Locally Differentially Private Analysis of Graph Statistics

Jacob Imola
UC San Diego

Takao Murakami
AIST

Kamalika Chaudhuri
UC San Diego

Abstract

Differentially private analysis of graphs is widely used for releasing statistics from sensitive graphs while still preserving user privacy. Most existing algorithms however are in a centralized privacy model, where a trusted data curator holds the entire graph. As this model raises a number of privacy and security issues – such as, the trustworthiness of the curator and the possibility of data breaches, it is desirable to consider algorithms in a more decentralized local model where no server holds the entire graph.

In this work, we consider a local model, and present algorithms for counting subgraphs – a fundamental task for analyzing the connection patterns in a graph – with LDP (Local Differential Privacy). For triangle counts, we present algorithms that use one and two rounds of interaction, and show that an additional round can significantly improve the utility. For \( k \)-star counts, we present an algorithm that achieves an order optimal estimation error in the non-interactive local model. We provide new lower-bounds on the estimation error for general graph statistics including triangle counts and \( k \)-star counts. Finally, we perform extensive experiments on two real datasets, and show that it is indeed possible to accurately estimate subgraph counts in the local differential privacy model.

1 Introduction

Analysis of network statistics is a useful tool for finding meaningful patterns in graph data, such as social, e-mail, citation and epidemiological networks. For example, the average degree (i.e., number of edges connected to a node) in a social graph can reveal the average connectivity, and subgraph counts (e.g., the number of triangles, stars, or cliques) can be used to measure centrality properties such as the clustering coefficient of a graph. However, the vast majority of graph analytics is carried out on sensitive data, which could be leaked through the results of graph analysis. Thus, there is a need to develop solutions that can analyze these graph properties while still preserving the privacy of individual people in the network.

The standard way to analyze graphs with privacy is through differentially private graph analysis [19, 20, 42]. Differential privacy provides individual privacy against adversaries with arbitrary background knowledge, and has currently emerged as the gold standard for private analytics. However, a vast majority of differentially private graph analysis algorithms are in the centralized (or global) model [10, 12, 13, 22, 28, 30, 36, 41, 42, 45, 49, 50], where a single trusted data curator holds the entire graph and releases sanitized versions of the statistics. By assuming a trusted party that can access the entire graph, it is possible to release accurate graph statistics (e.g., subgraph counts [28, 30, 45], degree distribution [13, 22, 41], spectra [50]) and synthetic graphs [12, 49].

In many applications however, a single trusted curator may not be practicable due to security or logistical reasons. A centralized data holder is amenable to security issues such as data breaches and leaks – a growing threat in recent years [33, 44]. Additionally, decentralized social networks [37, 43] (e.g., Diaspora [4]) have no central server that contains an entire social graph, and uses instead many servers all over the world, each containing the data of users who have chosen to register there. Finally, a centralized solution is also not applicable to fully decentralized applications, where the server does not automatically hold information connecting users. An example of this is a mobile application that asks each user how many of their friends they have seen today, and sends noisy counts to a central server. In this application, the server does not hold any individual edge, but can still aggregate the responses to determine the average mobility in an area.

The standard privacy solution that does not assume a trusted third party is LDP (Local Differential Privacy) [17, 29]. This is a special case of DP in the local model, where each user obfuscates her personal data by herself and sends the obfuscated data to a (possibly malicious) data collector. Since the data collector does not hold the original personal data, the risk of data breach is significantly reduced.
data, it does not suffer from data leakage issues. Therefore, LDP has recently attracted attention from both academia [6–8, 21, 26, 27, 34, 38, 48, 53] as well as industry [14, 46, 47]. However, the use of LDP has mostly been in the context of tabular data where each row corresponds to an individual, and little attention has been paid to LDP for more complex data such as graphs (see Section 2 for details).

In this paper, we consider LDP for graph data, and provide algorithms and theoretical performance guarantees for calculating graph statistics in this model. In particular, we focus on counting triangles and $k$-stars – the most basic and useful subgraphs. A triangle is a set of three nodes with three edges (we exclude automorphisms; i.e., $\#closed$ triplets $= 3 \times \#triangles$). A $k$-star consists of a central node connected to $k$ other nodes. Figure 1 shows an example of triangles and $k$-stars. Counting them is a fundamental task of analyzing the connection patterns in a graph, as measures of centrality such as the clustering coefficient can be calculated from triangle and 2-star counts as: $\frac{\#triangles}{\#2-stars}$ (in Figure 1, $\frac{3}{20} = 0.15$).

The main challenge in counting subgraphs in the local model is that existing techniques and their analysis do not directly apply. The existing work on LDP for tabular data assumes that each person’s data is independently and identically drawn from an underlying distribution. In graphs, this is no longer the case; e.g., each triangle is not independent, because multiple triangles can involve the same edge; each $k$-star is not independent for the same reason. Moreover, complex inter-dependencies involving multiple people are possible in graphs. For example, each user cannot count triangles involving herself, because she cannot see edges between other users; e.g., in Figure 1, user (node) $v_1$ cannot see an edge between $v_3$ and $v_4$.

We show that although these complex dependency among users introduces challenges, it also presents opportunities. Specifically, this kind of interdependency also implies that extra interaction between users and a data collector may be helpful depending on the prior responses. In this work, we investigate this issue and provide algorithms for accurately calculating subgraph counts under LDP.

Our contributions. To our knowledge, we are the first to provide algorithms and corresponding performance guarantees for counting triangles and $k$-stars in graphs under edge Local Differential Privacy. Specifically, our contributions are as follows:

- For triangles, we present two algorithms. The first is based on Warner’s RR (Randomized Response) [51] and empirical estimation [26, 34, 48]. We then present a more sophisticated algorithm that uses an additional round of interaction between users and data collector. We provide upper-bounds on the estimation error for each algorithm, and show that the latter can significantly reduce the estimation error.

- For $k$-stars, we present a simple algorithm using the Laplacian mechanism. We analyze the upper-bound on the estimation error for this algorithm, and show that it is order optimal in terms of the number of users among all LDP mechanisms that do not use additional interaction.

- We provide lower-bounds on the estimation error for estimating general graph functions including triangle counts and $k$-star counts in the local model. These are stronger than known upper bounds in the centralized model, and illustrate the limitations of the local model over the central.

- Finally, we evaluate our algorithms on two real datasets, and show that it is indeed possible to accurately estimate subgraph counts in the local model. In particular, we show that the interactive algorithm for triangle counts and the Laplacian algorithm for the $k$-stars provide small estimation errors when the number of users is large.

2 Related Work

Here we review the previous work on graph DP (Differential Privacy), LDP (Local DP), and upper/lower-bounds.

Graph DP. DP on graphs has been widely studied, with most prior work being in the centralized model [10, 12, 13, 22, 28, 30, 36, 41, 42, 45, 49, 50]. In this model, a number of algorithms have been proposed for releasing subgraph counts [28, 30, 45], degree distributions [13, 22, 41], eigenvalues and eigenvectors [50], and synthetic graphs [12, 49].

There has also been a handful of work on graph algorithms in the local DP model [39, 54, 55]. For example, Qin et al. [39] propose an algorithm for generating synthetic graphs, while Ye et al. [54] provide a method for graph metric estimation under LDP. Zhang et al. [55] propose an algorithm for software usage analysis under LDP, where a node represents a software component (e.g., function in a code) and an edge represents a control-flow between components. None of these works focus on subgraph counts.

Our work differs from these in two ways: (i) our work provides algorithms and theoretical performance guarantees for subgraph counts, (ii) we also lower-bounds on the estimation error. We note that although Warner’s RR (Randomized Response) has been applied to an adjacency matrix in [39, 54] for different purposes than ours, it can be used for counting triangles. However, it suffers from a very large estimation error.
Our one-round algorithm for triangles uses empirical estimation as a post-processing step, and we show in Appendix A that this empirical estimation step significantly reduces the estimation error.

**LDP.** Apart from graphs, a number of works have looked at analyzing statistics (e.g., discrete distribution estimation [6, 21, 26, 27, 34, 48, 53] and heavy hitters [7, 8, 38]) under LDP.

However, they use LDP in the context of tabular data, and do not consider the kind of complex interdependency in graph data (as described in Section 1). For example, the RR with empirical estimation is optimal in the low privacy regimes for estimating a distribution for tabular data [26, 27]. We apply the RR and empirical estimation to counting triangles, and show that it is suboptimal and significantly outperformed by a more sophisticated algorithm with more interaction between users and a data collector.

**Upper/lower-bounds.** Finally, we note that existing work on upper-bounds and lower-bounds cannot be directly applied to our setting. For example, there are upper-bounds [6, 23, 24, 26, 27, 53] and lower-bounds [5, 15, 16, 18, 23–25] on the estimation error (or sample complexity) in distribution estimation of tabular data. However, they assume that each original data value is independently sampled from an underlying distribution. They cannot be directly applied to our graph setting, because each triangle and each k-star involve multiple edges and are not independent (as described in Section 1). Rashchian et al. [40] provide lower-bounds on communication complexity (i.e., number of queries) of vector-matrix-vector queries for estimating subgraph counts. However, their lower-bounds are not on the estimation error, and cannot be applied to our problem.

## 3 Preliminaries

### 3.1 Graphs and Differential Privacy

**Graphs.** Let \( \mathbb{N}, \mathbb{Z}_{\geq 0}, \mathbb{R}, \) and \( \mathbb{R}_{\geq 0} \) be the sets of natural numbers, non-negative integers, real numbers, and non-negative real numbers, respectively. For \( a \in \mathbb{N} \), let \( [a] = \{1, 2, \ldots, a\} \).

We consider an undirected graph \( G = (V, E) \), where \( V \) is a set of nodes (i.e., users) and \( E \) is a set of edges. Let \( n \in \mathbb{N} \) be the number of users in \( V \), and let \( v_i \in V \) the \( i \)-th user; i.e., \( V = \{v_1, \ldots, v_n\} \). An edge \( (v_i, v_j) \in E \) represents a relationship between users \( v_i \in V \) and \( v_j \in V \). The number of edges connected to a single node is called the degree of the node. Let \( d_{\text{max}} \in \mathbb{N} \) be the maximum degree (i.e., maximum number of edges connected to a node) in graph \( G \). Let \( G' \) be the set of possible graphs \( G \) on \( n \) users. A graph \( G \in \mathcal{G} \) can be represented as a symmetric adjacency matrix \( A = (a_{i,j}) \in \{0, 1\}^{n \times n} \), where \( a_{i,j} = 1 \) if \( (v_i, v_j) \in E \) and \( a_{i,j} = 0 \) otherwise.

**Types of DP.** DP (Differential Privacy) [19, 20] is known as a gold standard for data privacy. According to the underlying architecture, DP can be divided into two types: centralized DP and LDP (Local DP). Centralized DP assumes the centralized model, where a “trusted” data collector collects the original personal data from all users and obfuscates a query (e.g., counting query, histogram query) on the set of personal data. LDP assumes the local model, where each user does not trust even the data collector. In this model, each user obfuscates a query on her personal data by herself and sends the obfuscated data to the data collector.

If the data are represented as a graph, we can consider two types of DP: edge DP and node DP [22, 42]. Edge DP considers two neighboring graphs \( G, G' \in \mathcal{G} \) that differ in one edge. In contrast, node DP considers two neighboring graphs \( G, G' \in \mathcal{G} \) in which \( G' \) is obtained from \( G \) by adding or removing one node along with its adjacent edges. Although node DP guarantees stronger privacy than edge DP, it is much harder to attain. Zhang et al. [55] proposed an algorithm for software usage analysis with node DP in the local model, where a node represents a software component and an edge represents a control-flow between components. However, we consider a totally different problem, where each node represents a user (rather than a software component). In this case, achieving node DP in the local model is extremely difficult, because each user needs to hide the existence of herself along with her all edges against the data collector. Thus we focus on edge DP in the same way as [39].

Below we explain edge DP in the centralized model.

**Centralized DP.** We call edge DP in the centralized model edge centralised DP. Formally, it is defined as follows:

**Definition 1 (\( \epsilon \)-edge centralised DP).** Let \( \epsilon \in \mathbb{R}_{\geq 0} \). A randomised algorithm \( \mathcal{M} \) with domain \( \mathcal{G} \) provides \( \epsilon \)-edge centralised DP if for any two neighboring graphs \( G, G' \in \mathcal{G} \) that differ in one edge and any \( S \subseteq \text{Range}(\mathcal{M}) \),

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

Edge centralised DP guarantees that an adversary who has observed the output of \( \mathcal{M} \) cannot determine whether it is come from \( G \) or \( G' \) with a certain degree of confidence. The parameter \( \epsilon \) is called the privacy budget. If \( \epsilon \) is close to zero, then \( G \) and \( G' \) are almost equally likely, which means that an edge in \( G \) is strongly protected.

We also note that edge DP can be used to protect \( \epsilon \in \mathbb{N} \) edges by using the composition theorem [20]. Specifically, if \( \mathcal{M} \) provides \( \epsilon \)-edge centralised DP, then for any two graphs \( G, G' \in \mathcal{G} \) that differ in \( k \) edges and any \( S \subseteq \text{Range}(\mathcal{M}) \), we obtain:

\[
\Pr[\mathcal{M}(G) \in S] \leq e^{\epsilon k} \Pr[\mathcal{M}(G') \in S]; \text{ i.e., } k \text{ edges are protected with privacy budget } ke.
\]

### 3.2 Local Differential Privacy

LDP (Local Differential Privacy) [17, 29] is a privacy metric to protect personal data of each user in the local model. LDP has
been originally introduced to protect each user’s data record that is independent from the other records. However, in a graph, each edge is connected to two users. Thus, when we define edge DP in the local model, we should consider what we want to protect. In this paper, we consider two definitions of edge DP in the local model: edge LDP in [39] and entire edge LDP introduced in this paper. Below, we will explain these two definitions in detail.

**Edge LDP.** Qin et al. [39] defined edge LDP based on a user’s neighbor list. Specifically, let \(a_i = (a_{i,1}, \ldots, a_{i,n}) \in \{0,1\}^n\) be a neighbor list of user \(v_i\). Note that \(a_i\) is the \(i\)-th row of the adjacency matrix \(A\) of graph \(G\). In other words, graph \(G\) can be represented as neighbor lists \(a_1, \ldots, a_n\).

Then edge LDP is defined as follows:

**Definition 2** (\(\varepsilon\)-edge LDP [39]). Let \(\varepsilon \in \mathbb{R}_{\geq 0}\). For any \(i \in [n]\), let \(R_i\) with domain \(\{0,1\}^n\) be a randomized algorithm of user \(v_i\). \(R_i\) provides \(\varepsilon\)-edge LDP if for any two neighbor lists \(a_i, a'_i \in \{0,1\}^n\) that differ in one bit and any \(S \subseteq \text{Range}(R_i)\),

\[
\Pr[R_i(a_i) \in S] \leq e^\varepsilon \Pr[R_i(a'_i) \in S].
\]  

(2)

Edge LDP in Definition 2 protects a single bit in a neighbor list with privacy budget \(\varepsilon\). As with edge centralized DP, edge LDP can also be used to protect \(k \in \mathbb{N}\) bits in a neighbor list by using the composition theorem; i.e., \(k\) bits in a neighbor list are protected with privacy budget \(k\varepsilon\).

**RR (Randomized Response).** As a simple example of a randomized algorithm \(R_i\) providing \(\varepsilon\)-edge LDP, we explain Warner’s RR (Randomized Response) [51] applied to a neighbor list, which is called the randomized neighbor list in [39].

Given a neighbor list \(a_i \in \{0,1\}^n\), this algorithm outputs a noisy neighbor lists \(b = (b_1, \ldots, b_n) \in \{0,1\}^n\) by flipping each bit in \(a_i\) with probability \(p = \frac{1}{\varepsilon + 1}\); i.e., for each \(j \in [n]\), \(b_j = a_{i,j}\) with probability \(p\) and \(b_j = 1 - a_{i,j}\) with probability \(1 - p\). Thus, \(\Pr[R_i(a_i) \in S] = \Pr[R_i(a'_i) \in S]

(2)\) differ by \(e^\varepsilon\) for \(a_i\) and \(a'_i\) that differ in one bit, this algorithm provides \(\varepsilon\)-edge LDP.

**Entire edge LDP.** We can also define edge LDP in the local model to protect one edge in graph \(G\) during the whole process. In this paper, we call this definition entire edge LDP to discriminate from edge LDP in [39].

We define entire edge LDP as follows:

**Definition 3** (\(\varepsilon\)-entire edge LDP). Let \(\varepsilon \in \mathbb{R}_{\geq 0}\). A tuple of randomized algorithms \((R_1, \ldots, R_n)\), each of which is with domain \(\{0,1\}^n\), provides \(\varepsilon\)-entire edge LDP if for any two neighboring graphs \(G, G' \in G\) that differ in one edge and any \(S \subseteq \text{Range}(R_i) \times \cdots \times \text{Range}(R_n)\),

\[
\Pr[(R_1(a_1), \ldots, R_n(a_n)) \in S]\leq e^\varepsilon \Pr[(R_1(a'_1), \ldots, R_n(a'_n)) \in S],
\]  

(3)

where \(a_i\) (resp. \(a'_i\)) \(\in \{0,1\}^n\) is the \(i\)-th row of the adjacency matrix of graph \(G\) (resp. \(G'\)).

**Proposition 1.** If randomized algorithms \(R_1, \ldots, R_n\) provide \(\varepsilon\)-edge LDP, then \((R_1, \ldots, R_n)\) provides \(2\varepsilon\)-entire edge LDP.

**Proof.** The existence of edge \((v_i, v_j) \in E\) affects two elements \(a_{i,j}, a_{j,i} \in \{0,1\}\) in the adjacency matrix \(A\). Then by the composition theorem [20], Proposition 1 holds.

Proposition 1 states that when we want to protect one edge as a whole, the privacy budget is at most doubled. Note, however, that some randomized algorithms do not have this doubling issue. For example, we can apply the RR to the \(i\)-th neighbor list \(a_i\) so that \(R_i\) outputs noisy bits \((b_1, \ldots, b_{i-1}) \in \{0,1\}^{i-1}\) for only users \(v_1, \ldots, v_{i-1}\) with smaller user IDs; i.e., for each \(j \in \{1, \ldots, i-1\}\), \(b_j \neq a_{i,j}\) with probability \(p = \frac{1}{\varepsilon + 1}\) and \(b_j = a_{i,j}\) with probability \(1 - p\). In other words, we can extend the RR for a neighbor list so that \((R_1, \ldots, R_n)\) outputs only the lower triangular part of the noisy adjacency matrix. Then all of \(R_1, \ldots, R_n\) provide \(\varepsilon\)-edge LDP. In addition, the existence of edge \((v_i, v_j) \in E\) \((i > j)\) affects only one element \(a_{i,j}\) in the lower triangular part of \(A\). Thus, \((R_1, \ldots, R_n)\) provides \(\varepsilon\)-entire edge LDP (not \(2\varepsilon\)).

Our proposed algorithm in Section 4.3 also has this property; i.e., it provides both \(\varepsilon\)-edge LDP and \(\varepsilon\)-entire edge LDP.

### 3.3 Global Sensitivity and Local Sensitivity

In this paper, we use the local sensitivity [36] to provide edge centralized DP or edge LDP with small noise. Here we explain the global sensitivity [20] and the local sensitivity [36].

Let \(D\) be the set of possible input data of a randomized algorithm. In edge centralized DP, \(D = G\). In edge LDP, \(D = \{0,1\}^n\). Let \(f: D \rightarrow \mathbb{R}\) be a function that takes data \(D \in D\) as input and outputs some statistics \(f(D) \in \mathbb{R}\) about the data. The most basic method for providing DP is to add the Laplacian noise proportional to the global sensitivity [20].

