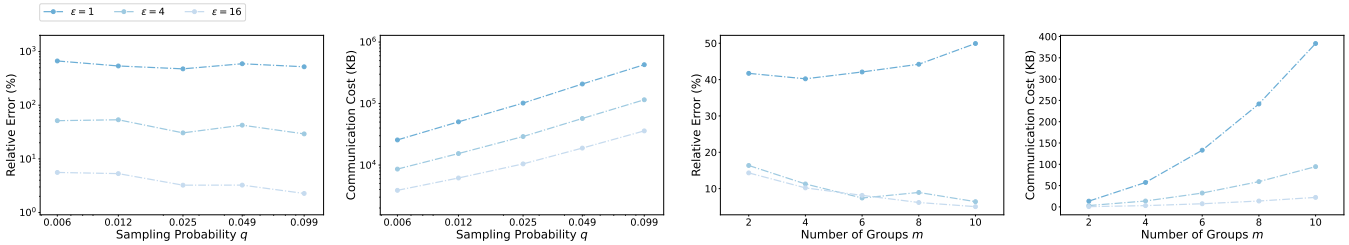

Figure 6: Experimental Results for Triangle Counting on Facebook with Various q and m.

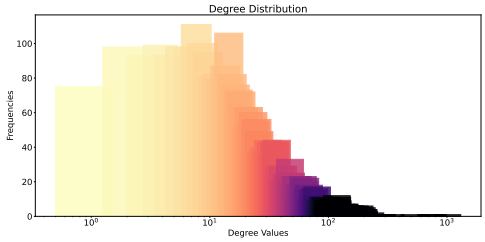

Figure 7: Degree distribution of Facebook.

Privacy Budget ε. We take triangle counting and 3-hop path counting on **AstroPh** as examples, and the results are shown in Figure 5. We first focus on our mechanism GenShuffle. The relative error and communication cost decrease as the privacy budget increases. Moreover, the relative error of GenShuffle$_n$ decreases much faster than that of GenShuffle$_m$'s where $m < n$. This is because the sampling error is small for GenShuffle$_n$. However, for others, the sampling error is the main component in the overall error. Similarly, for WedgeShuffle, with our improvement on their implementation, the error and the communication cost both decrease as the privacy budget increases. We then compare the mechanisms together for triangle counting. When the privacy budget ε is small, our mechanism GenShuffle outperforms the baselines in terms of accuracy. In contrast, when the privacy budget ε is large, all mechanisms except for GenShuffle$_n$ demonstrate similar performance due to the dominating sampling error. In this context, GenShuffle$_n$ can achieve considerably better accuracy.

Sampling Probability q and Number of Groups m. We start by analyzing the effect of q. Using triangle counting on **Facebook** as the example, we evaluate GenShuffle$_n$ with various ε and q. The results are shown in Figure 6. As we increase q, the error decreases while the communication cost increases. It is intuitive, as it represents the trade-off between accuracy and efficiency.

Next, we investigate the impact of the number of groups m. With $\varepsilon = 4$, the results presented in Table 4 indicate that a large m usually enhances performance for triangle counting and 3-hop path counting. However, for 4-cycle counting, a large m can yield unhelpful results, as explained before. We then analyze the effects for various ε, with the results illustrated in Figure 6. When ε is large, increasing the number of groups m leads to a reduction in error and a rise in communication costs. Conversely, when ε is small, such as $\varepsilon = 1$, the error may increase with a larger m. This phenomenon occurs because, with a smaller ε, the DP error is significantly greater than the sampling error, and it escalates considerably when we allocate the privacy budget across more groups.

To summarize, our protocols' empirical performance depends critically on m, while its optimal value can vary based on the density of the input graph, the subgraph pattern, and the privacy budget. While determining the optimal value of m in a principled manner remains an open question, we provide the following empirical guidelines according to our experimental study:

(1) A smaller m is generally better for sparse graphs, while a larger m is better for dense graphs.
(2) A smaller m is better for 4-cycle counting, while a larger m is better for triangle and 3-hop path counting.
(3) A smaller m is better when ε is small, while a larger m is better when ε is large.

