GSHOP: Towards Flexible Pricing for Graph Statistics

Chen Chen[§], Ye Yuan[§], Zhenyu Wen[†], Yu-Ping Wang[§], Guoren Wang[§],
[§]School of Computer Science and Technology, Beijing Institute of Technology, Beijing, China
[†]Institute of Cyberspace Security, Zhejiang University of Technology, Hangzhou, China
[§]{cchen, yuan-ye, wyp_cs, wanggr}@bit.edu.cn; [†]zhenyuwen@zjut.edu.cn

***Abstract*—The prevalence of online query services in human life has attracted significant interest from the fields of economics and databases in determining appropriate pricing for such services. Simultaneously, the utilization of graph analytics across various domains has resulted in substantial social and economic benefits in recent years. As the adoption of graph analytics continues to expand, there is a corresponding need to establish fair pricing models for the information contributed by each participant in the data ecosystem. However, current query-based pricing frameworks cannot be applied to price graph statistics, as they fail to consider buyers' affordability and prevent arbitrage trading. To address this gap, in this paper, we propose a novel framework GSHOP for pricing graph statistic queries. Instead of pricing a precise answer for a query, our framework offers the flexibility to price a set of answers injected with noise. Based on the framework, data owners initially create and publish extended local views (ELVs) to represent their graph data. Additionally, it allows buyers to tolerate a certain degree of noise added to the answer to reduce their payments. The framework accurately quantifies the relationship between noise and price to ensure that payment and compensation are reasonable for the buyer and owners, respectively. We also propose algorithms specifically designed for fundamental graph statistics, including node degrees and subgraph counts such as k-stars and k-cliques. Furthermore, we formally prove that the pricing framework is arbitrage-free. Extensive experimental results on real-life graph data validate the good performance of the proposed framework and algorithms.**

I. INTRODUCTION

In the era of big data, analysis of network statistics plays a pivotal role in identifying meaningful patterns within graph data across various domains [1], [2], including finance [3], bioinformatics [4] and healthcare [5]. Research and industrial efforts have largely focused on developing efficient algorithms to process graph data to meet the needs of different analysis tasks [6]. However, limited research so far has considered the cost of obtaining and managing data for graph data analytics.

Example 1. Bano, an international anti-fraud commissioner, wants to investigate the closeness of the relationship between all employees of a company and other company employees. To this end, he needs to acquire the clustering coefficient of the entire data graph G. This coefficient can be obtained by $\frac{3f_{\triangle}(G)}{f_{2\star}(G)}$, where $f_{\triangle}(G)$ and $f_{2\star}(G)$ are the triangle and 2-star counts within G respectively. Although the data may be available online, Bano lacks the technical skills and time to process it. The data marketplace would allow Bano to be charged based on his query task and budget constraint.

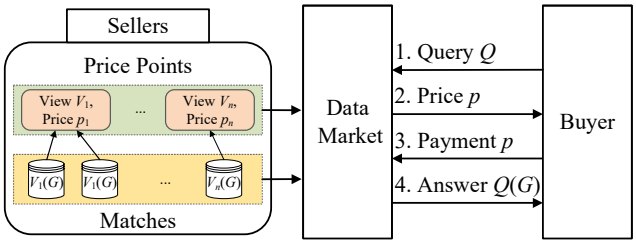


Fig. 1: A framework for query-based pricing.

However, data analysts like Bano usually spend a lot of money to buy datasets from companies (e.g., Bloomberg, Twitter), or through data markets (e.g., Acxiom [7], BDEX [8]) for conducting their analytics tasks. Existing pricing schemes force users to buy the whole dataset or support simplistic pricing mechanisms (e.g., filtering by keywords) [8], [9]. Also, some analysis tasks may be one-time, which can cause a huge waste of money. As a result, the data sellers may lose a lot of customers due to the unflexible pricing schemes. Thus, *there is a need to change the data markets that sell data only to markets that support query-based pricing.*

Query-based pricing for graph data. A high-level overview of the traditional query-based data market [10], [11] is presented in Figure 1. The data market involves three key agents: the *seller*, who provides datasets; the *buyer*, who seeks to purchase query answers; and the *market*, which facilitates interactions between sellers and buyers. Initially, each data seller (or data owner) defines a price point (V_i, p_i), where V_i represents a view of the data graph G, and p_i denotes its corresponding price. In the platform, a view V_i in G may have multiple matches $V_i(G)$ maintained. When a buyer submits a query Q over G, the market automatically computes the price p and the query answer $Q(G)$ by considering various combinations of views. However, if a buyer needs to query large amounts of data but with a limited budget, this transaction may not take place. To attract more users, the market should provide a flexible pricing mechanism to meet users' various requirements.