**Definition 4 (Global sensitivity).** The global sensitivity of a function \(f: D \rightarrow \mathbb{R}\) is given by:

\[GS_f = \max_{D,D' \in D,D' \sim D} |f(D) - f(D')|,\]

where \(D \sim D'\) represents that \(D\) and \(D'\) are neighbors; i.e., they differ in one edge in edge centralized DP, and differ in one bit in edge LDP.

In practice, the global sensitivity \(GS_f\) can be very large. Nissim et al. [36] introduced a local measure of sensitivity called the local sensitivity to address this issue.

**Definition 5 (Local sensitivity [36]).** The local sensitivity of a function \(f: D \rightarrow \mathbb{R}\) at \(D \in D\) is given by:

\[LS_f(D) = \max_{D' \in D,D' \sim D} |f(D) - f(D')|,\]

Note that \(GS_f = \max_{D \in D} LS_f(D)\). In practice, \(LS_f(D)\) can be much smaller than \(GS_f\). For example, the local sensitivity
of triangle counts in $G$ is at most the maximum degree $d_{\text{max}}$, which is much smaller than $G S_f = n - 2$ when $G$ is sparse.

The local sensitivity $L S_f (D)$ cannot be directly used, because the noise magnitude can leak some information about $G$. Karwa et al. [28] showed that in the centralized graph model, adding the Cauchy noise (rather than the Laplacian noise) with the local sensitivity to $k$-star or triangle counts in $G$ provides $\varepsilon$-edge central DP under some conditions. However, in the local graph model, it is even difficult for users to know the local sensitivity itself. In this paper, we address this issue by privately estimating $L S_f (D)$ with edge LDP and then applying graph projection [13, 30, 41], which removes some neighbors from a neighbor list, so that the local sensitivity is upper-bounded by the private estimate of $L S_f (D)$.

### 3.4 Graph Statistics and Utility Metrics

**Graph statistics.** We consider a graph function that takes a graph $G \in \mathcal{G}$ as input and outputs some graph statistics. Specifically, let $f_\Delta : \mathcal{G} \rightarrow \mathbb{Z}_{\geq 0}$ be a graph function that outputs the number of triangles in $G$. For $k \in \mathbb{N}$, let $f_k : \mathcal{G} \rightarrow \mathbb{Z}_{\geq 0}$ be a graph function that outputs the number of $k$-stars in $G$. For example, if a graph $G$ is as shown in Figure 1, then $f_\Delta (G) = 5$, $f_2 (G) = 20$, and $f_4 (G) = 8$. The clustering coefficient can also be calculated from $f_\Delta (G)$ and $f_2 (G)$ as: $\frac{2f_2 (G)}{f_\Delta (G)} = 0.75$.

Table 1 shows the basic notations used in this paper.

**Utility metrics.** We use the $l_2$ loss (i.e., squared error) [26, 34, 48] and the relative error [9, 11, 52] as utility metrics. Specifically, let $\hat{f} (G) \in \mathbb{R}$ be an estimate of graph statistics $f (G) \in \mathbb{R}$. Here $f$ can be instantiated by $f_\Delta$ or $f_k$: i.e., $\hat{f}_\Delta (G)$ and $\hat{f}_k (G)$ are the estimates of $f_\Delta (G)$ and $f_k (G)$, respectively. Let $l_2^2$ be the $l_2$ loss function, which maps the estimate $\hat{f} (G)$ and the true value $f (G)$ to the $l_2$ loss; i.e., $l_2^2 (\hat{f} (G), f (G)) = (\hat{f} (G) - f (G))^2$.

Note that when we use a randomized algorithm providing edge LDP (or edge centralized DP), $\hat{f} (G)$ depends on the randomness in the algorithm. In our theoretical analysis, we analyze the expectation of the $l_2$ loss over the randomness, as with [26, 34, 48].

When $f (G)$ is large, the $l_2$ loss can also be large. Thus in our experiments, we also evaluate the relative error, along with the $l_2$ loss. The relative error is defined as: $\frac{|\hat{f} (G) - f (G)|}{\max (f (G)) \eta}$, where $\eta \in \mathbb{R}_{\geq 0}$ is a very small positive value. Following the convention [9, 11, 52], we set $\eta = 0.001n$.

### 4 Algorithms

When the data collector does not have access to the adjacency matrix $A$, a unique communication challenge arises. There are several ways in which we may model how the data collector interacts with the users [17, 25, 39]. The simplest model would be to assume that the data collector sends one query $R_i$ to each user $v_i$ once and no communication among users occurs. The data collector would then receive independent copies of the random variables $(R_1 (a_1), \ldots, R_n (a_n))$. In this model, there is one-round interaction between each user and the data collector. We call this the one-round LDP model. For example, the RR (Randomized Response) for a neighbor list in Section 3.2 assumes this model.

However, in certain cases it may be possible for the data collector to send a query to each user multiple times. This could allow for more powerful queries that result in more accurate analysis of graph statistics or more accurate synthetic graphs [39]. We call this the multiple-rounds LDP model. For example, a synthetic graph generation technique in the two-rounds LDP model has been proposed in [39].

In Sections 4.1 and 4.2, we consider the problems of computing $f_k (G)$ and $f_\Delta (G)$ for a graph $G \in \mathcal{G}$ in the one-round LDP model. These problems are simple but are commonly used to understand the structure of $G$. The algorithms and bounds we have also highlight limitations of the one-round LDP model. Compared to the centralized graph DP model, the one-round LDP model cannot compute $f_k (G)$ as accurately. Furthermore, the algorithm for $f_\Delta (G)$ does not perform well. In Section 4.3, we propose a more sophisticated algorithm for computing $f_\Delta (G)$ in the two-rounds LDP model, and show that it provides much smaller expected $l_2$ loss than the algorithm in the one-round LDP model. In Section 4.4, we show a general result about lower bounds on the expected $l_2$ loss of graph statistics in LDP. The proofs of all statements in Section 4 are given in Appendix B.

#### 4.1 One-Round LDP Algorithms for $k$-Stars

**Algorithm.** We begin with the problem of computing $f_k (G)$ in the one-round LDP model. For this model, we introduce a simple algorithm using the Laplacian mechanism, and prove that this algorithm can achieve order optimal expected $l_2$ loss among all one-round LDP algorithms.

Algorithm 1 shows the one-round algorithm for $k$-stars. It takes as input a graph $G$ (represented as neighbor lists $a_1, \ldots, a_n \in \{0, 1\}^n$), the privacy budget $\varepsilon$, and a non-negative integer $d_{\text{max}} \in \mathbb{Z}_{\geq 0}$.

| Symbol | Description |
|--------|-------------|
| $G = (V, E)$ | Graph with $n$ nodes (users) $V$ and edges $E$ |
| $v_i$ | $i$-th user in $V$. |
| $d_{\text{max}}$ | Maximum degree of $G$. |
| $\mathcal{G}$ | Set of possible graphs on $n$ users. |
| $A$ | Adjacency matrix. |
| $a_i$ | $i$-th row of $A$ (i.e., neighbor list of $v_i$). |
| $R_i$ | Randomized algorithm on $a_i$. |
| $f_\Delta (G)$ | Number of triangles in $G$. |
| $f_k (G)$ | Number of $k$-stars in $G$. |
We denote this algorithm by \text{LocalLap}. Another example, if we know that the degree is smaller than \(\epsilon\)-edge about the maximum degree \(\Delta\). Therefore, we add \(\epsilon\)-Laplacian noise to \(k\)-star counts for user \(v\). Then it selects \(\hat{r}_i\) randomly generates a permutation of 1,...,\(n\) (the random seed can be different from user to user). Then it selects \(\hat{d}_{max}\) from her neighbor list \(a_i\). In line 3, user \(v_i\) uses a function (denoted by \text{GraphProjection}) that performs graph projection \([13,30,41]\) for \(a_i\), as follows. If \(d_i\) exceeds \(\hat{d}_{max}\), it randomly generates a permutation of 1,...,\(n\) (the random seed can be different from user to user). Then it selects \(\hat{d}_{max}\) neighbors from her neighbor list \(a_i\) in the order of the permutation. For example, if \(n = 6\), \(a_1 = \{0,1,0,1,1,1\}\), \(\hat{d}_{max} = 3\), and the permutation is 2,3,4,1,6,5, then it selects the second, fourth, and sixth users; i.e., \(a_1\) becomes \(a_1 = \{0,1,0,1,0,1\}\) after the graph projection. This guarantees that each user’s degree never exceeds \(\hat{d}_{max}\); i.e., \(d_i \leq \hat{d}_{max}\) after line 4. After the graph projection, user \(v_i\) counts the numbers of \(k\)-stars \(r_i\) in \(Z_{\geq 0}\) of which she is a center (line 5), and adds the Laplacian noise to \(r_i\). Here, since adding one edge results in the increase of at most \(\hat{d}_{max}\) \(k\)-stars, the sensitivity of \(k\)-star counts for user \(v_i\) is at most \(\frac{\hat{d}_{max}}{k-1}\) (after graph projection). Therefore, we add \(\text{Lap}(\Delta)\) to \(r_i\), where \(\Delta = \left(\frac{\hat{d}_{max}}{k-1}\right)\) and for \(b \in R_{\geq 0}\), \(\text{Lap}(b)\) is a random variable that represents the Laplacian noise with mean 0 and scale \(b\). The final answer of Algorithm 1 is simply the sum of all the noisy \(k\)-star counts. We denote this algorithm by \text{LocalLap}. The value of \(\hat{d}_{max}\) greatly affects the utility. If \(\hat{d}_{max}\) is too large, a large amount of the Laplacian noise would be added. However, if \(\hat{d}_{max}\) is too small, a great number of neighbors would be reduced by graph projection. When we have some prior knowledge about the maximum degree \(d_{max}\), we can set \(\hat{d}_{max}\) to an appropriate value. For example, the maximum number of connections allowed on Facebook is 5000 [2]. In this case, we can set \(\hat{d}_{max} = 5000\), and then graph projection does nothing. For another example, if we know that the degree is smaller than 1000 for most users, then we can set \(\hat{d}_{max} = 1000\) and perform graph projection for users whose degrees exceed \(\hat{d}_{max}\). In some applications, the data collector may not have such prior knowledge about \(d_{max}\). In this case, we can privately estimate \(d_{max}\) by allowing an additional round between each user and the data collector, and use the private estimate of \(d_{max}\) as \(\hat{d}_{max}\). We describe how to privately estimate \(d_{max}\) with edge LDP at the end of Section 4.1.

**Theoretical properties.** \text{LocalLap} has the following guarantees:

**Theorem 1.** \text{LocalLap} provides \(\epsilon\)-edge LDP.

**Theorem 2.** Let \(\hat{f}_{k*}(G,\epsilon,\hat{d}_{max})\) be the output of \text{LocalLap}. Then, for all \(k \in \mathbb{N}\), \(\epsilon \in \mathbb{R}_{\geq 0}\), \(d_{max} \in Z_{\geq 0}\), and \(G \in \mathcal{G}\) such that the maximum degree \(d_{max}\) of \(G\) is at most \(\hat{d}_{max}\), \(E[\hat{f}_{k*}(G,\epsilon,\hat{d}_{max}), f_{k*}(G)] = O \left( \frac{n \hat{d}_{max}^2}{2^n \epsilon^2} \right) \).

The factor of \(n\) in the expected \(l_2\) loss of \text{LocalLap} comes from the fact that we are adding the Laplacian noise for \(n\) times. In the centralized model, this factor of \(n\) is not there because the central data collector sees all \(k\)-stars; i.e., the data collector knows \(f_{k*}(G)\). The sensitivity of \(f_{k*}\) is at most \(2 \left( \frac{\hat{d}_{max}}{k-1} \right)\) (after graph projection) under edge centralized DP. Therefore, we can consider an algorithm that simply adds the Laplacian noise \(\text{Lap}(2 \left( \frac{\hat{d}_{max}}{k-1} \right) / \epsilon)\) to \(f_{k*}(G)\), and outputs \(f_{k*}(G) + \text{Lap}(2 \left( \frac{\hat{d}_{max}}{k-1} \right) / \epsilon)\). We denote this algorithm by \text{CentralLap}. Since the bias of the Laplacian noise is 0, \text{CentralLap} attains the expected \(l_2\) loss (= variance) of \(O \left( \frac{n \hat{d}_{max}^2}{2^n \epsilon^2} \right)\).

It seems impossible to avoid this factor of \(n\) in the one-round LDP model, as the data collector will be dealing with \(n\) independent answers to queries. Indeed, this is the case—we prove that the expected \(l_2\) error of \text{LocalLap} is order optimal among all one-round LDP algorithms, and the one-round LDP model cannot improve the factor of \(n\).

**Corollary 1.** Let \(\hat{f}_{k*}(G,\hat{d}_{max},\epsilon)\) be any one-round LDP algorithm that computes \(f_{k*}(G)\) satisfying \(\epsilon\)-edge LDP. Then, for all \(k,n\), \(\hat{d}_{max} \in \mathbb{N}\) and \(\epsilon \in \mathbb{R}_{\geq 0}\) such that \(n\) is even, there exists a set of graphs \(\mathcal{G}\) on \(n\) nodes such that the maximum degree of each \(G \in \mathcal{G}\) is at most \(\hat{d}_{max}\), and \(\frac{1}{|\mathcal{G}|} \sum_{G \in \mathcal{G}} E[l_2^2(\hat{f}_{k*}(G,\hat{d}_{max},\epsilon), f_{k*}(G))] \geq \Omega \left( \min\{1, \frac{2^{k \epsilon^2 \hat{d}_{max}^2}}{(e^\epsilon - 1)^2} \} n^{2k-2} \right)\).

This is a corollary of a more general result of Section 4.4. Thus, any algorithm computing \(k\)-stars cannot avoid the factor of \(n\) in its \(l_2^2\) loss. \(k\)-stars is an example where the noninteractive graph LDP model is strictly weaker than the centralized DP model.

Nevertheless, we note that \text{LocalLap} can accurately calculate \(f_{k*}(G)\) for a large number of users \(n\). Specifically, the relative error decreases with increase in \(n\), because \text{LocalLap} has a factor of \(n\) (not \(n^2\)) in the expected \(l_2\) error, i.e., \(E[(f_{k*}(G,\epsilon,\hat{d}_{max}) - f_{k*}(G))^2] = O(n)\), and \(f_{k*}(G)^2 \geq \hat{d}_{max}^2\).
Algorithm. Now, we focus our attention on the more challenging \( f_\Delta \) query. This query is more challenging in the graph LDP model because no user is aware of any triangle; i.e., user \( v_i \) is not aware of any triangle formed by \((v_i, v_j, v_k)\) because \( v_i \) cannot see any edge \((v_j, v_k)\) in graph \( G \).

One way to count \( f_\Delta(G) \) with edge LDP is to apply the RR (Randomized Response) to a neighbor list. For example, user \( v_i \) applies the RR to \( a_{i,1}, \ldots, a_{i,j-1} \) (which corresponds to users \( v_1, \ldots, v_{j-1} \)) with smaller user IDs in her neighbor list \( a_i \), i.e., we apply the RR to the lower triangular part of adjacency matrix \( A \), as described in Section 3.2. Then the data collector constructs a noisy graph \( G' = (V, E') \in G \) from the lower triangular part of the noisy adjacency matrix, and estimates the number of triangles from \( G' \). However, simply counting the triangles in \( G' \) can introduce a significant bias because \( G' \) is denser than \( G \) especially when \( \varepsilon \) is small. The following extreme example illustrates this point: even if \( G \) is the empty graph, each edge in \( G' \) is generated with probability \( p = \frac{1}{(n-1)} \approx 0.5 \) when \( \varepsilon \) is close to zero. Thus, the number of triangles in \( G' \) will be extremely large.

\[
\Omega(n^2).
\]

In our experiments, we show that the relative error of \( \text{LocalLap}_k \) is very small when \( n \) is large.

**Private calculation of \( \hat{d}_{\text{max}} \).** By allowing an additional round between each user and the data collector, we can privately estimate \( d_{\text{max}} \) and use the private estimate of \( d_{\text{max}} \) as \( \hat{d}_{\text{max}} \). Specifically, we divide the privacy budget \( \varepsilon \) into \( \varepsilon_0 \in \mathbb{R}_{\geq 0} \) and \( \varepsilon_1 \in \mathbb{R}_{\geq 0} \); i.e., \( \varepsilon = \varepsilon_0 + \varepsilon_1 \). We first estimate \( d_{\text{max}} \) with \( 0 \)-edge LDP and then run \( \text{LocalLap}_k \) with the remaining privacy budget \( \varepsilon_1 \). Note that \( \text{LocalLap}_k \) with the private calculation of \( d_{\text{max}} \) results in a two-rounds LDP algorithm.

We consider the following simple algorithm. At the first round, each user \( v_i \) adds the Laplacian noise \( \text{Lap}(\frac{\varepsilon_i}{\varepsilon_0}) \) to her degree \( d_i \). Let \( \hat{d}_i \in \mathbb{R} \) be the noisy degree of \( v_i \); i.e., \( \hat{d}_i = d_i + \text{Lap}(\frac{\varepsilon_i}{\varepsilon_0}) \). Then user \( v_i \) sends \( \hat{d}_i \) to the data collector. Let \( \hat{d}_{\text{max}} \in \mathbb{R} \) be the maximum value of the noisy degree; i.e., \( \hat{d}_{\text{max}} = \text{max}\{\hat{d}_1, \ldots, \hat{d}_n\} \). We call \( \hat{d}_{\text{max}} \) the noisy max degree. The data collector calculates the noisy max degree \( \hat{d}_{\text{max}} \) as an estimate of \( d_{\text{max}} \), and sends \( \hat{d}_{\text{max}} \) back to all users. At the second round, we run \( \text{LocalLap}_k \) with input \( G \) (represented as \( a_1, \ldots, a_n \), \( \varepsilon \)), and use \( \lfloor \hat{d}_{\text{max}} \rfloor \).

At the first round, the calculation of \( \hat{d}_{\text{max}} \) provides \( \varepsilon_0 \)-edge LDP because each user’s degree has the sensitivity 1 under edge LDP. At the second round, Theorem 1 guarantees that \( \text{LocalLap}_k \) provides \( \varepsilon_1 \)-edge LDP. Then by the composition theorem [20], this two-rounds algorithm provides \( \varepsilon \)-edge LDP in total (\( \varepsilon = \varepsilon_0 + \varepsilon_1 \)).

In our experiments, we show that this algorithm provides the utility close to \( \text{LocalLap}_k \) with the true max degree \( d_{\text{max}} \).

### 4.2 One-Round LDP Algorithms for Triangles.

**Algorithm.** Let \( G = (V, E) \) be a noisy graph generated by applying the RR to the lower triangular part of \( A \).

\[
\text{LocalLap}_k = \begin{cases}
\text{LocalLap}_k(G) & \text{if } \text{tr}(G) \geq k \\
\text{LocalLap}_k(G) & \text{otherwise}
\end{cases}
\]

**Result:** Private estimate of \( f_\Delta(G) \).
$m_1$, and $m_0$ from $G'$. Finally, the data collector outputs the right-hand side of (4). We denote this algorithm by LocalRR$\Delta$.

**Theoretical properties.** LocalRR$\Delta$ provides the following guarantee.

**Theorem 3.** LocalRR$\Delta$ provides $\epsilon$-edge LDP and $\epsilon$-entire edge LDP.

LocalRR$\Delta$ does not have the doubling issue (i.e., it provides not 2$\epsilon$ but $\epsilon$-edge LDP) because we apply the RR to the lower triangular part of $A$, as explained in Section 3.2.