Degree Upper Bound d. Throughout our analysis and experiments so far, we have assumed a degree upper bound d. While such an upper bound is required for the general case, it is only a soft requirement for GenShuffle$_m$ to count triangles or 4-cycles when $m \le d$, i.e., the privacy guarantee of the protocol still holds even if some

Table 5: Experimental Results for Triangle Counting on Facebook with Various d and m.

Relative Error (%)			
d	$\texttt{GenShuffle}_1$	$\texttt{GenShuffle}_5$	$\texttt{GenShuffle}_{20}$
1050	21.63	10.20	5.69
1000	20.61	11.38	5.74
300	20.04	10.33	6.69

nodes have degrees higher than the given d. This is because when $m \leq d$, we have $\check{d} = \min(d, m) = m$ (as defined in Section 4.1.3). Consequently, the privacy budget is allocated using m instead of d.

Furthermore, we examine its effect on the errors by conducting experiments on **Facebook**, whose degree distribution is shown in Figure 7, and the true maximum degree is 1045. We tested $\texttt{GenShuffle}_m$ with $m = 1$, 5, and 20 with $d = 1050$, 1000 and 300. The experimental results are shown in Table 5, from which we see that the impact of different d values on the error is not significant. This is because, when $m < d$, the mechanism uses only m to separate the privacy budget, making d irrelevant for performance.

6 Future Work

In this work, we focus mainly on one-round mechanisms. It is well-established that for subgraph counting under edge local DP, multi-round mechanisms exhibit superior performance compared to one-round mechanisms. Consequently, it would be intriguing to investigate potential enhancements to the mechanism when employing multiple rounds. Moreover, it is known that node DP is a stronger policy than edge DP, as it protects the presence of any node along with all its incident edges. An interesting open question remains whether node DP can be supported in local or shuffle DP.

Acknowledgments

This work has been supported by HKRGC under grants 16205422, 16204223, and 16203924. We appreciate the anonymous reviewers for their helpful comments on the manuscript.

References

[1] Borja Balle, Gilles Barthe, and Marco Gaboardi. 2018. Privacy Amplification by Subsampling: Tight Analyses via Couplings and Divergences. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*. Curran Associates Inc., Red Hook, NY, USA, 6280–6290.

[2] Jeremiah Blocki, Avrim Blum, Anupam Datta, and Or Sheffet. 2013. Differentially Private Data Analysis of Social Networks via Restricted Sensitivity. In *Proceedings of the 4th Conference on Innovations in Theoretical Computer Science*. Association for Computing Machinery, New York, NY, USA, 87–96.

[3] Keith Bonawitz, Vladimir Ivanov, Ben Kreuter, Antonio Marcedone, H. Brendan McMahan, Sarvar Patel, Daniel Ramage, Aaron Segal, and Karn Seth. 2017. Practical Secure Aggregation for Privacy-Preserving Machine Learning. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*. Association for Computing Machinery, New York, NY, USA, 1175–1191.

[4] T-H. Hubert Chan, Elaine Shi, and Dawn Song. 2012. Optimal Lower Bound for Differentially Private Multi-party Aggregation. In *Proceedings of the 2012 Annual European Symposium on Algorithms (ESA)*. Springer Berlin Heidelberg, 277–288.

[5] Lijie Chen, Badih Ghazi, Ravi Kumar, and Pasin Manurangsi. 2020. On Distributed Differential Privacy and Counting Distinct Elements. arXiv:2009.09604

[6] Shixi Chen and Shuigeng Zhou. 2013. Recursive mechanism: towards node differential privacy and unrestricted joins. In *Proceedings of the 2013 ACM SIGMOD International Conference on Management of Data*. Association for Computing Machinery, New York, NY, USA, 653–664.