Idea. To this end, the proposed GSHOP increases the flexibility of the prices by improving the diversity of the answers to a query. In other words, the *market* can add some noise to a query's answer while reducing the price of answering the query.

Challenges. To implement the GSHOP are facing the following technical challenges. C1) *Reasonable compensation to data owners*. As Figure 1 illustrated each owner should receive a compensation p_i as the reward for sharing data. However, when the data market adds noise to the answer to meet the buyer's budget constraints, determining appropriate compensation becomes challenging. To be precise, the noises are added to the final answer and it is difficult to determine the contribution of each data owner. C2) *Design an arbitrage-free mechanism to support diversity pricing*. To increase the diversity of the pricing for a given query, GSHOP may provide noisy answers. Simultaneously, GSHOP must prevent *arbitrage* opportunities, where a buyer may attempt to exploit the system by purchasing a combination of queries with high noise rather than a single query with low noise, in order to obtain a cheaper price. This increases the difficulty of designing an arbitrage-free mechanism to ensure fair transactions.

Contributions. Our key technical contribution is a simple and efficient query-based pricing framework GSHOP that employs a *noise-injection* mechanism to enable "flexible" pricing for graph statistic queries with formal guarantees. When a buyer requests a graph statistic query with a specified tolerance for noise, the data market adds random Laplace noise to the exact count and returns the perturbed result to the buyer. We demonstrate an inverse monotonic relationship between the accuracy of the answer and the variance of the noise. The pricing mechanism determines the price based on the variance of the noise injected into the true answer for fundamental graph statistics including node degree and subgraph counts such as k-stars and k-cliques. Lower variance noise implies a higher expected accuracy, and thus commands a higher price, while higher variance noise results in a lower expected accuracy and a lower price. This enables the buyer to either choose cheaper but less accurate answers or more accurate yet more expensive ones [12].

Our proposed pricing mechanism comes with a concise characterization of when a pricing function is provably arbitrage-free. In the main theoretical result of this paper, we demonstrate the pricing function of GSHOP, which satisfies both variance and query constraints, thereby guaranteeing an arbitrage-free environment.

Finally, we conduct an extensive empirical evaluation using real-life datasets. Our experiments validate the effectiveness of the proposed framework in providing a viable pricing strategy that offers diverse options to buyers. This enables them to pay at least 36.7% of the original price while still obtaining meaningful answers. Moreover, the framework ensures appropriate compensation for data owners, with the most significant information contributor receiving 81.1% of the original compensation.

In summary, this paper makes the following contributions.
(1) *Noisy pricing framework*. We propose a pricing framework GSHOP via a noise injection approach to provide a better service for users (Section III). The mechanism can accurately quantify the relation between noise and payment based on the compensation of each data owner.
(2) *Information contribution quantification*. We establish a metric for evaluating the information contributed by data owners, which serves as the basis for determining their compensation (Section IV). To ensure the fairness of this assessment, we propose algorithms specifically tailored for node degree, counting k-stars and k-cliques. These algorithms are designed to accurately estimate the information contribution of each individual data owner.
(3) *Arbitrage-free*. We formally prove that the noisy pricing framework is arbitrage-free (Section V). The framework achieves arbitrage-free by imposing a lower bound on the ratio between the price of highly noisy answers and low noisy answers.
(4) *Experiment*. We conduct comprehensive experiments on real-life graph data to evaluate the proposed noisy pricing framework and related algorithms (Section VI). The experimental results validate that our proposed algorithms provide reasonable compensation for data owners, a large affordability ratio, and thus accessibility for the buyer.