Unlike the RR and empirical estimation for tabular data [26], the expected $l_2$ loss of LocalRR$\Delta$ is complicated. To simplify the utility analysis, we assume that $G$ is generated from the Erdős-Rényi graph distribution $G(n, \alpha)$ with edge existence probability $\alpha$; i.e., each edge in $G$ with $n$ nodes is independently generated with probability $\alpha \in [0, 1]$.

**Theorem 4.** Let $G(n, \alpha)$ be the Erdős-Rényi graph distribution with edge existence probability $\alpha \in [0, 1]$. Let $p = \frac{1}{2^{1+\epsilon}}$ and $\beta = \alpha(1 - p) + (1 - \alpha)p$. Let $f_\Delta(G, \epsilon)$ be the output of LocalRR$\Delta$. If $G \sim G(n, \alpha)$, then for all $\epsilon \in \mathbb{R}_{\geq 0}$, 

$$
\mathbb{E}[f_\Delta^2(G, \epsilon), f_\Delta(G)] = O\left(\frac{e^\epsilon}{(e^\epsilon - 1)^2} \beta n^4\right).
$$

Note that we assume the Erdős-Rényi model only for the utility analysis of LocalRR$\Delta$, and do not assume this model for the other algorithms. The upper-bound of LocalRR$\Delta$ in Theorem 4 is less ideal than the upper-bounds of the other algorithms in that it does not consider all possible graphs $G \in \mathcal{G}$. Nevertheless, we also show that the $l_2$ loss of LocalRR$\Delta$ is roughly consistent with Theorem 4 in our experiments.

The parameters $\alpha$ and $\beta$ are edge existence probabilities in the original graph $G$ and noisy graph $G'$, respectively. Although $\alpha$ is very small in a sparse graph, $\beta$ can be large for small $\epsilon$. For example, if $\alpha \approx 0$ and $\epsilon = 1$, then $\beta \approx \frac{1}{2^{1+\epsilon}} = 0.27$.

Theorem 4 states that for large $n$, the $l_2$ loss of LocalRR$\Delta$ ($= O(n^4)$) is much larger than the $l_2$ loss of LocalRR$\Delta^*$ ($= O(n)$). This follows from the fact that user $v_i$ cannot see any edge $(v_j, v_k) \in E$ in graph $G$ and is not aware of any triangle formed by $(v_i, v_j, v_k)$.

In contrast, counting $f_\Delta(G)$ in the centralized model is much easier because the data collector sees all triangles in $G$; i.e., the data collector knows $f_\Delta(G)$. After we perform graph projection [13, 30, 41] so that each user’s degree does not exceed $\tilde{d}_{\text{max}}$, the sensitivity of $f_\Delta$ becomes at most $\tilde{d}_{\text{max}}$. Therefore, as with $k$-stars, we can consider a simple algorithm that outputs $f_\Delta(G) + \text{Lap}(\tilde{d}_{\text{max}}/\epsilon)$. We denote this algorithm by CentralLap$\Delta$. CentralLap$\Delta$ attains the expected $l_2$ loss ($= \text{variance}$) of $O\left(\frac{\tilde{d}_{\text{max}}}{\epsilon^2}\right)$.

The large $l_2$ loss of LocalRR$\Delta$ is caused by the fact that each edge is released independently with some probability of being flipped. In other words, there are three independent random variables that influence any triangle in $G'$. The next algorithm, using interaction, reduces this influencing number from three to one by using the fact that a user knows the existence of two edges for any triangle that involves the user.

### 4.3 Two-Rounds LDP Algorithms for Triangles

**Algorithm.** Allowing for two-rounds interaction, we are able to compute $f_\Delta$ with a significantly improved $l_2$ loss. As described in Section 4.2, it is impossible for user $v_i$ to see edge $(v_j, v_k) \in E$ in graph $G = (V, E)$ at the first round. However, if the data collector publishes a noisy graph $G' = (V, E')$ calculated by LocalRR$\Delta$ at the first round, then user $v_i$ can see a noisy edge $(v_j, v_k) \in E'$ in the noisy graph $G'$ at the second round. Then user $v_i$ can count the number of noisy triangles formed by $(v_j, v_j, v_k)$ such that $(v_j, v_j) \in E$, and $(v_j, v_k) \in E'$, and sends the noisy triangle counts with the Laplacian noise to the data collector in an analogous way to LocalLap$\Delta$. Since user $v_j$ always knows that two edges $(v_j, v_j)$ and $(v_j, v_k)$ exist in $G$, there is only one noisy edge in any noisy triangle (whereas all edges are noisy in LocalRR$\Delta$). This is an intuition behind our proposed two-rounds algorithm.

As with the RR in Section 4.2, simply counting the noisy triangles can introduce a bias. Therefore, we calculate an empirical estimate of $f_\Delta(G)$ from the noisy triangle counts. Specifically, the following is the empirical estimate of $f_\Delta(G)$:

**Proposition 3.** Let $G' = (V, E')$ be a noisy graph generated by applying the RR with privacy budget $\epsilon_1 \in \mathbb{R}_{\geq 0}$ to the lower triangular part of $A$. Let $p_1 = \frac{1}{2^{1+\epsilon_1}}$. Let $t_i \in \mathbb{Z}_{\geq 0}$ be the number of triplets $(v_i, v_j, v_k)$ such that $j < k < i$, $(v_i, v_j) \in E$, $(v_i, v_k) \in E$, and $(v_j, v_k) \in E'$. Let $s_i \in \mathbb{Z}_{\geq 0}$ be the number of triplets $(v_i, v_j, v_k)$ such that $j < k < i$, $(v_i, v_j) \in E$, and $(v_i, v_k) \in E$. Let $w_i = t_i - p_1 s_i$. Then

$$
\mathbb{E}\left[\frac{1}{1 - p_1} \sum_{i=1}^n w_i\right] = f_\Delta(G).
$$

Note that in Proposition 3, we count only triplets $(v_i, v_j, v_k)$ with $j < k < i$ to use only the lower triangular part of $A$. $t_i$ is the number of noisy triangles user $v_i$ can see at the second round. $s_i$ is the number of 2-stars of which user $v_i$ is a center. Since $t_i$ and $s_i$ can reveal information about an edge in $G$, user $v_i$ adds the Laplacian noise to $w_i (= t_i - p_1 s_i)$ in (5), and sends it to the data collector. Then the data collector calculates an unbiased estimate of $f_\Delta(G)$ by (5).

Algorithm 3 contains the formal description of this process. It takes as input a graph $G$ (represented as neighbor lists $a_1, \ldots, a_n$), the privacy budgets $\epsilon_1, \epsilon_2 \in \mathbb{R}_{\geq 0}$ at the first and second rounds, respectively, and a non-negative integer $d_{\text{max}} \in \mathbb{Z}_{\geq 0}$. At the first round, we apply the RR to the lower triangular part of $A$. At the second round, each user $v_i$ adds the Laplacian noise to $w_i (= t_i - p_1 s_i)$, whose sensitivity is at most $d_{\text{max}}$ (after graph projection), as we will prove in Theorem 5. We call this algorithm Local2Rounds$\Delta$.

**Theoretical properties.** Local2Rounds$\Delta$ has the following guarantee.
Theorem 5. Local2RoundsΔ provides (ε1 + ε2)-edge LDP and (ε1 + ε2)-edge LDP.

As with LocalRRΔ, Local2RoundsΔ does not have the doubling issue; i.e., it provides ε-edge LDP (not 2ε). This follows from the fact that we use only the lower triangular part of A; i.e., we assume j < k < i in counting ti and sj.

Theorem 6. Let \( \tilde{f}_Δ(G, ε_1, ε_2, d_{max}) \) be the output of Local2RoundsΔ. Then, for all ε1, ε2 ∈ ℝ≥0, \( d_{max} \in ℤ≥0 \), and \( G ∈ G \) such that the maximum degree \( d_{max} \) of \( G \) is at most \( d_{max} \), \( E[|f_{Δ}(G, ε_1, ε_2, d_{max})|] \) ≤ \( O\left(\frac{e^{ε_1^2}}{(1−e^{−ε_1})^2}\cdot d_{max}^3+\frac{e^{ε_2^2}}{ε_2^2}\cdot d_{max}^3\right) \)

Theorem 6 means that for triangles, the \( l_2 \) loss is reduced from \( O(n^4) \) to \( O(d_{max}^3) \) by introducing an additional round.

Private calculation of \( d_{max} \). As with k-stars, we can privately calculate \( d_{max} \) by using the method described in Section 4.1. Furthermore, the private calculation of \( d_{max} \) does not increase the number of rounds; i.e., we can run Local2RoundsΔ with the private calculation of \( d_{max} \) in two rounds.

Specifically, let \( ε_0 ∈ ℝ≥0 \) be the privacy budget for the private calculation of \( d_{max} \). At the first round, each user \( v_j \) adds \( \text{Lap}(\frac{ε_0}{ε_2}) \) to her degree \( d_j \), and sends the noisy degree \( \tilde{d}_j (= d_j + \text{Lap}(\frac{ε_0}{ε_2})) \) to the data collector, along with the outputs \( R_i = (RRG(a_{i1}),...,RRG(a_{i,n−1})) \) of the RR. The data collector calculates the noisy max degree \( \tilde{d}_{max} (= \max\{\tilde{d}_1,...,\tilde{d}_n\}) \) as an estimate of \( d_{max} \), and sends it back to all users. At the second round, we run Local2RoundsΔ with input \( \tilde{G} \) (represented as \( a_{i1},...,a_{in} \), \( ε_1, ε_2 \), and \( |d_{max}| \)).

At the first round, the calculation of \( \tilde{d}_{max} \) provides \( ε_0 \)-edge LDP. Note that it provides 2ε-edge entire LDP (i.e., it has the doubling issue) because one edge \( (v_i, v_j) ∈ E \) affects both of the degrees \( d_i \) and \( d_j \) by 1. At the second round, LocalLapΔ provides \((ε_1 + ε_2)\)-edge LDP and \((ε_1 + ε_2)\)-edge LDP (Theorem 5). Then by the composition theorem [20], this two-round algorithm provides \((ε_0 + ε_1 + ε_2)\)-edge LDP and \((2ε_0 + ε_1 + ε_2)\)-edge LDP. Although the total privacy budget is larger for entire edge LDP, the difference \( (= ε_0) \) can be very small. In fact, we set \((ε_0, ε_1, ε_2) = (0.1, 0.45, 0.45) \) or \((0.2, 0.9, 0.9) \) in our experiments (i.e., the difference is 0.1 or 0.2), and show that this algorithm provides almost the same utility as Local2RoundsΔ with the true max degree \( d_{max} \).

Time complexity. We also note that Local2RoundsΔ has an advantage over LocalRRΔ in terms of the time complexity in addition to the expected \( l_2 \) loss.

Specifically, LocalRRΔ is inefficient because the data collector has to count the number of triangles \( m_3 \) in the noisy graph \( G' \). Since the noisy graph \( G' \) is dense (especially when \( ε \) is small) and there are \((\frac{n^3}{2}) \) subgraphs with three nodes in \( G' \), the number of triangles is \( m_3 = O(n^3) \). Then, the time complexity of LocalRRΔ is also \( O(n^3) \), which is not practical for a graph with a large number of users \( n \). In fact, we implemented LocalRRΔ (ε = 1) with C/C++ and measured its running time using one node of a supercomputer (ABCI: AI Bridging Cloud Infrastructure [3]). When \( n = 5000, 10000, 20000, \) and 40000, the running time was 138, 1107, 9345, and 99561 seconds, respectively; i.e., the running time was almost cubic in \( n \). We can also estimate the running time for larger \( n \). For example, when \( n = 1000000 \), LocalRRΔ (ε = 1) would require about 35 years (= 1107 × 100^3 / (3600 × 24 × 365)).

In contrast, the time complexity of Local2RoundsΔ is \( (n^2 + nd_{max}^2) \), including the private calculation of \( d_{max} \). When we want to evaluate the utility of Local2RoundsΔ, we can apply the RR to only edges that are required at the second round; i.e., \((v_i, v_j) \in G' \) in line 8 of Algorithm 3. Then the time complexity can be reduced to \( O(nd_{max}^2) \) in total. We also confirmed that when \( n = 1000000 \), the running time of Local2RoundsΔ was 311 seconds (on one node the ABCI), which is \( 3.5 \times 10^6 \) times faster than LocalRRΔ.

4.4 Lower Bounds

We show a general lower bound on the \( l_2 \) loss of private estimators \( \tilde{f} \) of real-valued functions \( f \) in the one-round LDP model. Treating \( ε \) as a constant, we have shown that when \( d_{max} = d_{max} \), the expected \( l_2 \) loss of LocalLap\(ε_0\), is \( O(nd_{max}^2) \) (Theorem 2). However, in the centralized model, we can use the Laplace mechanism with sensitivity \( 2\tilde{d}_{max} \) in \( l_k \) to obtain \( l_2 \) errors of \( O(d_{max}^2) \) for \( f_k \). Thus, we wonder if the
factor of \( n \) is necessary in the one-round LDP model.

We answer this question. We show for many interesting queries \( f \), there is a lower bound for any private estimator \( \hat{f} \) on \( l_2(f, \hat{f}) \) that depends on \( n \). We require that \( f \) be monotone and increase by at least \( d \in \mathbb{N} \) over a set of input matrices.

That is, for \( A, A' \in \{0, 1\}^{n \times n} \), let \( \Delta \subseteq A' \) be the relation such that \( A' \) is obtained from \( A \) by changing some 0s to 1s. Then we require that if \( A \) and \( A' \) satisfies \( A \sim A' \) and \( \Delta \subseteq A' \), then \( f(A') - f(A) \geq D \). More specifically, we define the following cube-like structure on the input matrices:

**Definition 6.** \([(n, D)]\)-monotone cube for \( f \) Let \( D \in \mathbb{N} \). For \( k \in \mathbb{N} \), let \( G \in \mathcal{G} \) be a graph on \( n = 2k \) nodes and \( M = \{(v_1, v_2), (v_3, v_4), \ldots, (v_{2k-1}, v_{2k})\} \) for integers \( i_j \in [n] \) denote a perfect matching on the nodes, meaning each of \( i_1, \ldots, i_{2k} \) is distinct. Furthermore, suppose \( M \) is disjoint from \( G \), meaning \( (v_{2j-1}, v_{2j}) \notin G \) for any \( j \in [k] \). Let \( \mathcal{A} = \{G \cup N : N \subseteq M\} \). Notice \( \mathcal{A} \) is a set of \( 2^k \) adjacency matrices. We say \( \mathcal{A} \) defines an \((n, D)\) monotone cube for \( f \) if for all \( G_1, G_2 \in \mathcal{A} \) such that \( |G_1| + 1 = |G_2| \), we have \( f(G_2) - f(G_1) \geq D \).

For example, we can construct a monotone cube for a \( k \)-star function as follows. Assume \( d_{\text{max}} \) and \( n \) are both even. Let \( G \in \mathcal{G} \) be a \((d_{\text{max}} - 1)\)-regular graph of size \( n \), in which every node has degree \( d_{\text{max}} - 1 \). It is a standard result in graph theory that for any \( d \in \mathbb{N} \), \( d \)-regular graphs exist when \( n \) is even. Pick an arbitrary perfect matching \( M \) on the nodes. Now, let \( G' = G \setminus M \). Every node in \( G' \) has degree between \( d_{\text{max}} - 2 \) and \( d_{\text{max}} - 1 \). Adding an edge in \( M \) back to \( G \) will produce at least \((d_{\text{max}} - 2)^2 \) \( k \)-stars. Thus, \( \mathcal{A} = \{G' \cup N : N \subseteq M\} \) forms an \((n, (d_{\text{max}} - 2)^2)\) monotone cube for \( f_{k,*} \).

Using the cube-like structure of the \((n, D)\) monotone cube, we can prove a lower bound:

**Theorem 7.** Let \( f \) satisfy \( \epsilon \)-edge LDP in the one-round LDP model. Let \( \mathcal{A} \) be an \((n, D)\) monotone cube for \( f \). Let \( U(\mathcal{A}) \) be the uniform distribution over \( \mathcal{A} \). Then,

\[
\mathbb{E}_{\mathcal{A} \sim U(\mathcal{A})}[l_2^2(f(\mathcal{A}), \hat{f}(\mathcal{A}))] = \Omega(\min\{1, \left(\frac{2\epsilon^2}{(\epsilon^2 - 1)^2}\right)nD^2}).
\]

**Theorem 7.** combined with the fact that there exists an \((n, \left(\frac{d_{\text{max}} - 2}{k-1}\right))^2\) monotone cube for a \( k \)-star function implies Corollary 1. In Appendix B.11, we also construct an \((n, \left(\frac{d_{\text{max}}}{k-1}\right)^2)\) monotone cube for \( f_{\Delta} \) and establish a lower bound of \( \Omega(\min\{1, \left(\frac{2\epsilon^2}{(\epsilon^2 - 1)^2}\right)kd_{\text{max}}^2\}) \) for \( f_{\Delta} \).

The upper and lower bounds on the \( l_2 \) losses shown in this section appear in Table 2.

### 5 Experiments

Based on our theoretical results in Section 4, we would like to pose the following questions:

- For triangle counts, how much does the two-rounds interation help over a single round in practice?
- What is the privacy-utility trade-off for subgraph counts in the local model?

We conducted experiments to answer these questions.

#### 5.1 Experimental Setup

In our experiments, we used the following two large-scale datasets:

**IMDB.** The Internet Movie Database (denoted by IMDB) was used for the Graph Drawing 2005 contest [1]. It includes a bipartite graph between 896308 actors and 428440 movies, where an edge between an actor and a movie represents that the actor participated in the movie. We assumed actors as users. From the bipartite graph, we extracted a graph \( G' \) with 896308 nodes (actors), where an edge between two actors represents that they have played in the same movie. There are 57064358 edges in \( G' \), and the average degree in \( G' \) is 63.7 (\( \approx \frac{57064358}{896308} \)).

**Orkut.** Orkut is a social networking service operated by Google from 2004 to 2014. The Orkut online social network dataset (denoted by Orkut) [31] includes a graph \( G' \) with 3072441 users and 117185083 edges. The average degree in \( G' \) is 38.1 (\( \approx \frac{117185083}{3072441} \)). Therefore, Orkut is more sparse than IMDB (whose average degree in \( G' \) is 63.7).

For each dataset, we randomly selected \( n \) users from the whole graph \( G' \), and extracted a graph \( G = (V, E) \) with the \( n \) users. Then we estimated the number of triangles \( f_{\Delta}(G) \), the number of \( k \)-stars \( f_{k,*}(G) \), and the clustering coefficient \( \frac{3f_{k,*}(G)}{2f_{\Delta}(G)} \) using \( \epsilon \)-edge LDP (or \( \epsilon \)-edge centralized DP) algorithms in Section 4. Specifically, we used the following algorithms:

Table 2: Bounds on \( l_2 \) losses for privately estimating \( f_{k,*} \) and \( f_{\Delta} \). For upper-bounds, we assume that \( d_{\text{max}} = d_{\text{max}} \). For the centralized model, we use the Laplace mechanism. For the one-round \( f_{\Delta} \) algorithm, we apply Theorem 4 with constant \( \alpha \). For the two-round protocol \( f_{\Delta} \) algorithm, we apply Theorem 6 with \( \epsilon_1 = \epsilon_2 = \frac{\Delta}{2} \).
Algorithms for triangles. For algorithms for estimating \( f_\Delta(G) \), we used the following three algorithms: (1) the RR (Randomized Response) with the empirical estimation method in the local model (i.e., \( \text{LocalRR}_\Delta \) in Section 4.2), (2) the two-rounds algorithm in the local model (i.e., \( \text{Local2Rounds}_\Delta \) in Section 4.3), and (3) the Laplacian mechanism in the centralized model (i.e., \( \text{CentralLap}_\Delta \) in Section 4.2).