[7] Wei Dong, Juanru Fang, Ke Yi, Yuchao Tao, and Ashwin Machanavajjhala. 2022. R2T: Instance-optimal truncation for differentially private query evaluation with foreign keys. In *Proc. ACM SIGMOD International Conference on Management of Data*. Association for Computing Machinery, New York, NY, USA, 759–772.

[8] Wei Dong and Ke Yi. 2021. Residual Sensitivity for Differentially Private Multi-Way Joins. In *Proceedings of the 2021 International Conference on Management of Data*. Association for Computing Machinery, New York, NY, USA, 432–444.

[9] Wei Dong and Ke Yi. 2022. A Nearly Instance-optimal Differentially Private Mechanism for Conjunctive Queries. In *PODS*.

[10] Cynthia Dwork and Aaron Roth. 2014. The algorithmic foundations of differential privacy. *Foundations and Trends in Theoretical Computer Science* 9 (2014).

[11] Jean-Pierre Eckmann and Elisha Moses. 2002. Curvature of co-links uncovers hidden thematic layers in the World Wide Web. *Proceedings of the National Academy of Sciences* 99 (2002), 5825–5829.

[12] Talya Eden, Quanquan C. Liu, Sofya Raskhodnikova, and Adam Smith. 2023. Triangle Counting with Local Edge Differential Privacy. arXiv:2305.02263

[13] Badih Ghazi, Noah Golowich, Ravi Kumar, Rasmus Pagh, and Ameya Velingker. 2020. On the Power of Multiple Anonymous Messages. arXiv:1908.11358

[14] Badih Ghazi, Ravi Kumar, Pasin Manurangsi, and Rasmus Pagh. 2021. Private Counting from Anonymous Messages: Near-Optimal Accuracy with Vanishing Communication Overhead. arXiv:2106.04247

[15] Badih Ghazi, Ravi Kumar, Pasin Manurangsi, Rasmus Pagh, and Amer Sinha. 2021. Differentially Private Aggregation in the Shuffle Model: Almost Central Accuracy in Almost a Single Message. arXiv:2109.13158

[16] Yizhang He, Kai Wang, Wenjie Zhang, Xuemin Lin, Ying Zhang, and Wei Ni. 2025. Robust Privacy-Preserving Triangle Counting under Edge Local Differential Privacy. In *Proc. ACM SIGMOD International Conference on Management of Data*. Association for Computing Machinery, New York, NY, USA.

[17] Jacob Imola, Takao Murakami, and Kamalika Chaudhuri. 2021. Locally Differentially Private Analysis of Graph Statistics. In *30th USENIX Security Symposium (USENIX Security 21)*. USENIX Association, 983–1000.

[18] Jacob Imola, Takao Murakami, and Kamalika Chaudhuri. 2022. Communication-Efficient Triangle Counting under Local Differential Privacy. In *31st USENIX Security Symposium (USENIX Security 22)*. USENIX Association, Boston, MA, 537–554.

[19] Jacob Imola, Takao Murakami, and Kamalika Chaudhuri. 2022. Differentially Private Triangle and 4-Cycle Counting in the Shuffle Model. arXiv:2205.01429

[20] Vishesh Karwa, Sofya Raskhodnikova, Adam Smith, and Grigory Yaroslavtsev. 2011. Private analysis of graph structure. In *Proceedings of the VLDB Endowment*.

[21] Shiva Prasad Kasiviswanathan, Homin K. Lee, Kobbi Nissim, Sofya Raskhodnikova, and Adam Smith. 2010. What Can We Learn Privately? arXiv:0803.0924

[22] Shiva Prasad Kasiviswanathan, Kobbi Nissim, Sofya Raskhodnikova, and Adam Smith. 2013. Analyzing graphs with node differential privacy. In *Proceedings of the 10th Theory of Cryptography Conference on Theory of Cryptography*. Springer-Verlag, Berlin, Heidelberg, 457–476.

[23] Shang Liu, Yang Cao, Takao Murakami, Jinfei Liu, and Masatoshi Yoshikawa. 2024. CARGO: Crypto-Assisted Differentially Private Triangle Counting without Trusted Servers. In *2024 IEEE 40th International Conference on Data Engineering (ICDE)*. 1671–1684.