II. Problem Definition

A. Basic Concepts

Data graph. We consider an undirected graph, defined as $G = (V, E)$. where (1) $V = \{v_1, ..., v_n\}$ is the set of nodes; (2) $E \subseteq V \times V$ is a set of edges, in which (v_i, v_j) denotes a relationship between v_i and v_j.

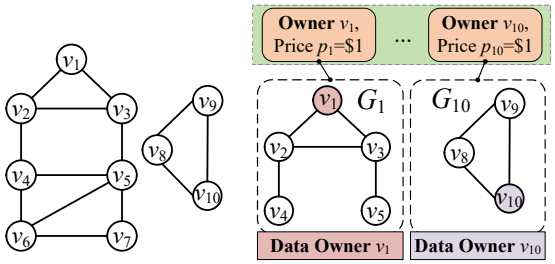
In many real-world applications, a data graph G is typically distributed among multiple data owners, each of whom possesses a limited local view of the complete data graph. For instance, the contact lists in every owner's mobile phones could be pieced together to form a giant contact graph, though no single data owner is aware of the whole social network structure.

Extended local view. Given a node $v_i \in V$, its two-hop extended view G_i consists of:

1) v_i's one-hop neighbors:$\{u|u \in V \wedge (u, v_i) \in E\}$.
2) Edges involving v_i: $\{e = (u, v_i)|e \in E\}$.
3) v_i's two-hop neighbors: $\{w|\exists u \in V, (u, v_i) \in E \wedge (u, w) \in E\}$.
4) Edges involving v_i's one-hop neighbors: $\{e = (u, w)|e \in E \wedge (u, v_i) \in E\}$.

For instance, with the default setting of Facebook (facebook.com), a user allows each of her friends to see all her connections. In the offline world, we also commonly accumulate knowledge on the relationships between our friends, e.g., when we attend a social event together. Figure 2(a) shows a data graph example made up by $\{G_1, ..., G_{10}\}$ provided by $\{v_1, ..., v_{10}\}$. The ELVs G_1 and G_{10} in Figure 2(b) are ELVs of v_1 and v_{10}.

Graph statistic query. We consider three fundamental query f of a graph G, i.e., node degree, k-star and k-clique, denoted by $f_d(G)$, $f_{k\star}(G)$ and $f_{k\mathbb{C}}(G)$. Formally, $f(G_i)$ denotes the graph statistic involved v_i. For example, $f_{\triangle}(G_i)$ denotes the number of triangles involved v_i in G_i.



(a) The data graph G (b) ELVs and Price Points

Fig. 2: The data graph and ELVs.

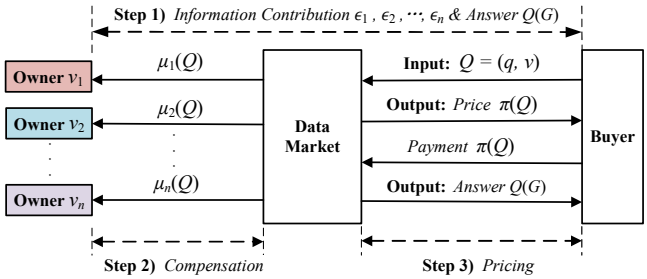


Fig. 3: The framework of the data market.

Price points. A data owner usually specifies an explicit price point (v_i, p_i), where v_i is the owner of an ELV G_i and $p_i \in \mathbb{R}^+$ is a fixed price for the true answer. We denote a finite set of price points $\mathcal{S}$ as $\mathcal{S} = \{(v_1, p_1), ..., (v_n, p_n)\}$.

Based on price points given by data owners, the data market can compute a reasonable price for the query. In detail, the price is the summation of all price points whose corresponding ELVs are involved in this graph statistic query.