Algorithms for \( k \)-stars. For algorithms for estimating \( f_k(G) \), we used the following two algorithms: (1) the Laplacian mechanism in the local model (i.e., \( \text{LocalLap}_k \) in Section 4.1) and (2) the Laplacian mechanism in the centralized model (i.e., \( \text{CentralLap}_k \) in Section 4.1).

For each algorithm, we evaluated the \( l_2 \) loss and the relative error (as described in Section 3.4), while keeping the values of \( n \) and \( \epsilon \). To stabilize the performance, we attempted \( \gamma \in \mathbb{N} \) ways to randomly select \( n \) users from \( G^* \), and averaged the utility value over all the \( \gamma \) ways to randomly select \( n \) users. When we changed \( n \) from 1000 to 10000, we set \( \gamma = 100 \) because the variance was large. For other cases, we set \( \gamma = 10 \).

5.2 Experimental Results

Relation between \( n \) and the \( l_2 \) loss. We first evaluated the \( l_2 \) loss of the estimates of \( f_2(G) \) (triangle counts), \( f_2(G) \) (2-star counts), and \( f_3(G) \) (3-star counts) while changing the number of users \( n \). Figures 3 and 4 shows the results (\( \epsilon = 1 \)). Here we changed \( n \) from 1000 to 200000 in IMDB, and from 1000 to 1600000 in Orkut. Note that we did not evaluate \( \text{LocalRR}_\Delta \) when \( n \) was larger than 10000, because \( \text{LocalRR}_\Delta \) was inefficient (the time complexity of \( \text{LocalRR}_\Delta \) is \( O(n^3) \), as described in Section 4.3). In \( \text{Local2Rounds}_\Delta \), we set \( \epsilon_1 = \epsilon_2 = \frac{1}{2} \), so that \( \epsilon = \epsilon_1 + \epsilon_2 = 1 \). In \( \text{Local2Rounds}_\Delta \), \( \text{CentralLap}_\Delta \), \( \text{LocalLap}_k \), and \( \text{CentralLap}_k \), we set \( d_{\text{max}} = d_{\text{max}} \) (i.e., we assumed that \( d_{\text{max}} \) is publicly available and did not perform graph projection), because we want to examine how well our theoretical results hold in our experiments. We also evaluate the effectiveness of the private calculation of \( d_{\text{max}} \) at the end of Section 5.2.

Figure 3 shows that \( \text{Local2Rounds}_\Delta \) significantly outperforms \( \text{LocalRR}_\Delta \). Specifically, the \( l_2 \) loss of \( \text{Local2Rounds}_\Delta \) is smaller than that of \( \text{LocalRR}_\Delta \) by a factor of about 10^2. The difference between \( \text{Local2Rounds}_\Delta \) and \( \text{LocalRR}_\Delta \) is larger in Orkut. This is because Orkut is more sparse, as described in Section 5.1. For example, when \( n = 10000 \), the maximum degree \( d_{\text{max}} \) in \( G \) was 73.5 and 27.8 on average in IMDB and Orkut, respectively. Recall that for a fixed \( \epsilon \), the expected \( l_2 \) loss of \( \text{Local2Rounds}_\Delta \) and \( \text{LocalRR}_\Delta \) can be expressed as \( O(n d_{\text{max}}^{\epsilon}) \) and \( O(n^{\epsilon}) \), respectively. Thus \( \text{Local2Rounds}_\Delta \) significantly outperforms \( \text{LocalRR}_\Delta \), especially in sparse graph datasets.

Figures 3 and 4 show that the \( l_2 \) loss is roughly consistent with our upper-bounds in terms of \( n \). Specifically, \( \text{LocalRR}_\Delta \), \( \text{Local2Rounds}_\Delta \), \( \text{CentralLap}_\Delta \), \( \text{LocalLap}_k \), and \( \text{CentralLap}_k \) achieve the expected \( l_2 \) loss of \( O(n^{\epsilon}) \), \( O(n d_{\text{max}}^{\epsilon}) \), \( O(d_{\text{max}}^{\epsilon}) \), \( O(n d_{\text{max}}^{2\epsilon-2}) \), and \( O(d_{\text{max}}^{2\epsilon-2}) \), respectively. Here note that each user’s degree increases roughly in proportion to \( n \) (though the degree is much smaller than \( n \)), as we randomly select \( n \) users from the whole graph \( G^* \). Assuming that \( d_{\text{max}} = O(n) \),
we want to estimate \( \triangle \) when \( n = 10000 \) \((\varepsilon_1 = \varepsilon_2 = \frac{\varepsilon}{2}, d_{max} = d_{max})\).

Figures 3 and 4 are roughly consistent with the upper-bounds. The figures also show the limitations of the local model in terms of the utility when compared to the centralized model.

**Relation between \( \varepsilon \) and the \( l_2 \) loss.** Next we evaluated the \( l_2 \) loss when we changed the privacy budget \( \varepsilon \) in edge LDP. Figure 5 shows the results for triangles and 2-stars \((n = 10000)\). Here we omit the result of 3-stars because it is similar to that of 2-stars. In Local2Rounds, we set \( \varepsilon_1 = \varepsilon_2 = \frac{\varepsilon}{2} \).

Figure 5 shows that the \( l_2 \) loss is roughly consistent with our upper-bounds in terms of \( \varepsilon \). For example, when we decrease \( \varepsilon \) from 0.4 to 0.1, the \( l_2 \) loss increases by a factor of about 5000, 200, 16 for both the datasets in LocalRR\(_{\Delta}\), Local2Rounds\(_{\Delta}\), and CentralLap\(_{\Delta}\), respectively. This is roughly consistent with our theoretical results that for small \( \varepsilon \), the expected \( l_2 \) loss of LocalRR\(_{\Delta}\), Local2Rounds\(_{\Delta}\), and CentralLap\(_{\Delta}\) is \( O(\varepsilon^{-6}) \), \( O(\varepsilon^{-4}) \), and \( O(\varepsilon^{-2}) \), respectively.

Figure 5 also shows that Local2Rounds\(_{\Delta}\) significantly outperforms LocalRR\(_{\Delta}\) especially when \( \varepsilon \) is small, which is also consistent with our theoretical results. Conversely, the difference between LocalRR\(_{\Delta}\) and Local2Rounds\(_{\Delta}\) is small when \( \varepsilon \) is large. This is because when \( \varepsilon \) is large, the RR outputs the true value with high probability. For example, when \( \varepsilon \geq 5 \), the RR outputs the true value with \( \frac{\varepsilon}{2 \varepsilon - 5} > 0.993 \). However, LocalRR\(_{\Delta}\) with such a large value of \( \varepsilon \) does not guarantee strong privacy because it outputs the true value in most cases. Local2Rounds\(_{\Delta}\) significantly outperforms LocalRR\(_{\Delta}\) when we want to estimate \( f_\triangle(G) \) or \( f_k(G) \) with a strong privacy guarantee; e.g., \( \varepsilon \leq 1 \)[32].

---

**Relative error.** As the number of users \( n \) increases, the numbers of triangles \( f_\triangle(G) \) and \( k \)-star counts \( f_k(G) \) increase, and this causes the increase of the \( l_2 \) loss. Therefore, we also evaluated the relative error, as described in Section 3.4.

Figure 6 shows the relation between \( n \) and the relative error. In the local model, we used Local2Rounds\(_{\Delta}\) \((\varepsilon = 1 \text{ or } 2)\) and LocalLap\(_{\star}\) \((\varepsilon = 1 \text{ or } 2)\) for estimating triangle counts \( f_\triangle(G) \) and \( k \)-star counts \( f_k(G) \), respectively \((d_{max} = d_{max})\).
\(d_{\text{max}} = d_{\text{max}}\) and calculated the clustering coefficient in the same way.

Figure 6 shows that for all cases, the relative error decreases with increase in \(n\). This is because both \(f_{\Delta}(G)\) and \(f_{\star}(G)\) significantly increase with increase in \(n\). Specifically, let \(f_{\Delta, i}(G) \in \mathbb{Z}_{\geq 0}\) the number of triangles that involve user \(v_i\), and \(f_{\star, i}(G) \in \mathbb{Z}_{\geq 0}\) be the number of \(k\)-stars of which user \(v_i\) is a center. Then \(f_{\Delta}(G) = \frac{1}{\Delta} \sum_{i=1}^{n} f_{\Delta, i}(G)\) and \(f_{\star}(G) = \sum_{i=1}^{n} f_{\star, i}(G)\). Since both \(f_{\Delta, i}(G)\) and \(f_{\star, i}(G)\) increase with increase in \(n\), both \(f_{\Delta}(G)\) and \(f_{\star}(G)\) increase at least in proportion to \(n\). Thus \(f_{\Delta}(G) \geq \Omega(n^2)\) and \(f_{\star}(G) \geq \Omega(n^2)\). In contrast, \(\text{Local2Rounds}_{\Delta}\), \(\text{LocalLap}_{\star}\), \(\text{CentralLap}_{\Delta}\), and \(\text{CentralLap}_{\star}\) achieve the expected \(l_2\) loss of \(O(n), O(n), O(1),\) and \(O(1)\), respectively (when we ignore \(d_{\text{max}}\) and \(\epsilon\)), all of which are smaller than \(O(n^2)\). Therefore, the relative error decreases with increase in \(n\).

This result demonstrates that we can accurately estimate graph statistics for large \(n\) in the local model. In particular, the relative error is smaller in IMDB, because IMDB is denser and includes a larger number of triangles and \(k\)-stars; i.e., the denominator of the relative error is large. For example, when \(n = 200000\) and \(\epsilon = 1\), the relative error is 0.30, 0.0028, and 0.015 for triangles, 2-stars, and 3-stars, respectively. Note that the clustering coefficient requires \(2\epsilon\), because we need to estimate both \(f_{\Delta}(G)\) and \(f_{\star}(G)\). Yet, we can still accurately calculate the clustering coefficient with a moderate privacy budget; e.g., the relative error of the clustering coefficient is 0.30 when the privacy budget is 2 (i.e., \(\epsilon = 1\)). If \(n\) is larger, then \(\epsilon\) would be smaller at the same value of the relative error.

**Private calculation of \(d_{\text{max}}\).** We have so far assumed that \(d_{\text{max}} = d_{\text{max}}\) (i.e., \(d_{\text{max}}\) is publicly available) in our experiments. However, it is difficult for the users and the data collector to know the exact value of \(d_{\text{max}}\) in advance. Therefore, we finally evaluate the methods to privately calculate \(d_{\text{max}}\) with \(\epsilon_0\)-edge LDP, which are described in Sections 4.1 and 4.3.

Specifically, we used \(\text{Local2Rounds}_{\Delta}\) and \(\text{LocalLap}_{\star}\) for estimating \(f_{\Delta}(G)\) and \(f_{\star}(G)\), respectively, and evaluated the following three methods for setting \(d_{\text{max}}\): (i) set \(d_{\text{max}} = n\); (ii) set \(d_{\text{max}} = d_{\text{max}}\); (iii) set \(d_{\text{max}} = \hat{d}_{\text{max}}\), where \(\hat{d}_{\text{max}}\) is the private estimate of \(d_{\text{max}}\) (i.e., noisy max degree) in Sections 4.1 and 4.3). Note that the first (resp. second) method results in adding the Laplacian noise with the global (resp. local) sensitivity without using graph projection. Therefore, we refer to the first method (i) as the *global sensitivity method*, the second method (ii) as the *local sensitivity method*, and the third method (iii) as the *noisy local sensitivity method*.

We set \(n = 200000\) in IMDB and \(n = 1600000\) in Orkut. Regarding the total privacy budget \(\epsilon\) in edge LDP for estimating \(f_{\Delta}(G)\) or \(f_{\star}(G)\), we set \(\epsilon = 1\) or 2. We used \(\epsilon_0\) for privately calculating \(d_{\text{max}}\) (i.e., \(\epsilon_0 = \frac{\epsilon}{\epsilon_0}\)), and the remaining privacy budget \(\epsilon - \epsilon_0\) as input to \(\text{Local2Rounds}_{\Delta}\) or \(\text{LocalLap}_{\star}\). In \(\text{Local2Rounds}_{\Delta}\), we set \(\epsilon_1 = \epsilon_2\); i.e., we set \((\epsilon_0, \epsilon_1, \epsilon_2) = (0.1, 0.45, 0.45)\) or \((0.2, 0.9, 0.9)\). Then we estimated the clustering coefficient in the same way as Figure 6.

Figure 7 shows the results. Figure 7 shows that the noisy local sensitivity method achieves the relative error close to (sometimes almost the same as) the local sensitivity method, and significantly outperforms the global sensitivity method. This means that we can privately estimate \(d_{\text{max}}\) without a significant loss of utility. Recall that the private calculation of \(d_{\text{max}}\), does not increase the number of rounds in \(\text{Local2Rounds}_{\Delta}\). Therefore, we can privately estimate \(d_{\text{max}}\) and then accurately estimate all the graph statistics (i.e., triangles, \(k\)-stars, and the clustering coefficient) within two rounds.

**Summary of results.** In summary, our experimental results showed that the estimation error of triangle counts is significantly reduced by introducing the interaction between users and a data collector. The results also showed that we can achieve small relative errors for a large number of users \(n\) with privacy budget \(\epsilon = 1\) or 2 in edge LDP. The results also showed that we can privately estimate the maximum degree \(d_{\text{max}}\), which is required in both \(\text{Local2Rounds}_{\Delta}\) and \(\text{LocalLap}_{\star}\), without a significant loss of utility.

### 6 Conclusions

We presented a series of algorithms for counting triangles and \(k\)-stars under LDP. We showed that an additional round can significantly reduce the estimation error in triangles, and the algorithm based on the Laplacian mechanism provides an order optimal error in the non-interactive local model. We
also showed lower-bounds for general functions including triangles and k-stars. We conducted experiments using two real datasets, and showed that our algorithms achieve small relative errors, especially when the number of users is large.

As future work, we would like to develop algorithms for other subgraph counts such as cliques and k-triangles [28].

References

[1] 12th Annual Graph Drawing Contest. http://mozart.diei.unipg.it/gdcontest/contest2005/index.html, 2005.

[2] What to Do When Your Facebook Profile is Maxed Out on Friends. https://authoritypublishing.com/social-media/what-to-do-when-your-facebook-profile-is-maxed-out-on-friends/, 2012.

[3] AI bridging cloud infrastructure (ABCI). https://abci.ai/, 2020.

[4] The diaspora* project. https://diasporafoundation.org/, 2020.

[5] Jayadev Acharya, Clément L. Canonne, Yuhan Liu, Ziteng Sun, and Himanshu Tyagi. Interactive inference under information constraints. CoRR, 2007.10976, 2020.

[6] Jayadev Acharya, Ziteng Sun, and Huanyu Zhang. Hadamard response: Estimating distributions privately, efficiently, and with little communication. In Proceedings of the 32nd Conference on Artificial Intelligence and Statistics (AISTATS’19), pages 1120–1129, 2019.

[7] Raef Bassily, Kobbi Nissim, Uri Stemmer, and Abhradeep Thakurta. Practical locally private heavy hitters. In Proceedings of the 31st Conference on Neural Information Processing Systems (NIPS’17), pages 2285–2293, 2017.

[8] Raef Bassily and Adam Smith. Local, private, efficient protocols for succinct histograms. In Proceedings of the 47th annual ACM Symposium on Theory of Computing (STOC’15), pages 127–135, 2015.

[9] Vincent Bindschaedler and Reza Shokri. Synthesizing plausible privacy-preserving location traces. In Proceedings of the 2016 IEEE Symposium on Security and Privacy (S&P’16), pages 546–563, 2016.

[10] Jeremiah Blocki, Avrim Blum, Anupam Datta, and Or Sheffet. The johnson-lindenstrauss transform itself preserves differential privacy. In Proceedings of the 2012 IEEE 53rd Annual Symposium on Foundations of Computer Science (FOCS’12), pages 410–419, 2012.

[11] Rui Chen, Gergely Acs, and Claude Castelluccia. Differentially private sequential data publication via variable-length n-grams. In Proceedings of the 2012 ACM Conference on Computer and Communications Security (CCS’12), pages 638–649, 2012.

[12] Xihui Chen, Sjouke Mauw, and Yuniar Ramírez-Cruz. Publishing community-preserving attributed social graphs with a differential privacy guarantee. Proceedings on Privacy Enhancing Technologies, (4):131–152, 2020.

[13] Wei-Yen Day, Ninghui Li, and Min Lyu. Publishing graph degree distribution with node differential privacy. In Proceedings of the 2016 ACM SIGMOD International Conference on Management of data (SIGMOD’16), pages 123–138, 2016.

[14] Bolin Ding, Janardhan Kulkarni, and Sergey Yekhanin. Collecting telemetry data privately. In Proceedings of the 31st Conference on Neural Information Processing Systems (NIPS’17), pages 3574–3583, 2017.

[15] John Duchi and Ryan Rogers. Lower Bounds for Locally Private Estimation via Communication Complexity. arXiv:1902.00582 [math, stat], May 2019. arXiv: 1902.00582.

[16] John Duchi, Martin Wainwright, and Michael Jordan. Minimax Optimal Procedures for Locally Private Estimation. arXiv:1604.02390 [cs, math, stat], November 2017. arXiv: 1604.02390.

[17] John C. Duchi, Michael I. Jordan, and Martin J. Wainwright. Local privacy and statistical minimax rates. In Proceedings of the IEEE 54th Annual Symposium on Foundations of Computer Science (FOCS’13), pages 429–438, 2013.

[18] John C. Duchi, Michael I. Jordan, and Martin J. Wainwright. Local privacy, data processing inequalities, and minimax rates. CoRR, 1302.3203, 2014.

[19] Cynthia Dwork. Differential privacy. In Proceedings of the 33rd international conference on Automata, Languages and Programming (ICALP’06), pages 1–12, 2006.

[20] Cynthia Dwork and Aaron Roth. The Algorithmic Foundations of Differential Privacy. Now Publishers, 2014.

[21] Giulia Fanti, Vasyl Pihur, and Ulfar Erlingsson. Building a RAPPOR with the unknown: Privacy-preserving learning of associations and data dictionaries. Proceedings on Privacy Enhancing Technologies (PoPETs), 2016(3):1–21, 2016.
[22] Michael Hay, Chao Li, Gerome Miklau, and David Jensen. Accurate estimation of the degree distribution of private networks. In Proceedings of the 2009 Ninth IEEE International Conference on Data Mining (ICDM’09), pages 169–178, 2009.

[23] Matthew Joseph, Janardhan Kulkarni, Jiemei Mao, and Zhiwei Steven Wu. Locally Private Gaussian Estimation. arXiv:1811.08382 [cs, stat], October 2019. arXiv: 1811.08382.

[24] Matthew Joseph, Jiemei Mao, Seth Neel, and Aaron Roth. The Role of Interactivity in Local Differential Privacy. arXiv:1904.03564 [cs, stat], November 2019. arXiv: 1904.03564.

[25] Matthew Joseph, Jiemei Mao, and Aaron Roth. Exponential separations in local differential privacy. In Proceedings of the Thirty-First Annual ACM-SIAM Symposium on Discrete Algorithms (SODA’20), pages 515–527, 2020.