[24] Yuhan Liu, Suyun Zhao, Yixuan Liu, Dan Zhao, Hong Chen, and Cuiping Li. 2022. Collecting Triangle Counts with Edge Relationship Local Differential Privacy. In *2022 IEEE 38th International Conference on Data Engineering (ICDE)*. 2008–2020.

[25] Guixun Luo, Yonghan Lin, Huilin Ai, Zhiyun Zhang, Naiyue Chen, and Li Duan. 2024. Privacy-Preserving Approximate Calculation of Subgraphs in Social Networking Services. In *2024 IEEE International Conference on Web Services (ICWS)*. 383–394.

[26] Qiyao Luo, Yilei Wang, and Ke Yi. 2022. Frequency Estimation in the Shuffle Model with Almost a Single Message. arXiv:2111.06833

[27] M. E. J. Newman. 2009. Random Graphs with Clustering. *Physical Review Letters* 103, 5 (2009).

[28] Gergely Palla, Imre Derényi, Illes Farkas, and Tamas Vicsek. 2005. Uncovering the overlapping community structure of complex networks in nature and society. *Nature* 435 (2005), 814–818.

[29] Haipei Sun, Xiaokui Xiao, Issa Khalil, Yin Yang, Zhan Qin, Hui (Wendy) Wang, and Ting Yu. 2019. Analyzing Subgraph Statistics from Extended Local Views with Decentralized Differential Privacy. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*. Association for Computing Machinery, New York, NY, USA, 703–717.

[30] Úlfar Erlingsson, Vitaly Feldman, Ilya Mironov, Ananth Raghunathan, Kunal Talwar, and Abhradeep Thakurta. 2019. Amplification by Shuffling: From Local to Central Differential Privacy via Anonymity. In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms*. Society for Industrial and Applied Mathematics, USA, 2468–2479.

[31] Stanley L. Warner. 1965. Randomized Response: A Survey Technique for Eliminating Evasive Answer Bias. *J. Amer. Statist. Assoc.* 60 (1965).

[32] Jun Zhang, Graham Cormode, Cecilia M Procopiuc, Divesh Srivastava, and Xiaokui Xiao. 2015. Private release of graph statistics using ladder functions. In *Proceedings of the 2015 ACM SIGMOD international conference on management of data*. 731–745.

A Appendix

A.1 Subgraph Counting under Local DP

Eden et al. [12] propose a mechanism that counts the number of triangles under local DP. The local randomizer is simply $\mathcal{R}_{\text{adj}}$. The analyst first estimates $\tilde{x}_{i,j}$ for any pair of $i, j \in [n], i < j$, and computes

$$\tilde{C}_{\triangle}(I) = 3 \cdot \sum_{\{i,j,k\} \in \binom{n}{3}, i<j<k} \tilde{x}_{i,j} \cdot \tilde{x}_{j,k} \cdot \tilde{x}_{i,k}.$$

The mechanism achieves the following guarantees of privacy and accuracy.

Theorem A.1. The mechanism satisfies pure ε-local DP. For any given privacy budget $\varepsilon = \Theta(1)$ and any instance $I \in \mathcal{I}$, $\mathbb{E}[\tilde{C}_{\triangle}(I)] = C_{\triangle}(I)$ and $\text{Var}[\tilde{C}_{\triangle}(I)] = O(n^3 + nd^3)$.