Example 1: The following example illustrates how to price a triangle counting query. Figure 2 shows a graph G with price points are $\{(v_1, \$1), ..., (v_{10}, \$1)\}$ (i.e., $p_i = \$1$ for $1 \leq i \leq 10$). When the buyer requests a triangle counting query for G, the data market calculates the number of triangles in each G_i. For example, it can be seen there exists one triangle involved v_1 in G_1, i.e., $f_\triangle(G_1) = 1$. Then the market can charges the buyer with a price $\sum_{i=1}^{10} p_i = \$10$ and returns $\{1, 1, 1, 1, 2, 2, 1, 1, 1, 1\}$ to him. Finally, the buyer can calculate the total number of triangles in G based on $f_\triangle(G_i)$, i.e., $f(G) = \frac{1}{3}\sum_{i=1}^{10} f_\triangle(G_i) = 4$ because each triangle is reported by its three distinct nodes.

Because the price of the accurate answer is usually high, to offer more choices to data buyers, the data market also sells noisy answers. Perturbation is a tool to lower the price for the buyer [12]. The buyer specifies how much noise he can tolerate when issuing the query. To formalize the relationship between the answer's noise and its price is one of the main goals of this paper.

Noise. Each data buyer can request his query $Q = (f, \mathbf{v})$, where f is a graph statistic query, and $\mathbf{v}$ denotes a tolerable variance of noise added to the true answer $\{f(G_1), ..., f(G_n)\}$. This feature gives the buyer more pricing options by increasing $\mathbf{v}$. We note that the self-defined noise variance allows the buyer to adjust the answer's accuracy with a certain confidence based on Chebyshev's inequality [13].

B. Price flexible data market

Figure 3 shows the high-level framework of the price flexible data market, composed of three steps:

Step 1) The buyer requests a query $Q = (f, \mathbf{v})$, where f represents a graph statistic query and $\mathbf{v}$ denotes the acceptable variance. The data market first computes the true answer $\{f(G_1), ..., f(G_n)\}$, and then introduces noise sampled from a distribution with a mean of 0 and a variance of $\mathbf{v}$ to obtain $Q(G)$. By responding to the query Q, the data market leverages information ϵ_i contributed by each data owner v_i. When the exact graph statistics of G_i is returned, we set $\epsilon_i \to \infty$ to indicate the maximal information contribution.

Step 2) Each data owner provides the price point (v_i, p_i), which means that if $\mathbf{v} = 0$ such that $\epsilon_i \to \infty$, the data market needs to pay p_i. Based on this setting, the data market compensates the data owner v_i with $\mu_i(Q)$ for her information contribution ϵ_i properly.

Step 3) The data market charges the buyer with price $\pi(Q)$, where $\pi(Q)$ is sufficient to cover all compensations such that $\pi(Q) \geq \sum_{i=1}^{n} \mu_i(Q)$. After receiving the payment $\pi(Q)$ from the buyer, the data market gives the answer $Q(G)$ to him.

Arbitrage is an undesirable property that allows a buyer to obtain the answer to a query more cheaply than its advertised price by deriving the answer from a less expensive alternative set of queries [14].

Example 2: As illustrated in Example 1, the price of the accurate answer $f_\triangle(G)$ is \$10, which is considered too expensive by the buyer. Instead, he is presented with an alternative option to purchasing a perturbed answer $Q(G)$ added noise with variance 2 for \$2, which produces an error of ± 2 with 50% confidence. Suppose the buyer is also offered another noisy option with variance 20 for \$0.1. In this scenario, no savvy buyer would pay for the previous answer because he could purchase the new answer ten times for a total cost of \$1 and compute their averages. This is an example of arbitrage and the data market should avoid it.

Problem definition. Given a data graph G and the data market framework above, our goal is to enable quantifying the relationship between the payment and the noise of the answer specified by the buyer. That is, given a query $Q = (f, \mathbf{v})$, we aim to determine the compensation $\mu_i(Q)$ for each data owner v_i based on an information contribution measure ϵ_i, which is expected to decrease as $\mathbf{v}$ increases. Then we give the payment $\pi(Q)$ required to ensure that $\pi(Q) \geq \sum_{i=1}^{n} \mu_i(Q)$, in order to cover all compensations, while the pricing function $\pi(Q)$ is arbitrage-free simultaneously.