[26] Peter Kairouz, Keith Bonawitz, and Daniel Ramage. Discrete distribution estimation under local privacy. In Proceedings of the 33rd International Conference on Machine Learning (ICML’16), pages 2436–2444, 2016.

[27] Peter Kairouz, Sewoong Oh, and Pramod Viswanath. Extremal mechanisms for local differential privacy. Journal of Machine Learning Research, 17(1):492–542, 2016.

[28] Vishesh Karwa, Sofya Raskhodnikova, Adam Smith, and Grigory Yaroslavtsev. Private analysis of graph structure. Proceedings of the VLDB Endowment, 4(11):1146–1157, 2011.

[29] Shiva Prasad Kasiviswanathan, Homin K. Lee, Kobbi Nissim, and Sofya Raskhodnikova. What can we learn privately? In Proceedings of the 2008 49th Annual IEEE Symposium on Foundations of Computer Science (FOCS’08), pages 531–540, 2008.

[30] Shiva Prasad Kasiviswanathan, Kobbi Nissim, Sofya Raskhodnikova, and Adam Smith. Analyzing graphs with node differential privacy. In Proceedings of the 10th theory of cryptography conference on Theory of Cryptography (TCC’13), pages 457–476, 2013.

[31] Jure Leskovec and Andrej Krevl. SNAP Datasets: Stanford large network dataset collection. http://snap.stanford.edu/data, 2014.

[32] Ninghui Li, Min Lyu, and Dong Su. Differential Privacy: From Theory to Practice. Morgan & Claypool Publishers, 2016.

[33] Chris Morris. Hackers had a banner year in 2019. https://fortune.com/2020/01/28/2019-data-breach-increases-hackers/, 2020.

[34] Takao Murakami and Yusuke Kawamoto. Utility-optimized local differential privacy mechanisms for distribution estimation. In Proceedings of the 28th USENIX Security Symposium (USENIX Security’19), pages 1877–1894, 2019.

[35] Kevin P. Murphy. Machine Learning: A Probabilistic Perspective. The MIT Press, 2012.

[36] Kobbi Nissim, Sofya Raskhodnikova, and Adam Smith. Smooth sensitivity and sampling in private data analysis. In Proceedings of the 39th Annual ACM Symposium on Theory of Computing (STOC’07), pages 75–84, 2007.

[37] Thomas Paul, Antonino Fumulani, and Thorsten Strufe. A survey on decentralized online social networks. Computer Networks, 75:437–452, 2014.

[38] Zhan Qin, Yin Yang, Ting Yu, Issa Khalil, Xiaokui Xiao, and Kui Ren. Heavy hitter estimation over set-valued data with local differential privacy. In Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security (CCS’16), pages 192–203, 2016.

[39] Zhan Qin, Ting Yu, Yin Yang, Issa Khalil, Xiaokui Xiao, and Kui Ren. Generating synthetic decentralized social graphs with local differential privacy. In Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security (CCS’17), pages 425–438, 2017.

[40] Cyrus Rashtchian, David P. Woodruff, and Hanlin Zhu. Vector-matrix-vector queries for solving linear algebra, statistics, and graph problems. CoRR, 2006.14015, 2020.

[41] Sofya Raskhodnikova and Adam Smith. Efficient lipschitz extensions for high-dimensional graph statistics and node private degree distributions. CoRR, 1504.07912, 2015.

[42] Sofya Raskhodnikova and Adam Smith. Differentially Private Analysis of Graphs, pages 543–547. Springer, 2016.

[43] Andrea De Salve, Paolo Mori, and Laura Ricci. A survey on privacy in decentralized online social networks. Computer Science Review, 27:154–176, 2018.

[44] Tara Seals. Data breaches increase 40% in 2016. https://www.infosecurity-magazine.com/news/data-breaches-increase-40-in-2016/, 2017.

[45] Shuang Song, Susan Little, Sanjay Mehta, Staal Vinterboy, and Kamalika Chaudhuri. Differentially private
continual release of graph statistics. CoRR, 1809.02575, 2018.

[46] Abhradeep Guha Thakurta, Andrew H. Vyrros, Umesh S. Vaishampayan, Gaurav Kapoor, Julien Freudiger, Vivek Rangarajan Sridhar, and Doug Davidson. Learning New Words, US Patent 9,594,741, Mar. 14 2017.

[47] Úlfar Erlingsson, Vasyl Pihur, and Aleksandra Korolova. RAPPOR: Randomized aggregatable privacy-preserving ordinal response. In Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security (CCS’14), pages 1054–1067, 2014.

[48] Tianhao Wang, Jeremiah Blocki, Ninghui Li, and Somesh Jha. Locally differentially private protocols for frequency estimation. In Proceedings of the 26th USENIX Security Symposium (USENIX Security’17), pages 729–745, 2017.

[49] Yue Wang and Xintao Wu. Preserving differential privacy in degree-correlation based graph generation. Transactions on Data Privacy, 6(2), 2013.

[50] Yue Wang, Xintao Wu, and Leting Wu. Differential privacy preserving spectral graph analysis. In Proceedings of the 17th Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD’13), pages 329–340, 2013.

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

[52] Xiaokui Xiao, Gabriel Bender, Michael Hay, and Johannes Gehrke. ireduct: Differential privacy with reduced relative errors. In Proceedings of the 2011 ACM SIGMOD International Conference on Management of data (SIGMOD’11), pages 229–240, 2011.

[53] Min Ye and Alexander Barga. Optimal schemes for discrete distribution estimation under local differential privacy. In Proceedings of the 2017 IEEE International Symposium on Information Theory (ISIT’17), pages 759—763, 2017.

[54] Qingqing Ye, Haibo Hu, Man Ho Au, Xiaofeng Meng, and Xiaokui Xiao. Towards locally differentially private generic graph metric estimation. In Proceedings of the IEEE 36th International Conference on Data Engineering (ICDE’20), pages 1922–1925, 2020.

[55] Hailong Zhang, Suflan Latif, Raef Bassily, and Atanas Rountev. Differentially-private control-flow node coverage for software usage analysis. In Proceedings of the 29th USENIX Security Symposium (USENIX Security’20), pages 1021–1038, 2020.

---

### A Effectiveness of empirical estimation in LocalRRΔ

In Section 4.2, we presented LocalRRΔ, which uses the empirical estimation method after the RR. Here we show the effectiveness of empirical estimation by comparing LocalRRΔ with the RR without empirical estimation. We also note that the RR has been applied to the adjacency matrix without empirical estimation in [39, 54] for different purposes than counting triangles.

As the RR without empirical estimation, we applied the RR to the lower triangular part of the adjacency matrix A; i.e., we ran lines 1 to 6 in Algorithm 2. Then we output the number of noisy triangles m3. We denote this algorithm by RR w/o emp.

Figure 8 shows the l2 loss of LocalRRΔ and RR w/o emp when we changed n from 1000 to 10000 or ε in edge LDP from 0.1 to 2. The experimental set-up is the same as Section 5.1. Figure 8 shows that LocalRRΔ significantly outperforms RR w/o emp, which means that the l2 loss is significantly reduced by empirical estimation. As shown in Section 5, the l2 loss of LocalRRΔ is also significantly reduced by an additional round of interaction.

### B Proof of Statements in Section 4

Here we prove the statements in Section 4. Our proofs will repeatedly use the well-known bias-variance decomposition [35], which we briefly explain below. We denote the variance of the random variable X by Var[X]. If we are producing a private, randomized estimate f(G) of the graph function f(G),
then the expected $I_2$ loss can be written as:
\[
\mathbb{E}[I_2^2(\hat{f}(G), f(G))] = \left(\mathbb{E}[\hat{f}(G)] - f(G)\right)^2 + \mathbb{V}[\hat{f}(G)].
\] (6)

The first term is the bias, and the second term is the variance. If the estimate is unbiased (i.e., $\mathbb{E}[\hat{f}(G)] = f(G)$), then the expected $I_2$ loss is equal to the variance.

B.1 Proof of Theorem 1

Let $R_k$ be LocalLap$_k$. Let $d_l, d'_l \in \mathbb{Z}_{\geq 0}$ be the number of “1”s in two neighbor lists $a_l, a'_l \in \{0, 1\}^n$ that differ in one bit. Let $r_l = \left(d_l \choose 1\right)$ and $r'_l = \left(d'_l \choose 1\right)$. Below we consider two cases about $d_l$: when $d_l < \tilde{d}_{\text{max}}$ and when $d_l \geq \tilde{d}_{\text{max}}$.

**Case 1:** $d_l < \tilde{d}_{\text{max}}$. In this case, both $a_l$ and $a'_l$ do not change after graph projection, as $d'_l \leq d_l + 1 \leq \tilde{d}_{\text{max}}$. Then we obtain:
\[
\Pr[R_k(a_l) = \hat{r}_i] = \exp \left( -\frac{\varepsilon |\hat{r}_i - r_l|}{\Delta} \right)
\]
\[
\Pr[R_k(a'_l) = \hat{r}_i] = \exp \left( -\frac{\varepsilon |\hat{r}_i - r'_l|}{\Delta} \right),
\]
where $\Delta = \left(\tilde{d}_{\text{max}}\right)_{k-1}$. Therefore,
\[
\frac{\Pr[R_k(a_l) = \hat{r}_i]}{\Pr[R_k(a'_l) = \hat{r}_i]} = \exp \left( \frac{\varepsilon |\hat{r}_i - r'_l|}{\Delta} - \frac{\varepsilon |\hat{r}_i - r_l|}{\Delta} \right)
\]
\[
\leq \exp \left( \frac{\varepsilon |r'_l - r_l|}{\Delta} \right),
\] (7)
(by the triangle inequality).

If $d'_l = d_l + 1$, then $|r'_l - r_l|$ in (7) can be written as follows:
\[
|r'_l - r_l| = \left( \frac{d_l + 1}{k} \right) - \left( \frac{d_l}{k} \right) = \left( \frac{d_l}{k} \right) < \left( \frac{\tilde{d}_{\text{max}}}{k-1} \right) = \Delta,
\]
Since we add Lap(\frac{\varepsilon}{\Delta}) to $r_l$, we obtain:
\[
\Pr[R_k(a_l) = \hat{r}_i] \leq e^\varepsilon \Pr[R_k(a'_l) = \hat{r}_i].
\] (8)

If $d'_l = d_l - 1$, then $|r'_l - r_l| = \left( \frac{d_l}{k} \right) - \left( \frac{d_l - 1}{k} \right) = \left( \frac{d_l - 1}{k} \right) < \Delta$ and (8) holds. Therefore, LocalLap$_k$ provides $\varepsilon$-edge LDP.

**Case 2:** $d_l \geq \tilde{d}_{\text{max}}$. Assume that $d'_l = d_l + 1$. In this case, $d'_l > \tilde{d}_{\text{max}}$. Therefore, $d'_l$ becomes $\tilde{d}_{\text{max}}$ after graph projection. In addition, $d_l$ also becomes $\tilde{d}_{\text{max}}$ after graph projection. Therefore, we obtain $d_l = d'_l = \tilde{d}_{\text{max}}$ after graph projection. Thus \[ \Pr[R_k(a_l) = \hat{r}_i] = \Pr[R_k(a'_l) = \hat{r}_i]. \]

Assume that $d'_l = d_l - 1$. If $d_l > \tilde{d}_{\text{max}}$, then $d'_l = \tilde{d}_{\text{max}}$ after graph projection. Thus \[ \Pr[R_k(a_l) = \hat{r}_i] = \Pr[R_k(a'_l) = \hat{r}_i]. \]

If $d_l = \tilde{d}_{\text{max}}$, then (8) holds. Therefore, LocalLap$_k$ provides $\varepsilon$-edge LDP.

B.2 Proof of Theorem 2

Assuming the maximum degree $d_{\text{max}}$ of $G$ is at most $\tilde{d}_{\text{max}}$, the only randomness in the algorithm will be the Laplace noise since graph projection will not occur. Since the Laplacian noise $\text{Lap}(\frac{\varepsilon}{\Delta})$ has mean 0, the estimate $\hat{f}_k(G, \varepsilon, \tilde{d}_{\text{max}})$ is unbiased. Then by the bias-variance decomposition [35], the expected $I_2$ loss $\mathbb{E}[I_2^2(\hat{f}_k(G, \varepsilon, \tilde{d}_{\text{max}}), f(G))]$ is equal to the variance of $\hat{f}_k(G, \varepsilon, \tilde{d}_{\text{max}})$. The variance of $\hat{f}_k(G, \varepsilon, \tilde{d}_{\text{max}})$ can be written as follows:
\[
\mathbb{V}[\hat{f}_k(G, \varepsilon, \tilde{d}_{\text{max}})] = \mathbb{V} \left[ \sum_{i=1}^{n} \text{Lap} \left( \frac{\Delta}{\varepsilon} \right) \right]
\]
\[
= \frac{n\Delta^2}{\varepsilon^2}.
\]
Since $\Delta = \left(\tilde{d}_{\text{max}}\right)_{k-1} = O(\tilde{d}_{\text{max}})$, we obtain:
\[
\mathbb{E}[I_2^2(\hat{f}_k(G, \varepsilon, \tilde{d}_{\text{max}}), f(G))] = \mathbb{V}[\hat{f}_k(G, \varepsilon, \tilde{d}_{\text{max}})]
\]
\[
= O \left( \frac{n\tilde{d}_{\text{max}}^2}{\varepsilon^2} \right).
\]

B.3 Proof of Proposition 2

Let $\mu = \varepsilon^2$ and $Q \in [0, 1]^{4 \times 4}$ be a $4 \times 4$ matrix such that:
\[
Q = \frac{1}{(\mu + 1)^3} \begin{pmatrix}
\mu^2 & 3\mu^2 & 3\mu & 1 \\
\mu^2 & 2\mu^2 + 1 & \mu & 0 \\
\mu^2 & 2\mu^2 + 1 & \mu & 0 \\
1 & 3\mu & 3\mu^2 & \mu^3
\end{pmatrix}.
\] (9)

Let $c_3, c_2, c_1, c_0 \in \mathbb{Z}_{\geq 0}$ be respectively the number of triangles, 2-edges, 1-edge, and no-edges in $G$. Then we obtain:
\[
(\mathbb{E}[m_3], \mathbb{E}[m_2], \mathbb{E}[m_1], \mathbb{E}[m_0]) = (c_3, c_2, c_1, c_0)Q.
\] (10)

In other words, $Q$ is a transition matrix from a type of subgraph (i.e., triangle, 2-edges, 1-edge, or no-edge) in $G$ to a type of subgraph in $G'$. Let $\hat{c}_3, \hat{c}_2, \hat{c}_1, \hat{c}_0 \in \mathbb{R}$ be the empirical estimate of $(c_3, c_2, c_1, c_0)$. By (10), they can be written as follows:
\[
(\hat{c}_3, \hat{c}_2, \hat{c}_1, \hat{c}_0) = (m_3, m_2, m_1, m_0)Q^{-1}.
\] (11)

Let $Q^{-1}$ be the $(i, j)$-th element of $Q^{-1}$. By using Cramer’s rule, we obtain:
\[
Q^{-1}_{1,1} = \frac{\mu^3}{(\mu - 1)^2}, \quad Q^{-1}_{2,1} = -\frac{\mu^2}{(\mu - 1)^2},
\] (12)
\[
Q^{-1}_{3,1} = \frac{\mu}{(\mu - 1)^2}, \quad Q^{-1}_{4,1} = -\frac{1}{(\mu - 1)^2}.
\] (13)

By (11), (12), and (13), we obtain:
\[
\hat{c}_3 = \frac{\mu^3}{(\mu - 1)^2}m_3 - \frac{\mu^2}{(\mu - 1)^2}m_2 + \frac{\mu}{(\mu - 1)^2}m_1 - \frac{1}{(\mu - 1)^2}m_0.
\]

Since the empirical estimate is unbiased [26,48], we obtain (4) in Proposition 2.
B.4  Proof of Theorem 3

Since LocalRRΔ applies the RR to the lower triangular part of the adjacency matrix A, it provides ε-edge LDP for \((R_1, \ldots, R_n)\). Lines 5 to 8 in Algorithm 2 are post-processing of \((R_1, \ldots, R_n)\). Thus, by the immunity to post-processing \([20]\), LocalRRΔ provides ε-edge LDP for the output \(\frac{1}{n-1}(\mu^2 m_3 - \mu^2 m_2 + \mu m_1 - m_0)\).

In addition, the existence of edge \((v_i, v_j) \in E (i > j)\) affects only one element \(a_{i,j}\) in the lower triangular part of A. Therefore, LocalRRΔ provides ε-entire edge LDP.

B.5  Proof of Theorem 4

By Proposition 2, the estimate \(\hat{f}_\Delta(G, \varepsilon)\) by LocalRRΔ is unbiased. Then by the bias-variance decomposition \([35]\), the expected \(I_2\) loss \(E[\hat{f}_\Delta^2(G, \varepsilon), f_\Delta(G)]\) is equal to the variance of \(\hat{f}_\Delta(G, \varepsilon)\). Let \(a_3 = \frac{\mu}{(\mu - 1)^3}, a_2 = -\frac{\mu^2}{(\mu - 1)^2}, a_1 = \frac{\mu}{(\mu - 1)},\) and \(a_0 = \frac{1}{(\mu - 1)}\). Then the variance of \(\hat{f}_\Delta(G, \varepsilon)\) can be written as follows:

\[
\begin{align*}
\mathbb{V}[\hat{f}_\Delta(G, \varepsilon)] &= \mathbb{V}[a_3 m_3 + a_2 m_2 + a_1 m_1 + a_0 m_0] \\
&= a_3^2 \mathbb{V}[m_3] + a_2^2 \mathbb{V}[m_2] + a_1^2 \mathbb{V}[m_1] + a_0^2 \mathbb{V}[m_0] \\
&\quad + \sum_{i=0}^{3} \sum_{j=0, j \neq i}^{3} 2a_i a_j \text{cov}(m_i, m_j),
\end{align*}
\]

(14)

where \(\text{cov}(m_i, m_j)\) represents the covariance of \(m_i\) and \(m_j\). The covariance \(\text{cov}(m_i, m_j)\) can be written as follows:

\[
\begin{align*}
\text{cov}(m_i, m_j) &\leq \sqrt{\mathbb{V}[m_i] \mathbb{V}[m_j]} \\
&\quad \text{(by Cauchy-Schwarz inequality)} \\
&\leq \max\{\mathbb{V}[m_i], \mathbb{V}[m_j]\} \\
&\leq \mathbb{V}[m_i] + \mathbb{V}[m_j].
\end{align*}
\]

(15)

By (14) and (15), we obtain:

\[
\begin{align*}
\mathbb{V}[\hat{f}_\Delta(G, \varepsilon)] &\leq (a_3^2 + 4a_3(a_2 + a_1 + a_0)) \mathbb{V}[m_3] \\
&\quad + (a_2^2 + 4a_2(a_1 + a_1 + a_0)) \mathbb{V}[m_2] \\
&\quad + (a_1^2 + 4a_1(a_3 + a_2 + a_0)) \mathbb{V}[m_1] \\
&\quad + (a_0^2 + 4a_0(a_3 + a_2 + a_1)) \mathbb{V}[m_0] \\
&= O\left(\frac{e^{\rho \varepsilon}}{(e^\varepsilon - 1)^6} (\mathbb{V}[m_3] + \mathbb{V}[m_2] + \mathbb{V}[m_1] + \mathbb{V}[m_0])\right).
\end{align*}
\]

(16)

Below we calculate \(\mathbb{V}[m_3], \mathbb{V}[m_2], \mathbb{V}[m_1],\) and \(\mathbb{V}[m_0]\) by assuming the Erdős-Rényi model \(G(n, \alpha)\) for G:

Lemma 1. Let \(G \sim G(n, \alpha)\). Let \(p = \frac{\alpha}{n-1}\) and \(\beta = \alpha(1-p) + (1-\alpha)p\). Then \(\mathbb{V}[m_3] = O(\beta^3 n^4 + \beta^2 n^3), \mathbb{V}[m_2] = O(\beta^3 n^4 + \beta^2 n^3),\) and \(\mathbb{V}[m_1] = \mathbb{V}[m_0] = O(\beta n^4)\).