A.1.1 Extension to 3-hop Path Counting. We first consider the simple extension: The mechanism can be extended to count the number of 3-hop paths under local DP by computing

$$\tilde{C}_{\sqcup}(I) = 2 \cdot \sum_{\{i,j,k,\ell\} \in \binom{n}{4}, i<\ell} \tilde{x}_{i,j} \cdot \tilde{x}_{j,k} \cdot \tilde{x}_{k,\ell}.$$

Theorem A.2. For any given privacy budget ε and any instance $I \in \mathcal{I}$, $\mathbb{E}[\tilde{C}_{\sqcup}(I)] = C_{\sqcup}(I)$ and

$$\text{Var}[\tilde{C}_{\sqcup}(I)] = O\left(\frac{n^4}{\varepsilon^6}\right) + O\left(\frac{n^3d^2}{\varepsilon^4}\right) + O\left(\frac{n^2d^4}{\varepsilon^2}\right).$$

Proof. The expectation is

$$\begin{aligned}
&\mathbb{E}[\tilde{C}_{\sqcup}(I)] \\
=&2 \cdot \sum_{\{i,j,k,\ell\} \in \binom{n}{4}, i<\ell} \mathbb{E}[\tilde{x}_{i,j} \cdot \tilde{x}_{j,k} \cdot \tilde{x}_{k,\ell}] \\
=&2 \cdot \sum_{\{i,j,k,\ell\} \in \binom{n}{4}, i<\ell} \mathbb{E}[\tilde{x}_{i,j}] \cdot \mathbb{E}[\tilde{x}_{j,k}] \cdot \mathbb{E}[\tilde{x}_{k,\ell}] \\
=&2 \cdot \sum_{\{i,j,k,\ell\} \in \binom{n}{4}, i<\ell} x_{i,j} \cdot x_{j,k} \cdot x_{k,\ell} \\
=&C_{\sqcup}(I)
\end{aligned}$$

We then estimate the variance. We first estimate the covariance for different cases. First, for any $i, j, k, \ell, \ell' \in \binom{n}{5}$,

$$\text{Cov}[\tilde{x}_{i,j} \cdot \tilde{x}_{j,k} \cdot \tilde{x}_{k,\ell}, \tilde{x}_{i,j} \cdot \tilde{x}_{j,k} \cdot \tilde{x}_{k,\ell'}] = O\left(\frac{1}{\varepsilon^4}\right) \cdot x_{k,\ell} \cdot x_{k,\ell'},$$

thus,

$$\sum_{i,j,k,\ell,\ell' \in \binom{n}{5}, \ell<\ell'} \text{Cov}[\tilde{x}_{i,j} \cdot \tilde{x}_{j,k} \cdot \tilde{x}_{k,\ell}, \tilde{x}_{i,j} \cdot \tilde{x}_{j,k} \cdot \tilde{x}_{k,\ell'}] = O\left(\frac{n^3d^2}{\varepsilon^4}\right).$$

Similarly, we have

$$\begin{aligned}
&\sum_{i,j,k,\ell,\ell' \in \binom{n}{5}} \text{Cov}[\tilde{x}_{i,j} \cdot \tilde{x}_{j,k} \cdot \tilde{x}_{k,\ell}, \tilde{x}_{\ell',i} \cdot \tilde{x}_{i,j} \cdot \tilde{x}_{j,k}] \\
&\qquad + \text{Cov}[\tilde{x}_{i,j} \cdot \tilde{x}_{j,k} \cdot \tilde{x}_{k,\ell}, \tilde{x}_{\ell,i} \cdot \tilde{x}_{i,j} \cdot \tilde{x}_{j,k}] = O\left(\frac{n^3d^2}{\varepsilon^4}\right).
\end{aligned}$$

These are the two cases where the two products share two $\tilde{x}_{\cdot,\cdot}$ terms. Similarly, we can analyze the cases where the two products only share one $\tilde{x}_{\cdot,\cdot}$ term, and the overall covariance is $O\left(\frac{n^2d^4}{\varepsilon^2}\right)$. When the two products do not share any $\tilde{x}_{\cdot,\cdot}$ term, the covariance is 0. Therefore, the whole variance is