III. Query Pricing Framework

A. System Framework

In this section, we introduce an algorithm framework for pricing noisy graph statistics. The framework we examine in Figure 3 is for data markets with noisy pricing. The buyer

initiates a query $Q = (f, \mathbf{v})$, where f is a graph statistic query, and $\mathbf{v}$ indicates the acceptable level of noise to be added to the true answer.

To obtain the contribution of the data owner v_i, the framework employs G_i, which is associated with v_i. By comparing the outputs generated with and without the corresponding edge in G_i, the framework can evaluate the impact of v_i on the overall result. The parameter ϵ_i represents the impact and can be used to quantify the information contributed by v_i to the graph [15].

It restricts the noise to a Laplace distribution since there is a formula connecting ϵ_i to the variance $\mathbf{v}$ [6]. The data market can quantify the relationship between the compensation and the $\mathbf{v}$ for each data owner using the information contribution ϵ_i [16]. Based on each data owner's price point (v_i, p_i), the data market compensates the owner v_i with $\mu_i(Q)$ according to her contribution ϵ_i. To maintain the utility of the data market at zero or higher, the price $\pi(Q)$ charged to the buyer must be sufficient to cover all the compensations.

In general, the query pricing framework GSHOP consists of three parts: computing the answer and information contribution, getting the compensation, and calculating the price paid by the buyer. The Algorithm 1 shows the details of our framework.

Algorithm 1: Framework of GSHOP

Input: Query $Q = (f, \mathbf{v})$, data graph G, price points $\mathcal{S}$
Output: Price $\pi(Q)$, answer $Q(G)$

// 1) Compute contribution and answer

1 $\{\epsilon_1, ...\epsilon_n\}, Q(G) = \begin{cases} \text{Node Degree} & (a) \\ k\text{-star} & (b) \\ k\text{-clique} & (c) \end{cases}$

// 2) Compute each v_i's compensation

2 **for** $i = 1$ **to** n **do**

3 $\quad \mu_i(Q) = \frac{2p_i}{\Pi} \arctan(b_i \epsilon_i)$;

// 3) Compute the query's price

4 $\pi(Q) = c \sum_{i=1}^{n} \mu_i(Q)$;

Step 1) Compute the answer $Q(G)$ for the query $Q = (f, \mathbf{v})$ and calculate the information contribution ϵ_i of each data owner (line 1). The data market begins by calculating the exact statistics $\{f(G_1), ..., f(G_n)\}$ and then introduces noise sampled from a Laplace distribution with a mean of 0 and a variance of $\mathbf{v}$ to obtain the corresponding answer $Q(G)$. To ensure fair compensation for each data owner, the data market must provide a reasonable quantification of the contribution of their data. In order to achieve this, the information contribution ϵ_i is utilized to compare the output of a mechanism with and without the inclusion of each data owner's data. To mathematically derive the information contribution of data owners, we employ the concept of *sensitivity*. By exploring sensitivity, we can accurately quantify the impact of each data owner's contribution. Specifically, we design algorithms for fundamental graph statistics, including node degrees and subgraph counts such as k-stars and k-cliques.

Step 2) Get the compensation $\mu_i(Q)$ for each participant v_i (lines 2-3). The data market must compensate each data owner for her information contribution ϵ_i with $\mu_i(Q)$. The price point (v_i, p_i) means that the price of the true answer $f(G_i)$ equals p_i. To give a bounded output for $\epsilon_i \to \infty$, we apply the $\arctan(\cdot)$ function to handle it. We set $\mu_i(Q) = \frac{2p_i}{\Pi} \arctan(b_i \epsilon_i)$ for a constant $b_i > 0$, because when $\epsilon_i \to \infty$, the data market gives the true answer $f(G)$ and compensates the data owner v_i with $\mu_i(Q) = p_i$.

Step 3) Calculate the payment price $\pi(Q)$ for the buyer (line 4). To make price $\pi(Q)$ sufficient to cover all compensation $\mu_i(Q)$, it set $\pi(Q) = c \sum_{i=1}^{n} \mu_i(Q)$ and $c \geq 1$.

To attack the hard problem of designing an arbitrage-free pricing function, the most important step is to measure every owner's information contribution reasonably in Step 1).