Before going into the proof of Lemma 1, we prove Theorem 4 using Lemma 1. By (16) and Lemma 1, we obtain:

\[
\mathbb{V}[\hat{f}_\Delta(G, \varepsilon)] = O\left(\frac{e^{\rho \varepsilon}}{(e^\varepsilon - 1)^6} \beta n^4\right),
\]

which proves Theorem 4. \(\square\)

We now prove Lemma 1:

Proof of Lemma 1. First we show the variance of \(m_3\) and \(m_0\). Then we show the variance of \(m_2\) and \(m_1\).

Variance of \(m_3\) and \(m_0\). Since each edge in the original graph G is independently generated with probability \(\alpha \in [0, 1]\), each edge in the noisy graph \(G'\) is independently generated with probability \(\beta = \alpha(1-p) + (1-\alpha)p \in [0, 1]\), where \(p = \frac{1}{n-1}\). Thus \(m_3\) is the number of triangles in graph \(G' \sim G(n, \beta)\).

For \(i, j, k \in [n], \) let \(y_{i,j,k} \in \{0, 1\}\) be a variable that takes 1 if and only if \((v_i, v_j, v_k)\) forms a triangle. Then \(\mathbb{E}[m_3^2]\) can be written as follows:

\[
\mathbb{E}[m_3^2] = \sum_{i < j < k} \sum_{f < j < k'} \mathbb{E}[y_{i,j,k} y_{f,j,k'}]
\]

(17)

\(\mathbb{E}[y_{i,j,k} y_{f,j,k'}]\) in (17) is the probability that both \((v_i, v_j, v_k)\) and \((v_f, v_j, v_k')\) form a triangle. This event can be divided into the following four types:

1. \((i, j, k) = (i', j', k')\). There are \(\binom{n}{3}\) such terms in (17). For each term, \(\mathbb{E}[y_{i,j,k} y_{f,j,k'}] = \beta^3\).

2. \((i, j, k)\) and \((i', j', k')\) have two elements in common. There are \(\binom{n}{3} \cdot \binom{n-2}{3} = 12 \binom{n}{3}\) such terms in (17). For each term, \(\mathbb{E}[y_{i,j,k} y_{f,j,k'}] = \beta^5\).

3. \((i, j, k)\) and \((i', j', k')\) have one element in common. There are \(n \cdot \binom{n-2}{3} \cdot \binom{n-3}{2} = 30 \binom{n}{3}\) such terms in (17). For each term, \(\mathbb{E}[y_{i,j,k} y_{f,j,k'}] = \beta^6\).

4. \((i, j, k)\) and \((i', j', k')\) have no common elements. There are \(\binom{n}{3} \cdot \binom{n-2}{3} = 20 \binom{n}{6}\) such terms in (17). For each term, \(\mathbb{E}[y_{i,j,k} y_{f,j,k'}] = \beta^6\).

Moreover, \(\mathbb{E}[m_3^2] = \binom{n}{3} \beta^6\). Therefore, the variance of \(m_3\) can be written as follows:

\[
\begin{align*}
\mathbb{V}[m_3] &= \binom{n}{3} \beta^3 + 12 \binom{n}{3} \beta^5 + 30 \binom{n}{3} \beta^3 - \binom{n}{3}^2 \beta^6 \\
&= \binom{n}{3} \beta^3 (1 - \beta^3) + 12 \binom{n}{3} \beta^5 (1 - \beta) \\
&= O(\beta^5 n^4 + \beta^3 n^3).
\end{align*}
\]

By changing \(\beta\) to \(1-\beta\) and counting triangles, we get a random variable with the same distribution as \(m_0\). Thus:

\[
\begin{align*}
\mathbb{V}[m_0] &= \binom{n}{3} (1 - \beta)^3 (1 - (1 - \beta)^3) + 12 \binom{n}{3} (1 - \beta)^5 \\
&= O(\beta n^4).
\end{align*}
\]

Variance of \(m_2\) and \(m_1\). For \(i, j, k \in [n], \) let \(z_{i,j,k} \in \{0, 1\}\) be
a variable that takes 1 if and only if \((v_i, v_j, v_k)\) forms 2-edges (i.e., exactly one edge is missing in the three nodes). Then \(E[m_2^2]\) can be written as follows:

\[
E[m_2^2] = \sum_{i < j < k} \sum_{j < k'} E[z_{i,j,k}z_{i,j,k'}]
\]  

(18)

\(E[z_{i,j,k}z_{i,j,k'}]\) in (18) is the probability that both \((v_i, v_j, v_k)\) and \((v_i, v_j', v_k')\) form 2-edges. This event can be divided into the following four types:

1. \((i, j, k) = (i', j', k')\). There are \(\binom{3}{3}\) such terms in (18). For each term, \(E[z_{i,j,k}z_{i,j,k'}] = 3\beta^2(1 - \beta)\).

2. \((i, j, k)\) and \((i', j', k')\) have two elements in common. There are \(\binom{3}{3}(n-2)(n-3) = 12\binom{3}{3}\) such terms in (18). For example, consider a term in which \(i = i' = 1, j = j' = 2, k = 3,\) and \(k' = 4\). Both \((v_1, v_2, v_3)\) and \((v_1, v_2, v_4)\) form 2-edges if:
   a. \((v_1, v_2), (v_1, v_3), (v_2, v_3) \in E', (v_2, v_4) \not\in E'\),
   b. \((v_1, v_2), (v_1, v_3), (v_2, v_4) \in E', (v_2, v_3) \not\in E'\),
   c. \((v_1, v_2), (v_1, v_3), (v_2, v_4) \not\in E', (v_2, v_3) \in E'\),
   d. \((v_1, v_2), (v_1, v_3), (v_2, v_4) \not\in E', (v_2, v_3) \not\in E'\),
   or
   e. \((v_1, v_3), (v_1, v_4), (v_2, v_3), (v_2, v_4) \not\in E', (v_2, v_3) \in E'\).

Thus, \(E[z_{i,j,k}z_{i,j,k'}] = 4\beta^2(1 - \beta)^2 + 3\beta^4(1 - \beta)\) for this term. Similarly, \(E[z_{i,j,k}z_{i,j,k'}] = 3\beta^2(1 - \beta)^2 + 9\beta^4(1 - \beta)\) for the other terms.

3. \((i, j, k)\) and \((i', j', k')\) have one element in common. There are \(n(n-1)(n-2)/2 = 30\binom{3}{3}\) such terms in (18). For each term, \(E[z_{i,j,k}z_{i,j,k'}] = 3\beta^2(1 - \beta)^2 = 9\beta^4(1 - \beta)^2\).

4. \((i, j, k)\) and \((i', j', k')\) have no common elements. There are \(\binom{3}{3}\binom{3}{3} = 20\binom{3}{3}\) such terms in (18). For each term, \(E[z_{i,j,k}z_{i,j,k'}] = 3\beta^2(1 - \beta)^2 = 9\beta^4(1 - \beta)^2\).

Moreover, \(E[m_2^2] = 3\binom{3}{3}\beta^2(1 - \beta)^2 = 9\binom{3}{3}\beta^4(1 - \beta)^2\). Therefore, the variance of \(m_2\) can be written as follows:

\[
\mathbb{V}[m_2] = 3\binom{3}{3}\beta^2(1 - \beta)^2 + 12\binom{3}{3}\beta^4(1 - \beta)^2 \times 12\binom{3}{3}\beta^4(1 - \beta)^2 + 12\binom{3}{3}\beta^4(1 - \beta)^2 - 9\binom{3}{3}\beta^4(1 - \beta)^2.
\]

By simple calculations,

\[
270\binom{3}{3} + 180\binom{3}{3} - 9\binom{3}{3} = -108\binom{3}{3} - 9\binom{3}{3}.
\]

Thus we obtain:

\[
\mathbb{V}[m_2] = 3\binom{3}{3}\beta^2(1 - \beta)(1 - 3\beta^2(1 - \beta))
\]

Similarly, the variance of \(m_1\) can be written as follows:

\[
\mathbb{V}[m_1] = 3\binom{3}{3}\beta(1 - \beta)^2(1 - 3\beta)(1 - \beta)^2 + 12\binom{3}{3}\beta(1 - \beta)^3(4\beta + (1 - \beta) - 9\beta(1 - \beta)) = O(\beta n^2).
\]

\(\square\)

### B.6 Proof of Proposition 3

Let \(t_s = \sum_{i=1}^{n} t_i\) and \(s_s = \sum_{i=1}^{n} s_i\). Let \(s_s^\triangle\) be the number of triplets \((v_i, v_j, v_k)\) such that \(j < k < i, a_{i,j} = a_{i,k} = 1,\) and \(a_{j,k} = 0\). Note that \(s_s^\triangle = s_s^\triangle + s_s^\triangle\) and \(s_s^\triangle = f_{\Delta}(G)\).

Consider a triangle \((v_i, v_j, v_k)\) in \(G\). This triangle is counted \(1 - p_1(\frac{\sum_{i=1}^{n} t_i}{n})\) times in expectation in \(t_s\). Consider 2-edges \((v_i, v_j, v_k)\) in \(G\) (i.e., exactly one edge is missing in the three nodes). This is counted \(p_1(\frac{\sum_{i=1}^{n} t_i}{n})\) times in expectation in \(t_s\). No other events can change \(t_s\). Therefore, we obtain:

\[
E[t_s] = (1 - p_1)s_s^\triangle + p_1 s_s^\triangle.
\]

By \(s_s = s_s^\triangle + s_s^\triangle\) and \(s_s^\triangle = f_{\Delta}(G)\), we obtain:

\[
E\left[\sum_{i=1}^{n} w_i\right] = E\left[\sum_{i=1}^{n} (t_i - p_1 s_i)\right]
\]

\[
= E[t_s - p_1 s_s]
\]

\[
= E[t_s] - p_1 E[s_s^\triangle + s_s^\triangle]
\]

\[
= (1 - p_1)s_s^\triangle + p_1 s_s^\triangle - p_1(s_s^\triangle + s_s^\triangle)
\]

\[
= (1 - 2p_1)f_{\Delta}(G),
\]

hence

\[
E\left[\frac{1}{1 - 2p_1} \sum_{i=1}^{n} w_i\right] = f_{\Delta}(G).
\]

\(\square\)

### B.7 Proof of Theorem 5

Let \(\mathcal{B}_2\) be Local2Rounds\(_{\Delta}\). Consider two neighbor lists \(a_i, a'_i \in \{0, 1\}^n\) that differ in one bit. Let \(d_i\) (resp. \(d_i'\)) \(\in \mathbb{Z}_{\geq 0}\) be the number of “1”s in \(a_i\) (resp. \(a'_i\)). Let \(\bar{a}_i\) (resp. \(\bar{a}'_i\)) \(\in \{0, 1\}^n\) be neighbor lists obtained by setting all of the \(i\)-th to the \(n\)-th elements in \(a_i\) (resp. \(a'_i\)) to 0. Let \(d_i\) (resp. \(d_i'\)) \(\in \mathbb{Z}_{\geq 0}\) be the number of “1”s in \(\bar{a}_i\) (resp. \(\bar{a}'_i\)). For example, if \(n = 6, a_4 = (1, 0, 1, 0, 1, 1),\) and \(a'_4 = (1, 1, 1, 0, 1, 1),\) then \(d_4 = 4,\)

\[
d'_4 = 5, \bar{a}_4 = (1, 0, 1, 0, 0, 0), \bar{a}'_4 = (1, 1, 1, 0, 0, 0), d_4 = 2,\) and \(d_4' = 3.\)

Furthermore, let \(t_i\) (resp. \(t'_i\)) \(\in \mathbb{Z}_{\geq 0}\) be the number of triplets \((v_i, v_j, v_k)\) such that \(j < i < k, (v_i, v_j) \in E, (v_i, v_k) \in E,\) and \((v_j, v_k) \in E'\) in \(a_i\) (resp. \(a'_i\)). Let \(s_i\) (resp. \(s'_i\)) \(\in \mathbb{Z}_{\geq 0}\) be the
number of triplets \((v_i, v_j, v_k)\) such that \(j < k < i\), \((v_i, v_j) \in E\), and \((v_i, v_k) \in E\) in \(a_i\) (resp. \(a'_i\)). Let \(w_i = t_i - p_i s_i\) and \(w'_i = t'_i - p_i s'_i\). Below we consider two cases about \(d_i\); when \(d_i < \bar{d}_{\max}\) and when \(d_i \geq \bar{d}_{\max}\).

**Case 1:** \(d_i < \bar{d}_{\max}\). Assume that \(d'_i = d_i + 1\). In this case, we have either \(\tilde{a}_i = a_i\) or \(d'_i = \bar{d}_i + 1\). If \(\tilde{a}_i = a_i\), then \(s_i = \bar{s}_i\), \(t_i = t'_i\), and \(w_i = w'_i\); hence \(\Pr[\mathcal{R}(\tilde{a}_i) = \tilde{w}_i] = \Pr[\mathcal{R}(a'_i) = \tilde{w}_i] = \Pr[\mathcal{R}(a'_i) = w'_i]\).

If \(d'_i = \bar{d}_i + 1\), then \(s_i = \bar{s}_i\) and \(\tilde{a}_i = \tilde{a}_i\). Therefore, \(s_i - s_i' = (\bar{d}_i + 1) / 2 - (\bar{d}_i / 2) = \bar{d}_i\).

In addition, since we consider an additional constraint “\((v_j, v_k) \in E'\)” in counting \(t_i\) and \(t'_i\), we have \(t'_i - t_i \leq s'_i - s_i\). Therefore,

\[
|w'_i - w_i| = |t'_i - t_i - p_1(s'_i - s_i)| \leq (1 - p_1)\bar{d}_i.
\]

This results from \(d_i < \bar{d}_{\max}\). If \(p_1 > 0\) and \(d_i < \bar{d}_{\max}\), then \(|w'_i - w_i| < \bar{d}_{\max}\).

Since we add \(\text{Lap}(\gamma_{\max}/\varepsilon_2)\) to \(w_i\), we obtain:

\[
\Pr[\mathcal{R}(\tilde{a}_i) = \tilde{w}_i] \leq e^{\varepsilon_2} \Pr[\mathcal{R}(a'_i) = \tilde{w}_i]. 
\quad (19)
\]

**Case 2:** \(d_i \geq \bar{d}_{\max}\). Assume that \(d'_i = d_i + 1\). In this case, we have \(d_i = \bar{d}_i + 1\) and \(d'_i = \bar{d}_i + 1\). If \(\tilde{a}_i = a_i\), then \(\Pr[\mathcal{R}(\tilde{a}_i) = \tilde{w}_i] = \Pr[\mathcal{R}(a'_i) = \tilde{w}_i] = \Pr[\mathcal{R}(a'_i) = w'_i]\).

If \(\tilde{a}_i = a_i\), then \(s_i = \bar{s}_i\), \(d_i = \bar{d}_i\), and \(w_i = w'_i\); hence \(\Pr[\mathcal{R}(\tilde{a}_i) = \tilde{w}_i] = \Pr[\mathcal{R}(a'_i) = \tilde{w}_i] = \Pr[\mathcal{R}(a'_i) = w'_i]\).

Because \(\tilde{a}_i = a_i\), \(\tilde{a}_i = a'_i\), and \(\tilde{a}_i = a'_i\), the expected \(\Delta\)-edge LDP at the second round. Since \(\tilde{a}_i = a_i\), \(\tilde{a}_i = a'_i\), and \(\tilde{a}_i = a'_i\), the expected \(\Delta\)-edge LDP in total by the composition theorem [20].

**B.8 Proof of Theorem 6**

When the maximum degree \(d_{\max}\) of \(G\) is at most \(\bar{d}_{\max}\), no graph projection will occur. By Proposition 3, the estimate \(f_\Delta(G, \varepsilon)\) by Local2Rounds\(_\Delta\) is unbiased.

By bias-variance decomposition (6), the expected \(I_2(\hat{f}_\Delta(G, \varepsilon), f_\Delta(G))\) is equal to \(\mathbb{V}[\hat{f}_\Delta(G, \varepsilon)]\). Recall that \(p_1 = \frac{1}{(1-\epsilon_1)^2}\). \(\mathbb{V}[\hat{f}_\Delta(G, \varepsilon)]\) can be written as follows:

\[
\mathbb{V}[\hat{f}_\Delta(G, \varepsilon)] = \frac{1}{(1-2p_1)^2} \mathbb{V}\left[\sum_{i=1}^{n} \tilde{w}_i\right] = \frac{1}{(1-2p_1)^2} \mathbb{V}\left[\sum_{i=1}^{n} \tilde{t}_i - p_1 s_i + \text{Lap}(\gamma_{\max}(1-p_1)/\varepsilon_2)\right] = \frac{1}{(1-2p_1)^2} \mathbb{V}\left[\sum_{i=1}^{n} \tilde{t}_i - p_1 s_i + \text{Lap}(\gamma_{\max}(1-p_1)/\varepsilon_2)\right] = \frac{1}{(1-2p_1)^2} \mathbb{V}\left[\sum_{i=1}^{n} \tilde{t}_i - p_1 s_i + \text{Lap}(\gamma_{\max}(1-p_1)/\varepsilon_2)\right] = \frac{1}{(1-2p_1)^2} \mathbb{V}\left[\sum_{i=1}^{n} \tilde{t}_i + \frac{n}{(1-2p_1)^2} \gamma_{\max}(1-p_1)^2/\varepsilon_2^2\right]. 
\quad (21)
\]

In the last line, we are able to get rid of the \(s_i\)’s because they are deterministic. We are also able to sum the variances of the Lap random variables since they are independent; we are not able to do the same with the sum of the \(t_i\)’s.