$$\begin{aligned}
&\text{Var}[\frac{1}{2} \cdot \tilde{C}_{\sqcup}(I)] \\
=&\sum_{\{i,j,k,\ell\} \in \binom{n}{4}, i<\ell} \text{Var}[\tilde{x}_{i,j} \cdot \tilde{x}_{j,k} \cdot \tilde{x}_{k,\ell}] + O\left(\frac{n^3d^2}{\varepsilon^4}\right) + O\left(\frac{n^3d^2}{\varepsilon^4}\right) \\
=&\sum_{\{i,j,k,\ell\} \in \binom{n}{4}, i<\ell} O\left(\frac{1}{\varepsilon^6}\right) + O\left(\frac{n^3d^2}{\varepsilon^4}\right) + O\left(\frac{n^3d^2}{\varepsilon^4}\right) \\
=&O\left(\frac{n^4}{\varepsilon^6}\right) + O\left(\frac{n^3d^2}{\varepsilon^4}\right) + O\left(\frac{n^2d^4}{\varepsilon^2}\right)
\end{aligned}$$

□

A.1.2 Extension to 4-cycle Counting. The mechanism can also be extended to count the number of 4-cycles under local DP by computing

$$\tilde{C}_{\square}(I) = 4 \cdot \sum_{\{i,j,k,\ell\} \in \binom{n}{4}, i<j<\ell, i<k} \tilde{x}_{i,j} \cdot \tilde{x}_{j,k} \cdot \tilde{x}_{k,\ell} \cdot \tilde{x}_{\ell,i}.$$

Theorem A.3. For any given privacy budget ε and any instance $I \in \mathcal{I}$, $\mathbb{E}[\tilde{C}_{\square}(I)] = C_{\square}(I)$, and

$$\text{Var}[\tilde{C}_{\square}(I)] = O\left(\frac{n^4}{\varepsilon^8}\right) + O\left(\frac{n^2d^3}{\varepsilon^4}\right) + O\left(\frac{nd^5}{\varepsilon^2}\right).$$

Proof. The expectation is

$$\begin{aligned}
&\mathbb{E}[\tilde{C}_{\square}(I)] \\
=&4 \cdot \sum_{\{i,j,k,\ell\} \in \binom{n}{4}, i<j<\ell, i<k} \mathbb{E}[\tilde{x}_{i,j} \cdot \tilde{x}_{j,k} \cdot \tilde{x}_{k,\ell} \cdot \tilde{x}_{\ell,i}] \\
=&4 \cdot \sum_{\{i,j,k,\ell\} \in \binom{n}{4}, i<j<\ell, i<k} \mathbb{E}[\tilde{x}_{i,j}] \cdot \mathbb{E}[\tilde{x}_{j,k}] \cdot \mathbb{E}[\tilde{x}_{k,\ell}] \cdot \mathbb{E}[\tilde{x}_{\ell,i}] \\
=&4 \cdot \sum_{\{i,j,k,\ell\} \in \binom{n}{4}, i<j<\ell, i<k} x_{i,j} \cdot x_{j,k} \cdot x_{k,\ell} \cdot x_{\ell,i} \\
=&C_{\square}(I)
\end{aligned}$$

and the variance is

$$\begin{aligned}
&\text{Var}[\frac{1}{4} \cdot \tilde{C}_{\square}(I)] \\
=&\sum_{\{i,j,k,\ell\} \in \binom{n}{4}, i<j<\ell, i<k} \text{Var}[\tilde{x}_{i,j} \cdot \tilde{x}_{j,k} \cdot \tilde{x}_{k,\ell} \cdot \tilde{x}_{\ell,i}] + 2\times \\
&\Bigg(\sum_{\{i,j,k,\ell,\ell'\} \in \binom{n}{5}, i<k, \ell<\ell'} \text{Cov}(\tilde{x}_{i,j} \cdot \tilde{x}_{j,k} \cdot \tilde{x}_{k,\ell} \cdot \tilde{x}_{\ell,i}, \tilde{x}_{i,j} \cdot \tilde{x}_{j,k} \cdot \tilde{x}_{k,\ell'} \cdot \tilde{x}_{\ell',i}) \\
&\qquad + \\
&\sum_{\{i,j,k,k',\ell,\ell'\} \in \binom{n}{6}, i<j, k<k'} \text{Cov}(\tilde{x}_{i,j} \cdot \tilde{x}_{j,k} \cdot \tilde{x}_{k,\ell} \cdot \tilde{x}_{\ell,i}, \tilde{x}_{i,j} \cdot \tilde{x}_{j,k'} \cdot \tilde{x}_{k',\ell'} \cdot \tilde{x}_{\ell',i}) \Bigg)
\end{aligned}$$

$$
\begin{aligned}
&= \sum_{\{i,j,k,\ell\}\in\binom{n}{4}, i<j<\ell, i<k} \Bigg(\mathbb{E}[\tilde{x}_{i,j}^2]\cdot\mathbb{E}[\tilde{x}_{j,k}^2]\cdot\mathbb{E}[\tilde{x}_{k,\ell}^2]\cdot\mathbb{E}[\tilde{x}_{\ell,i}^2] - \\
&\quad [\mathbb{E}[\tilde{x}_{i,j}]]^2\cdot[\mathbb{E}[\tilde{x}_{j,k}]]^2\cdot\mathbb{E}[[\tilde{x}_{k,\ell}]]^2\cdot\mathbb{E}[[\tilde{x}_{\ell,i}]]^2\Bigg) + 2\times \\
&\quad \Bigg(\sum_{\{i,j,k,\ell,\ell'\}\in\binom{n}{5}, i<k, \ell<\ell'} (\mathbb{E}[\tilde{x}_{i,j}^2]\cdot\mathbb{E}[\tilde{x}_{j,k}^2] - [\mathbb{E}[\tilde{x}_{i,j}]]^2\cdot[\mathbb{E}[\tilde{x}_{j,k}]]^2)\cdot \\
&\quad \mathbb{E}[\tilde{x}_{k,\ell}]\cdot\mathbb{E}[\tilde{x}_{\ell,i}]\cdot\mathbb{E}[\tilde{x}_{k,\ell'}]\cdot\mathbb{E}[\tilde{x}_{\ell',i}] \\
&\qquad\qquad + \\
&\quad \sum_{\{i,j,k,k',\ell,\ell'\}\in\binom{n}{6}, i<j, k<k'} (\mathbb{E}[\tilde{x}_{i,j}^2] - [\mathbb{E}[\tilde{x}_{i,j}]]^2)\cdot \\
&\quad \mathbb{E}[\tilde{x}_{j,k}]\cdot\mathbb{E}[\tilde{x}_{k,\ell}]\cdot\mathbb{E}[\tilde{x}_{\ell,i}]\cdot\mathbb{E}[\tilde{x}_{j,k'}]\cdot\mathbb{E}[\tilde{x}_{k',\ell'}]\cdot\mathbb{E}[\tilde{x}_{\ell',i}]\Bigg) \\
&= \sum_{\{i,j,k,\ell\}\in\binom{n}{4}, i<j<\ell, i<k} O\left(\frac{1}{\varepsilon^8}\right) + 2\times \\
&\quad \Bigg(\sum_{\{i,j,k,\ell,\ell'\}\in\binom{n}{5}, i<k, \ell<\ell'} O\left(\frac{1}{\varepsilon^4}\right)\cdot x_{k,\ell}\cdot x_{\ell,i}\cdot x_{k,\ell'}\cdot x_{\ell',i} \\
&\qquad\qquad + \\
&\quad \sum_{\{i,j,k,k',\ell,\ell'\}\in\binom{n}{6}, i<j, k<k'} O\left(\frac{1}{\varepsilon^2}\right)\cdot x_{j,k}\cdot x_{k,\ell}\cdot x_{\ell,i}\cdot x_{j,k'}\cdot x_{k',\ell'}\cdot x_{\ell',i}\Bigg) \\
&= O\left(\frac{n^4}{\varepsilon^8}\right) + O\left(\frac{n^2d^3}{\varepsilon^4}\right) + O\left(\frac{nd^5}{\varepsilon^2}\right)
\end{aligned}
$$

When $\varepsilon = \Theta(1)$, the variance can be simplified to $O(n^4 + nd^5)$. □