Recall the definition of \(E'\) computed by the first round of Local2Rounds\(_\Delta\)—the noisy edges released by randomized response. Now,

\[
t_i = \sum_{a_{ij}=a_{jk}=1,j<k<i} 1((v_j, v_k) \in E').
\]

20
This gives

\[
\sum_{i=1}^{n} t_i = \sum_{i=1}^{n} \sum_{a_{i,j} = a_{i,k} = 1 \text{ s.t. } v_j \neq v_k} 1((v_j, v_k) \in E')
\]
\[
= \sum_{1 \leq j < k \leq n} \sum_{a_{i,j} = a_{i,k} = 1} 1((v_j, v_k) \in E')
\]
\[
= \sum_{1 \leq j < k \leq n} \sum_{1 \leq i \leq n} \{ 1 \leq i \leq \max \{ a_{i,j}, a_{i,k} \} \} 1((v_j, v_k) \in E')
\]

Let \( c_{jk} = \{ 1 \leq i \leq \max \{ a_{i,j}, a_{i,k} \} \} \). Notice that \( 1((v_j, v_k) \in E') \) are independent events. Thus, the variance of the above expression is

\[
\mathbb{E} \left[ \sum_{i=1}^{n} t_i \right] = \mathbb{E} \left[ \sum_{1 \leq j < k \leq n} c_{jk} 1((v_j, v_k) \in E') \right]
\]
\[
= \sum_{1 \leq j < k \leq n} c_{jk}^2 \mathbb{E} \left[ 1((v_j, v_k) \in E') \right]
\]
\[
= p_1(1 - p_1) \sum_{1 \leq j < k \leq n} c_{jk}^2.
\]

(22)

c_{jk} is the number of ordered 2-paths from \( j \) to \( k \) in \( G \). Because the degree of user \( v_j \) is at most \( d_{\text{max}} \), \( 0 \leq c_{jk} \leq d_{\text{max}} \). There are at most \( nd_{\text{max}}^2 \) ordered 2-paths in \( G \), since there are only \( d_{\text{max}} \) vertices to go to once a first is picked. Thus, \( \sum_{1 \leq j < k \leq n} c_{jk} \leq nd_{\text{max}}^2 \). Using a Jensen’s inequality style argument, the best way to maximize (22) is to have all \( c_{jk} \) be 0 or \( d_{\text{max}} \). At most \( nd_{\text{max}} \) of the \( c_{ij} \) can be \( d_{\text{max}} \), and the rest are zero. Thus,

\[
\mathbb{E} \left[ \sum_{i=1}^{n} t_i \right] = p_1(1 - p_1) \sum_{1 \leq j < k \leq n} c_{ij}^2
\]
\[
\leq p_1(1 - p_1) nd_{\text{max}} \cdot d_{\text{max}}^2.
\]

Plugging this into (21)

\[
\mathbb{E}[\hat{f}_\triangle(G, \varepsilon)] \leq p_1(1 - p_1) nd_{\text{max}}^3 + 2 nd_{\text{max}}^2 (1 - p_1)^2 (1 - 2p_1)^2 \varepsilon \frac{ \varepsilon^1}{2}
\]
\[
\leq O \left( \frac{p_1 nd_{\text{max}}^3 + nd_{\text{max}}^2}{(1 - 2p_1)^2} \right)
\]
\[
\leq O \left( \frac{\varepsilon^1}{(1 - \frac{\varepsilon^1}{2})^2} \right) \left( nd_{\text{max}}^3 + \frac{\varepsilon^1}{\varepsilon} nd_{\text{max}}^2 \right).
\]

We assume there is a reflexive, symmetric relation \( \sim \) on \( X \) which we use as the neighborhood condition for \( \varepsilon \) differential privacy. We extend \( \sim \) to the space \( X^n \) by saying if \( A, A' \in X^m \), then \( A \sim A' \) if they are equal on all but one coordinate \( i \), and if the \( i \)-th coordinate of \( A' \) is \( a_i' \), then \( a_i \sim a_i' \).

We also assume there is a partial order \( \preceq \) on \( X \), and we extend this to \( X^n \) by saying \( A \preceq A' \) if \( a_i \preceq a_i' \) for all \( 1 \leq i \leq n \). For \( a, a' \in X \) or \( X^m \), we say \( a \preceq a' \) if \( a \preceq a' \) and \( a \preceq a' \).

**Definition 7** (General \((n, D)\) monotone cube for \( f \)). Let \( \{ (a_1, a_1'), \ldots, (a_n, a_n') \} \) be \( n \) pairs of vectors in \( X \) such that for \( i \in [n] \), \( a_i \sim a_i' \). Let \( A = \prod_{j=1}^{n} (a_j, a_j') \) where the product is the Cartesian product. Notice \( A \) has size \( 2^n \), and the relation \( \sim \) defines a directed cube with a source and a sink. Let \( n, D \in \mathbb{N} \). We say \( A \) forms a general \((n, D)\) monotone cube for \( f \) if for all \( A, A' \in A \) such that \( A \sim A' \), we have \( f(A') - f(A) \geq D \).

The following theorem for general \( f \) will help us establish Theorem 7. We prove this theorem in Section B.10.

**Theorem 8.** Let \( f : X^m \rightarrow \mathbb{R} \) be defined by \( f(A) = \hat{f}(R_1(a_1), \ldots, R_n(a_n)) \) where each \( R_k \) satisfies \( \varepsilon \) DP using relationship \( \sim \) and the \( R_k \) are mutually independent. Let \( A \) be a general \((n, D)\) monotone cube for \( f \). Let \( U(A) \) be the uniform distribution over \( A \). Then, \( \mathbb{E}_{A \sim U(A)} [\| f(A) \|] = \Omega(\min \{ 1, \frac{\varepsilon}{(\varepsilon-1)^2} \} nD^2) \).

To prove Theorem 7, recall in the one-round edge local DP model, we have

\[
\hat{f}(A) = \hat{f}(R_1(a_1), \ldots, R_n(a_n))
\]

where each \( R_k \) is an arbitrary function on domain \( \{0, 1\}^n \) satisfying \( \varepsilon \) differential privacy. Let \( A \) be the \((n, D)\) monotone cube for \( f \) given in the statement of Theorem 7. According to Definition 6, let

\[
M = \{ (v_{i_1}, v_{i_2}), \ldots, (v_{i_{2k-1}}, v_{i_{2k}}) \}
\]

be the perfect matching and \( G \) be the graph that defines \( A \). The idea of this proof will be to pair up users \( v_{i_{2j-1}}, v_{i_{2j}} \) to form a general \((n, D)\) monotone cube on this new space, and then apply Theorem 8. For \( 1 \leq j \leq k \), define

\[
b_j = \{ a_{i_{2j-1}}, a_{i_{2j}} \}
\]

Let \( B = \{ b_1, \ldots, b_k \} \). Each \( b_j \) lies in the space \( X = (\{0, 1\}^n)^2 \). We define \( \sim \) and \( \preceq \) on \( X \). Say \( (a_1, a_1'), (a_2, a_2') \in X^2 \), say \( (a_1, a_1') \sim (a_2, a_2') \) if \( a_1 \) (resp. \( a_1' \)) is equal to \( a_2 \) (resp. \( a_2' \)) after changing one coordinate. Let \( \subseteq \) be the subset operation treating elements of \( X \) as indicator vectors.

Now suppose \( B = G \). For \( 1 \leq j \leq k \), let \( i_{2j-1} = s \) and \( i_{2j} = t \). For each \( b_j = \{ a_s, a_t \} \), we have that \( a_2 = a_0 = 0 \) because \( (s, t) \) is in \( M \) and \( M, A \) are disjoint by definition. Let \( a'_{i_s} \) be \( a_i \) with \( a'_{i_t} = 1 \) and \( a'_{i_t} = a_i \) with \( a'_{i_s} = 1 \). Finally, let \( b'_j = \{ a'_{i_s}, a'_{i_t} \} \); i.e., we’ve added edge \( (s, t) \) to users \( s \) and \( t \). Notice that

\[
A = \bigcup_{j=1}^{k} \{ b_j, b'_j \}
\]
\( \mathcal{A} \) is an \((n, D)\) monotone cube for the function \( g : \mathcal{X}^k \to \mathbb{R} \) defined by \( g(B) = f(A) \). Now, define
\[
S_j(b_j) = (\mathcal{R}_{i_1}, \ldots, (a_{i_{j-1}})\mathcal{R}_{a_{i_j}})
\]
By composition of differential privacy, each \( S_j \) satisfies 2\( \varepsilon \)-DP.
Define the estimator \( \hat{g} \) by
\[
\hat{g}(B) = \hat{f}(S_1(b_1), \ldots, S_k(b_k)) = \hat{f}(A)
\]
We can apply Theorem 8 on \( g, \hat{g}, \mathcal{A} \), and the collection \( S_1, \ldots, S_k \) to conclude that
\[
\mathbb{E}_{B \sim U(\mathcal{A})}[\hat{f}(B)] = \mathbb{E}[\hat{f}(A)] = \mathbb{E}[\hat{f}(B)]
\]
We are done noticing that \( g(B) = f(B) \) and \( \hat{g}(B) = \hat{f}(B) \).

**B.10 Proof of Theorem 8**

Use the same setup as Section B.9. Using the bias-variance decomposition,
\[
\mathbb{E}_f[(f(A) - \hat{f}(A))^2] = (f(A) - \mathbb{E}[\hat{f}(A)])^2 + \mathbb{V}[\hat{f}(A)]
\]
Using the law of total variance, \( \mathbb{V}[\hat{f}(A)] \) can be further decomposed into
\[
\mathbb{V}[\hat{f}(A)] = \mathbb{E}_{\mathcal{R}_1, \ldots, \mathcal{R}_k}[\mathbb{V}_f[\hat{f}(\mathcal{R}_1(a_1), \ldots, \mathcal{R}_k(a_n))]] + \mathbb{V}_{\mathcal{R}_1, \ldots, \mathcal{R}_k}[\mathbb{E}_f[\hat{f}(\mathcal{R}_1(a_1), \ldots, \mathcal{R}_k(a_n))]]
\]
The first term of the sum on the RHS vanishes if \( \hat{f} \) is deterministic. Thus, if \( \hat{f} \) is not deterministic, \( f(\mathcal{A}) \) will be outperformed by the estimator \( \hat{g}(A) = \mathbb{E}_f[\hat{f}(\mathcal{R}_1(a_1), \ldots, \mathcal{R}_k(a_n))] \) because such an estimator would satisfy \( \mathbb{E}[\hat{g}(A)] = \mathbb{E}[\hat{f}(A)] \) and \( \mathbb{V}[\hat{f}(A)] \geq \mathbb{V}[\hat{g}(A)] \). Hence, from now on we assume \( \hat{f} \) is deterministic.
Thus,
\[
\mathbb{E}_{A \sim U(\mathcal{A})}[\hat{f}(A)] = \mathbb{E}_{A \sim U(\mathcal{A})}[f(A) - \mathbb{E}[\hat{f}(A)]]^2 + \mathbb{E}_{A \sim U(\mathcal{A})}[\mathbb{V}[\hat{f}(A)]] (23)
\]
We start by changing the \( \mathbb{V}[\hat{f}(A)] \) term to something more useful. Define
\[
I(f) = \mathbb{E}_{A \sim U(\mathcal{A})} \sum_{A' \sim A} (f(A) - f(A'))^2
\]
For a fixed \( A \), the set \( \{A' \sim A \} \) satisfies the conditions of Lemma 2. Therefore, for all \( A \in \mathcal{A} \),
\[
\mathbb{V}[\hat{f}(A)] \geq \frac{e^\varepsilon}{(e^\varepsilon - 1)^2} \sum_{A' \sim A} (\mathbb{E}[\hat{f}(A)] - \mathbb{E}[\hat{f}(A')])^2 (24)
\]
Taking the expectation over all \( A \sim U(\mathcal{A}) \), (24) reads
\[
\mathbb{E}_{A \sim U(\mathcal{A})}[\mathbb{V}[\hat{f}(A)]] \geq \frac{e^\varepsilon}{(e^\varepsilon - 1)^2} \mathbb{I}(\hat{f}(A)) (25)
\]
Define the following quantities
\[
g(A) = \mathbb{E}[\hat{f}(A)]
\]
\[
\|f - g\|_2^2 = \mathbb{E}_{A \sim U(\mathcal{A})}[(f(A) - g(A))^2] (26)
\]
Plugging (25), (26), and (27) into (23),
\[
\mathbb{E}_{A \sim U(\mathcal{A})}[\mathbb{E}[\hat{f}(A)|g(A)]] \geq \|f - g\|_2^2 + \frac{e^\varepsilon}{(e^\varepsilon - 1)^2} \mathbb{I}(g) (28)
\]
Let \( \mathcal{A}_i \) be the elements of \( \mathcal{A} \) with exactly \( i \) coordinates coming from the set \( \{a_1, \ldots, a_n\} \) and \( n - i \) coordinates coming from the set \( \{a_{n+1}, \ldots, a_n\} \) We have \( |\mathcal{A}_i| = \binom{n}{i} \). For \( 0 \leq i \leq n \), define the following averages:
\[
F_i = \frac{1}{\binom{n}{i}} \sum_{A \in \mathcal{A}_i} f(A) \quad G_i = \frac{1}{\binom{n}{i}} \sum_{A \in \mathcal{A}_i} g(A)
\]
We will express \( \|f - g\|_2^2 \) and \( \mathbb{I}(g) \) in terms of the averages \( F_i \) and \( G_i \). Then we can write
\[
\mathbb{I}(g) = \frac{1}{2n} \sum_{A \in \mathcal{A}} \sum_{A' \sim A} (g(A) - g(A'))^2
\]
\[
= \frac{1}{2n} \sum_{A \in \mathcal{A}} \sum_{A' \sim A} (g(A) - g(A'))^2
\]
\[
+ \frac{1}{2n} \sum_{A \in \mathcal{A}} \sum_{A' \sim A} (g(A) - g(A'))^2
\]
\[
= \frac{1}{2^{n-1}} \sum_{A \in \mathcal{A}} \sum_{A' \sim A} (g(A) - g(A'))^2
\]
\[
= \frac{1}{2^{n-1}} \sum_{i=0}^{n-1} \sum_{A \in \mathcal{A}_i} \sum_{A' \sim A} (g(A) - g(A'))^2 (29)
\]
There are \((n - i) \binom{n}{i}\) elements in the set \( \{A, A' : A \in \mathcal{A}, A' \in \mathcal{A}_{i+1}, A \sim A'\} \). Denote \((n - i) \binom{n}{i}\) by \( C \). The quantity
\[
\frac{1}{C} \sum_{A \in \mathcal{A}_i} \sum_{A' \sim A} (g(A) - g(A'))^2 (30)
\]
is an average. By Jensen’s inequality, using the convexity of \( f(x) = x^2 \),
\[
\frac{1}{C} \sum_{A \in \mathcal{A}_i} \sum_{A' \sim A} (g(A) - g(A'))^2 \geq \left( \frac{1}{C} \sum_{A \in \mathcal{A}_i} \sum_{A' \sim A} (g(A') - g(A))^2 \right) (31)
\]
\[
= \left( \frac{1}{C} \sum_{A \in \mathcal{A}_i} \sum_{A' \sim A} g(A') - \sum_{A' \sim A} g(A)) \right)^2 (32)
\]
For any $A' \in \mathcal{A}_{i+1}$, there are $(i+1)$ matrices $A$ in $\mathcal{A}_i$ such that $A \rightsquigarrow A'$. For any $A \in \mathcal{A}_i$, there are $(n-i)$ matrices $A'$ in $\mathcal{A}_{i+1}$ such that $A \rightsquigarrow A'$. Thus, (32) becomes

\[
= \left( \frac{1}{C} \sum_{A' \in \mathcal{A}_{i+1}} (i+1)g(A') - \frac{1}{C} \sum_{A \in \mathcal{A}_i} (n-i)g(A) \right)^2
\]

\[
= \left( \frac{1}{C} (i+1) \binom{n}{i+1} G_{i+1} - \frac{1}{C} (n-i) \binom{n}{i} G_{i} \right)^2
\]

\[
= (G_{i+1} - G_{i})^2
\]

Hence (30) is at least $(G_{i+1} - G_{i})^2$. Plugging this into (29),

\[
I(g) \geq \frac{1}{2^{n-1}} \sum_{i=0}^{n-1} (n-i) \binom{n}{i} (G_{i+1} - G_{i})^2
\]

(33)

Next,

\[
\|f - g\|_2^2 = \frac{1}{2^n} \sum_{A \in \mathcal{A}} (f(A) - g(A))^2
\]

\[
= \frac{1}{2^n} \sum_{i=0}^{n} \sum_{A \in \mathcal{A}_i} (f(A) - g(A))^2
\]

Once again by Jensen’s inequality,

\[
\frac{1}{\binom{n}{i}} \sum_{A \in \mathcal{A}_i} (f(A) - g(A))^2 \geq \left( \frac{1}{\binom{n}{i}} \sum_{A \in \mathcal{A}_i} f(A) - g(A) \right)^2
\]

\[
= (F_i - G_i)^2
\]

(34)

Finally,

\[
\|f - g\|_2^2 \geq \frac{1}{2^n} \sum_{i=0}^{n} \binom{n}{i} (F_i - G_i)^2
\]

(35)

Applying this to (33) and (34),

\[
I(g) \geq \frac{1}{2^{n-1}} \sum_{i=A}^{B-1} (n - (\frac{n}{2} + \sqrt{i} - 1)) K \frac{2^n}{\sqrt{n}} (G_{i+1} - G_{i})^2
\]

\[
\geq 2(\frac{\sqrt{n}}{2} - 2) K \frac{B-1}{i=A} (G_{i+1} - G_{i})^2
\]

\[
\geq 0.5K \sqrt{n} \sum_{i=A}^{B-1} (G_{i+1} - G_{i})^2
\]

\[
\geq Kn \frac{1}{B - A} \sum_{i=A}^{B-1} (G_{i+1} - G_{i})^2
\]

(36)

If $I(g) \geq \frac{n^2 K}{4}$, then by (28), $E_{A \sim R}(\|f(\hat{\mathcal{A}}), f(A)\|)$ would be $\Omega\left(\frac{e^2}{(n-1)^2} n^{D^2}\right)$ and we are done. Otherwise, suppose $I(g) \leq \frac{n^2 K}{4}$, and by (35),

\[
\frac{1}{B - A + 1} \sum_{i=A}^{B-1} (F_i - G_i)^2 \leq D^2
\]

(37)

Since $F_{i+1} - F_i \geq D$, using Lemma 6, we can conclude

\[
\frac{1}{B - A + 1} \sum_{i=A}^{B-1} (F_i - G_i)^2 \geq \Omega((B - A + 1) D^2) = \Omega(nD^2)
\]

By (36), this means $\|f - g\|_2^2 = \Omega(nD^2)$, and by (28), $E_{A \sim R}(\|f(\hat{\mathcal{A}}), f(A)\|)$ is still $\Omega(nD^2)$.

**Lemma 2.** Suppose $\hat{f} : X^n \rightarrow \mathbb{R}$ is a function providing $\epsilon$-local edge DP in the one-round model. Let $A, A_1, \ldots, A_n \sim X^n$ be such that $A \sim A_i$, and $A, A_i$ differ in exactly the $i$th coordinate. Then,

\[
\sum_{i=1}^{n} (E[\hat{f}(A)] - E[\hat{f}(A_i)])^2 \leq \frac{(e^2 - 1)^2}{e^2} V[\hat{f}(A)]
\]

Proof. Recall that in the one-round model

\[
\hat{f}(A) = \hat{f}(R(A_1), \ldots, R(A_n))
\]

Define $\tilde{f}(v) : \{0, 1\}^n \rightarrow \mathbb{R}$ as

\[
\tilde{f}(v) = E[\hat{f}(R(A_1), \ldots, R_i(A_i-1), v, R_{i+1}(A_{i+1}), \ldots, R_n(A_n))]
\]

Notice that $E[\tilde{f}(R(A_i))] = E[\hat{f}(A)]$. Furthermore, if we let $A_i'$ be the $i$th row of $A$, we have $E[\tilde{f}(A_i')]) = E[\hat{f}(A_i)]$. Finally, $f_i(R(\cdot))$ satisfies $\epsilon$-DP by post-processing, so applying
Lemma 3 to distributions \( \hat{f}_i(\mathcal{R}(a_i)) \) and \( \hat{f}_i(\mathcal{R}(a'_i)) \).

\[
(E[\hat{f}_i(\mathcal{R}(a_i))] - E[\hat{f}_i(\mathcal{R}(a'_i))]])^2 \leq \frac{(e^\epsilon - 1)^2}{e^\epsilon} V[\hat{f}_i(\mathcal{R}(a_i))]
\]

Applying Lemma 4 to the RHS,

\[
\sum_{i=1}^{n} (E[\hat{f}_i(\mathcal{A})] - E[\hat{f}_i(\mathcal{A})])^2 \leq \frac{(e^\epsilon - 1)^2}{e^\epsilon} \sum_{i=1}^{n} V[\hat{f}_i(\mathcal{R}(a_i))]
\]

Summing over \( 1 \leq i \leq n \),

\[
\sum_{i=1}^{n} V[\hat{f}_i(\mathcal{R}(a_i))] \leq \sum_{i=1}^{n} V[\hat{f}_i(\mathcal{R}(a_1), \ldots, \mathcal{R}(a_n))]) = V[\hat{f}(\mathcal{A})]
\]

This completes the proof. \( \square \)

Recall the definition of the max divergence: \( D_{\max}(P, Q) = \sup_{\alpha} \frac{1}{\alpha+1} \ln \frac{P(x)}{Q(x)} \).

**Lemma 3.** Let \( P \) be a distribution on \( \mathbb{R} \). Then, for all distributions \( Q \) on \( \mathbb{R} \) such that \( D_{\max}(P, Q) \leq \epsilon \) and \( D_{\max}(Q, P) \leq \epsilon \),

\[
(E[P] - E[Q])^2 \leq \frac{(e^\epsilon - 1)^2}{e^\epsilon} V[P].
\]

**Proof.** Let \( \alpha = e^\epsilon \). Let \( R \) be the distribution that maximizes \( E[R] - E[P] \) subject to the constraint that \( D_{\max}(P, R) \leq \epsilon \) and \( D_{\max}(R, P) \leq \epsilon \). With a slight abuse of notation, we denote by \( P \) (resp. \( R \)) a random variable generated from a distribution \( P \) (resp. \( R \)). Let \( \mu = \frac{\alpha}{\alpha+1} \) and \( p_\mu \) be such that \( \mu = \Pr[P < p_\mu] \).

**Claim** If there is some \( E \subseteq \langle -\infty, p_\mu \rangle \) such that \( R(E) > 0 \) and \( \frac{P(E)}{R(E)} < \alpha \), then there is some \( F \subseteq [p_\mu, \infty) \) such that \( R(F) > 0 \) and \( \frac{P(F)}{R(F)} > \frac{\alpha}{\alpha+1} \).

**Proof.** Suppose the opposite—that for all \( F \subseteq [p_\mu, \infty) \), either \( R(F) = 0 \) or \( \frac{P(F)}{R(F)} \leq \frac{1}{\alpha} \). Let \( F = [p_\mu, \infty) \). We cannot have \( R(F) = 0 \) because \( P(F) > 0 \) and \( R(F) > \frac{P(F)}{\alpha} \) (by \( D_{\max}(P, R) \leq \epsilon \) and \( \alpha = e^\epsilon \)). Thus, \( \frac{P(F)}{R(F)} \leq \frac{1}{\alpha} \) and equality is tight because \( D_{\max}(P, R) \leq \alpha \).

In other words, \( \frac{P(F)}{R(F)} = \frac{1}{\alpha} \).

However,

\[
\Pr[P < p_\mu] = \frac{P(E) + P(\overline{E})}{R(E) + R(\overline{E})} = \frac{P(E)}{R(E) + R(\overline{E})} + \frac{P(\overline{E})}{R(\overline{E}) + R(\overline{E})} < \alpha \frac{R(\overline{E})}{R(\overline{E}) + R(\overline{E})} + \alpha \frac{R(E)}{R(\overline{E}) + R(\overline{E})}
\]

(by \( \frac{P(E)}{R(E)} < \alpha \) and \( \frac{P(\overline{E})}{R(\overline{E})} \leq \alpha \))

\[
< \alpha
\]

Recall \( \Pr[P < p_\mu] = \frac{\alpha}{\alpha+1} \). We have

\[
\frac{1}{\alpha} = \frac{\Pr[P > p_\mu]}{\Pr[Q > p_\mu]} = \frac{1}{(\alpha+1) \Pr[Q > p_\mu]}
\]

which means \( \Pr[Q > p_\mu] > \frac{\alpha}{\alpha+1} \). Similarly,

\[
\frac{\alpha}{\Pr[Q < p_\mu]} = \frac{\alpha}{(\alpha+1) \Pr[Q < p_\mu]}
\]

which means \( \Pr[Q < p_\mu] > \frac{\alpha}{\alpha+1} \). This gives a contradiction since \( \Pr[Q > p_\mu] + \Pr[Q < p_\mu] = 1 \).

\( \square \)

The claim means that any \( R \) for which we can find an \( E \subseteq \langle -\infty, p_\mu \rangle \) such that \( R(E) > 0 \) and \( \frac{P(E)}{R(E)} < \alpha \) does not maximize \( E[R] - E[P] \). This is because, for such an \( R \), we could find an \( F \) such that \( \frac{P(F)}{R(F)} > \frac{\alpha}{\alpha+1} \). For \( \epsilon \) sufficiently small, we could then define the random variable \( R_\epsilon \) with probability mass function

\[
R_\epsilon(x) = \begin{cases} 
R(x) - \frac{\delta}{R(E)} & x \in E \\
R(x) + \frac{\delta}{R(F)} & x \in F \\
R(x) & \text{otherwise}
\end{cases}
\]

For \( \delta > 0 \), we have \( E[R_\epsilon] > E[R] \), and \( \delta \) can be set sufficiently small such that \( R_\epsilon \) does not break the max-divergence constraints.

Hence, for all \( E \subseteq \langle -\infty, p_\mu \rangle \), if \( R(E) = 0 \), then \( P(E) = 0 \) because \( D_{\max}(P, R) \leq \alpha \). If \( R(E) > 0 \), then \( \frac{P(E)}{R(E)} = \alpha R(E) \).

We can repeat a similar argument as above to conclude that

\[
\Pr[P > p_\mu] = \alpha \Pr[P > p_\mu] = \alpha \Pr[P > p_\mu] + \alpha \Pr[P < p_\mu] E[P | P < p_\mu]
\]

Now that \( R \) is completely defined, we can solve for \( E[R] \).

\[
E[P] = \Pr[P > p_\mu] E[P | P > p_\mu] + \Pr[P < p_\mu] E[P | P < p_\mu] = (1 - \mu) E[P | P > p_\mu] + \mu E[P | P < p_\mu]
\]

We know \( \Pr[R > p_\mu] = \alpha \Pr[P > p_\mu] = \alpha (1 - \mu) \) and \( \Pr[R < p_\mu] = \alpha^{-1} \Pr[P < p_\mu] = \alpha^{-1} \mu \). Since they have the same distribution above \( p_\mu \), we have \( E[R | R > p_\mu] = \alpha \Pr[P > p_\mu] + \alpha^{-1} \mu E[P | P < p_\mu] \) and \( E[R | R < p_\mu] = \alpha \Pr[P > p_\mu] + (1 - \mu) E[P | P < p_\mu] \).

Finally,

\[
E[R] = \Pr[R > p_\mu] E[R | R > p_\mu] + \Pr[R > p_\mu] E[R | R > p_\mu] = \alpha (1 - \mu) E[R | R > p_\mu] + \alpha^{-1} \mu E[R | R < p_\mu]
\]

\[
\alpha (1 - \mu) E[P | P > p_\mu] + \alpha^{-1} \mu E[P | P < p_\mu]
\]

\[
E[P | P > p_\mu] + (1 - \mu) E[P | P < p_\mu]
\]

(by \( \frac{\alpha}{\alpha+1} \iff \alpha = \frac{\mu}{1-\mu} \))

Finally,

\[
E[R] - E[P] = (2\mu - 1) (E[P | P > p_\mu] - E[P | P < p_\mu])
\]

To finish, let \( X \) be a random variable that takes 1 if \( P > p_\mu \) and 0 otherwise. Then by the law of total variance,

\[
E[X] = E[X | X] + E[X | X] \geq E[X | X]
\]
The probability that $P > p_\mu$ is $1 - \mu$; thus $\mathbb{E}[P|X]$ is a random variable taking value $\mathbb{E}[P|P > p_\mu]$ with probability $1 - \mu$ and taking value $\mathbb{E}[P|P < p_\mu]$ otherwise. Its variance is:

$$\mathbb{V}[\mathbb{E}[P|X]] = \mu(1 - \mu)(\mathbb{E}[P|P > p_\mu] - \mathbb{E}[P|P < p_\mu])^2$$

Plugging (37) into this, we get

$$\mathbb{V}[P] \geq \mu(1 - \mu)(\mathbb{E}[P|P > p_\mu] - \mathbb{E}[P|P < p_\mu])^2$$

$$= \mu(1 - \mu) \left( \frac{1}{(2\mu - 1)^2} \right) (\mathbb{E}[R] - \mathbb{E}[P])^2 \quad \text{(by (37))}$$

$$= \frac{\alpha}{(\alpha - 1)^2} (\mathbb{E}[R] - \mathbb{E}[P])^2$$

$$\square$$

**Lemma 4.** Let $P_1, \ldots, P_n$ be independent distributions. Let $f(P_1, \ldots, P_n)$ be a deterministic, real-valued function. Then,

$$\sum_{i=1}^n \mathbb{V}_i \mathbb{E} \left[ f(P_1, \ldots, P_n) \right] \leq \mathbb{V}[f(P_1, \ldots, P_n)]$$

**Proof.** Write $P^S = \{P_i : i \in S\}$. We can write the claim as:

$$\sum_{i=1}^n \mathbb{V}_{P_i} \mathbb{E} \mathbb{P}_{\{i\}} \left[ f(P_1, \ldots, P_n) \right] \leq \mathbb{V}[f(P_1, \ldots, P_n)]$$

We will proceed by induction on $n$. The base case $n = 1$ is trivial. For the general case, we will use the law of total variance:

$$\mathbb{V}[Y] = \mathbb{E}_X \mathbb{V}[Y|X] + \mathbb{V}_X \mathbb{E}[Y|X]$$

Let $Y = f(P_1, \ldots, P_n)$ and $X = P_1$. We have $Y|X = f(P_1, \ldots, P_n)|P_1$. Because of independence, $f(P_1, \ldots, P_n)|(P_1 = x) = f(P_1, \ldots, P_{n-1}, x)$. Plugging into the law of total variance, we get

$$\mathbb{V}_P[f(P_1, \ldots, P_n)] = \mathbb{E}_P \mathbb{V}_{P_{\pi^{-1}}}[f(P_1, \ldots, P_n)]$$

Applying Lemma 5 to the RHS, we get

$$\mathbb{V}_P[f(P_1, \ldots, P_n)] \geq \mathbb{V}_{P_{\pi^{-1}}}[\mathbb{E}_P[f(P_1, \ldots, P_n)]]$$

Applying the inductive hypothesis to the function $g(P_1, \ldots, P_{n-1}) = \mathbb{E}_{P_1}[f(P_1, \ldots, P_n)]$, we have

$$\mathbb{V}_{P_{\pi^{-1}}}[\mathbb{E}_P[f(P_1, \ldots, P_n)]] = \sum_{i=1}^{n-1} \mathbb{V}_{P_i} \mathbb{E}_{P_{\pi^{-1}}} \mathbb{P}_{\{i\}} \mathbb{E}_P[f(P_1, \ldots, P_n)]$$

$$= \sum_{i=1}^{n-1} \mathbb{V}_{P_i} \mathbb{E}_{\pi^{-1}} \mathbb{P}_{\{i\}} \left[ f(P_1, \ldots, P_n) \right]$$

Applying the inductive hypothesis to the function $g(P_1, \ldots, P_{n-1}) = \mathbb{E}_{P_1}[f(P_1, \ldots, P_n)]$, we have

$$\mathbb{V}_{P_{\pi^{-1}}}[\mathbb{E}_P[f(P_1, \ldots, P_n)]] = \sum_{i=1}^{n-1} \mathbb{V}_{P_i} \mathbb{E}_{P_{\pi^{-1}}} \mathbb{P}_{\{i\}} \mathbb{E}_P[f(P_1, \ldots, P_n)]$$

$$= \sum_{i=1}^{n-1} \mathbb{V}_{P_i} \mathbb{E}_{\pi^{-1}} \mathbb{P}_{\{i\}} \left[ f(P_1, \ldots, P_n) \right]$$

Combining (38) and (39), we are done. $\square$

**Lemma 5.** If $X, Y$ are independent, then for any real-valued function $f$,

$$\mathbb{E}_X \mathbb{V}_Y f(X,Y) \geq \mathbb{V}_Y \mathbb{E}_X f(X,Y)$$

**Proof.** Let the domain of $X$ be $X$ and suppose it is discrete: $X = \{x_1, \ldots, x_n\}$. Now,

$$\mathbb{V}_Y[\mathbb{E}_X f(X,Y)] = \mathbb{V}_Y \left[ \sum_{i=1}^n f(x_i, Y) \Pr[X = x_i] \right]$$

Let $\sigma^2_{x_i} = \mathbb{V}_Y[f(x_i, Y)]$. Now,

$$\mathbb{V}_Y \left[ \sum_{i=1}^n f(x_i, Y) \Pr[X = x_i] \right] = \sum_{1 \leq i, j \leq n} \Pr[X = x_i] \Pr[X = x_j] \text{Cov}(f(x_i, Y), f(x_j, Y))$$

$$\leq \sum_{1 \leq i, j \leq n} \Pr[X = x_i] \Pr[X = x_j] \sigma_{x_i} \sigma_{x_j}$$

$$= \left( \sum_{i=1}^n \Pr[X = x_i] \sigma_{x_i} \right)^2$$

$$\leq \mathbb{E}_X [\sigma^2_X]^2$$

We may conclude the result for continuous $X$ by using an arbitrarily fine discretization of $X$. $\square$

**Lemma 6.** Suppose $x_1, \ldots, x_L$ and $y_1, \ldots, y_L$ are real numbers such that $y_i + D \leq y_{i+1}$. Suppose further that $\frac{1}{L-1} \sum_{i=1}^{L-1} (x_{i+1} - x_i)^2 \leq \frac{D^2}{L}$. Then, $\frac{1}{L-1} \sum_{i=1}^{L-1} (x_i - y_i)^2 \geq \Omega(L^2D^2)$.

**Proof.** We first compute $\sup_{i,j} |x_i - x_j|$. There is a monotonic sequence $\{x'_i : 1 \leq i \leq L\}$ such that $\frac{1}{L-1} \sum_{i=1}^{L-1} (x'_i - x'_1)^2 = \frac{1}{L-1} \sum_{i=1}^{L-1} (x_{i+1} - x'_1)^2$ and $\sup_{i,j} |x_i - x_j| \leq \sup_{i,j} |x'_i - x'_j|$. The sequence is given by $x'_1 = x_1$, $x'_i = x_{i-1}$, $x'_L = x_L$, and $\{x'_i\}$ is monotonic. Because $\{x'_i : 1 \leq i \leq L\}$ is monotonic, we have $\sup_{i,j} |x'_i - x'_j| = |x'_L - x'_1|$. Using Jensen’s inequality, $\frac{1}{L-1} \sum_{i=1}^{L-1} (x'_i - x'_1)^2 \geq \left( \frac{1}{L-1} \sum_{i=1}^{L-1} (x'_i - x'_1) \right)^2 \geq \left( \frac{1}{L-1} (x'_L - x'_1) \right)^2$. Therefore, $\sup_{i,j} |x_i - x_j|^2 \leq (x'_L - x'_1)^2 \leq (L - 1) \sum_{i=1}^{L-1} (x_{i+1} - x'_1)^2 \leq \frac{LD^2}{4}$. The given information about $y$ implies that $|y_i - y_j| \geq |i - j|D$. For any $1 \leq i < L$, let $k = i + \frac{3}{4}$. Using the triangle inequality,

$$|x_i - y_i| + |y_k - x_k| + |x_k - x_i| \geq |y_k - y_i| \geq |i - k|D = \frac{3LD}{4}$$
Because \(|x_i - x_k| \leq \sup_j |x_j - x_k| \leq \frac{LD}{2}\), this implies
\[|x_i - y_l| + |y_k - x_k| \geq \frac{LD}{4}\]
and so for all \(1 \leq i < \frac{L}{4}\), either \(|x_i - y_l| > \frac{LD}{8}\) or \(|x_{i+3/4} - y_{i+3/4}| \geq \frac{LD}{8}\). Thus, \(\sum_{i=\frac{L}{4}}^{L} (x_i - y_l)^2 + \sum_{i=\frac{L}{4}}^{L} (x_{i} - y_l)^2 \geq \frac{L (LD)^2}{4}\), and the result is shown.

**Lemma 7.** \((n/2 - \sqrt{n}) \geq K \frac{2n}{\sqrt{n}}\) for a global constant \(K\).

**Proof.** It is a standard result that \((\frac{n}{2}) = (1 + O(1)) \frac{2^n}{\sqrt{n}}\)

However,
\[
\frac{(n/2 - \sqrt{n})}{(n/2)} = \frac{(n/2)! (n/2)!}{(n/2 - \sqrt{n})! (n/2 + \sqrt{n})!} = \prod_{i=0}^{\sqrt{n}-1} \frac{n/2 - i}{n/2 + \sqrt{n} - i}
\]

We know that
\[
\left(\frac{n/2 - \sqrt{n}}{n/2}\right)^{\sqrt{n}} \leq \prod_{i=0}^{\sqrt{n}-1} \frac{n/2 - i}{n/2 + \sqrt{n} - i} \leq \left(\frac{n/2}{n/2 + \sqrt{n}}\right)^{\sqrt{n}}
\]
\[
\left(\frac{\sqrt{n}/2 - 1}{\sqrt{n}/2}\right)^{\sqrt{n}} \leq \prod_{i=0}^{\sqrt{n}-1} \frac{n/2 - i}{n/2 + \sqrt{n} - i} \leq \left(\frac{\sqrt{n}/2}{\sqrt{n}/2 + 1}\right)^{\sqrt{n}}
\]

However, both the LHS and the RHS approach \(e^{-2}\) (because \(\lim_{n \to \infty} (1 + \frac{1}{n})^n = e^0\)) and the result is shown. 

**B.11 Construction of an \((n, \frac{d_{max}}{2} - 2)\) monotone cube for \(f_\triangle\)**

Suppose \(n\) is even and \(d_{max}\) is divisible by 4. Since \(d_{max} < n\), it is possible to write \(n = k \frac{d_{max}}{2} + r\) for integers \(k, r\) such that \(k \geq 1\) and \(1 \leq r < \frac{d_{max}}{2}\). Because \(k \frac{d_{max}}{2}\) and \(n\) are even, we must have \(r\) is even. Now, write \(n = (k - 1) \frac{d_{max}}{2} + (r + \frac{d_{max}}{2})\).

This means we can define the graph \(G\) on \(n\) vertices consisting of \((k - 1)\) cliques of size \(\frac{d_{max}}{2}\) and one final clique of an even size between \(\frac{d_{max}}{2}\) and \(d_{max}\) with all cliques disjoint.

Since \(G\) consists of even-sized cliques, it contains a perfect matching \(M\). Let \(G' = G \setminus M\). Each edge in \(G\) is part of at least \(\frac{d_{max}}{2} - 2\) triangles. For each pair of edges in \(M\), the triangles of \(G\) of which they are part are disjoint. Thus, for any \(N \subseteq M\) with \(e \in N\), removing \(e\) from \(G \setminus N\) will remove at least \(\frac{d_{max}}{2} - 2\) triangles. This implies \(\mathcal{A} = \{G \cup N : N \subseteq M\}\) is an \((n, \frac{d_{max}}{2} - 2)\) monotone cube for \(f_\triangle\).