To this end, we first introduce a general method to measure the information contributed by the variance of noise and the sensitivity in Section III-B, and propose algorithms to impose a bound ϵ_i on the maximum ratio between the probabilities of returning a graph statistic result with and without any edge from G_i in Section IV. Finally, we prove that all the pricing functions $\pi(Q)$ based on these three methods are arbitrage-free in Section V.

B. Technical Foundation

When the data market answers noisy graph statistics with a randomized mechanism $\mathcal{M}$ and sells it to the buyer, some individual information of data owners would be employed. To give a reasonable quantification of their information contribution, we compare the output of $\mathcal{M}$ with and without their edges in G_i. Based on disciplines of the perturbation mechanism, we formally define the individual contribution ϵ_i of the data owner v_i for a query Q, and further give its upper bound related to the sensitivity $DS_f(v_i)$ and the variance of noise $\mathbf{v}$.

We first consider a pair of neighboring graphs G and G' differing in one edge, and $\{G'_1, ..., G'_n\}$ are the neighboring ELVs of $\{G_1, ..., G_n\}$ with respect to G and G'. In particular, we first define the notion of neighboring ELV, as follows.

Definition 1 (Neighboring extended local view): Given a graph $G = (V, E)$, a participant $v_i \in V$, its ELV $G_i \subseteq G$, and a neighboring graph G' of G. The neighboring graph G'_i of G_i is the extended view of v_i in G'.

Based on the notion of neighboring ELVs, we define the notion of global sensitivity as follows.

Definition 2 (Global Sensitivity): Given a set of nodes $V = \{v_i | 1 \leq i \leq n\}$, and a query f, the global sensitivity of f at v_i is defined as:

$$DS_f(v_i) = \max_{G, G'} \sum_{j=1}^{n} |f(G_j) - f(G'_j)| \tag{1}$$

where G and G' are two arbitrary neighboring graphs that only differ in one edge $e \in G_i$.

Intuitively, $DS_f(v_i)$ computes the maximum sum of subgraph count differences over all pairs of neighboring ELVs that differ from G_i in exactly one incident edge.

By comparing the output of the mechanism $\mathcal{M}$ over the target edges, we define individual information contribution as follows.

Definition 3 (Individual Information Contribution): The individual information contribution of data owner v_i in the class of randomized mechanisms $\{\mathcal{M}_j|1 \leq j \leq n\}$ with domain $\{G_j|1 \leq j \leq n\}$ and outputs $\{O_j|1 \leq j \leq n\}$ considering two arbitrary neighboring graphs G and G' which only differ in one edge $e \in G_i$, is:

$$\epsilon_i = \max_{G,G'} |\log \frac{P(\mathcal{M}_1(G_1) = O_1, ..., \mathcal{M}_n(G_n) = O_n)}{P(\mathcal{M}_1(G'_1) = O_1, ..., \mathcal{M}_n(G'_n) = O_n)}| \quad (2)$$

with probability $1-\delta$, where δ is typically less than the inverse of the number of edges.

We further give an upper bound of individual information contribution ϵ_i, where the randomized mechanism $\mathcal{M}$ is known to be the Laplace mechanism. In Laplace mechanism, given a query $Q = (f, \mathbf{v})$, data owners directly report their noisy versions of statistics by injecting the Laplace noise.

Theorem 1: Let $\mathcal{M}$ be the Laplace mechanism, $DS_f(v_i)$ be the global sensitivity of v_i with the graph statistic query f, and $\mathbf{v}$ be the variance of noise. The individual information contribution of the data owner v_i is bounded by:

$$\epsilon_i \leq \frac{DS_f(v_i)}{\sqrt{\mathbf{v}/2}} \quad (3)$$

Proof. In the Laplace mechanism, the noise η is drawn from the Laplace distribution $Lap(\lambda)$, where $\lambda = \sqrt{\mathbf{v}/2}$. Let $f(G_j)$ and $f(G'_j)$ be the graph statistics on G_j and G'_j, where G_j and G'_j are neighboring ELVs of v_j with respect to G and G' (G and G' are neighboring graphs that only differ in $e \in G_i$). Since the data market outputs O_j by injecting Laplace noise $Lap(\lambda)$ into the graph statistics, we then derive that:

$$\begin{aligned}
\epsilon_i &= \max_{G,G'} |\log \frac{P(\mathcal{M}_1(G_1) = O_1, ..., \mathcal{M}_n(G_n) = O_n)}{P(\mathcal{M}_1(G'_1) = O_1, ..., \mathcal{M}_n(G'_n) = O_n)}| \\
&= \max_{G,G'} |\log \frac{P(f(G_1) + \eta = O_1, ..., f(G_n) + \eta = O_n)}{P(f(G'_1) + \eta = O_1, ..., f(G'_n) + \eta = O_n)}| \\
&= \max_{G,G'} |\log \frac{\prod_{j=1}^{n} \exp(\frac{1}{\lambda}|O_j - f(G_j)|)}{\prod_{j=1}^{n} \exp(\frac{1}{\lambda}|O_j - f(G'_j)|)}| \\
&\leq \max_{G,G'} \frac{\sum_{j=1}^{n} |f(G_j) - f(G'_j)|}{\lambda} \leq \frac{DS_f(v_i)}{\lambda}
\end{aligned}$$

where $DS_f(v_i)$ is the upper bound of $\sum_{j=1}^{n} |f(G_j) - f(G'_j)|$.

IV. Algorithms for Measuring Contribution

In this section, we focus on node degrees and subgraph counts, specifically k-star and k-clique, to demonstrate the methodology for calculating the individual information contribution, denoted as ϵ_i, for each owner.

A. Node Degree

Given that adding or removing an edge in G only affects the degrees of two nodes, each by 1, the sensitivity of the set $\{d(v_1), ..., d(v_n)\}$ is 2. In other words, the sensitivity of the degree function f_d is $DS_{f_d}(v_i) = 2$. Consequently, the information contribution of each data owner, when reporting $\{d(v_1), ..., d(v_n)\}$ by injecting Laplace noise with variance $\mathbf{v}$, is bounded by $\epsilon_i \leq 2/\sqrt{\mathbf{v}/2}$.

This is consistent with intuition that, when every node in the social network reports its degree status, there will inevitably be two degree information changes in the result, whether a particular edge of it is included or not. Therefore, each data owner has an equal information contribution $2/\sqrt{\mathbf{v}/2}$.

B. k-star Counting

A k-star refers to a structure where a central node is connected to k other nodes. Recall that the number of k-stars in G_i is equal to $\binom{d(v_i)}{k}$, where $d(v_i)$ represents the degree of node i, for all $k \geq 2$. When adding or removing an edge connected to v_i, the potential impact on k-stars is limited to a maximum of $2\binom{n-1}{k-1}$. This limitation arises because adding an edge (v_i, v_j) affects at most $\binom{d(v_i)}{k-1} + \binom{d(v_j)}{k-1}$ k-stars [17], and both $d(v_i)$ and $d(v_j)$ cannot exceed $n-1$. Given this scenario, each data owner's information contribution is bounded by $\epsilon_i \leq 2\binom{n-1}{k-1}/\sqrt{\mathbf{v}/2}$.

However, it should be noted that the magnitude of contribution, denoted as ϵ_i, for k-star counting is not equal across all data owners. This is because adding or removing an edge in different G_i can impact a different number of k-stars. In the context of privacy-preserving data analysis, a local measure of sensitivity is introduced in [18]. This measure takes into account the specific characteristics of each data owner's graph and provides a more accurate assessment of their individual contribution.

Definition 4 (Local sensitivity): Given a global graph $G = (V, E)$ containing nodes $V = \{v_i|1 \leq i \leq n\}$, and a query f, the local sensitivity of f at v_i is defined as:

$$LS_f(v_i) = \max_{G'} \sum_{j=1}^{n} |f(G_j) - f(G'_j)| \quad (4)$$

G' is a neighboring graph of G which only differs in one edge $e \in G_i$. Observe that $DS_f(v_i) = \max_G LS_f(v_i)$.

We utilizes the local sensitivity $LS_{f_{k\star}}(v_i)$ defined in Definition 4, which is an instance-specific sensitivity measurement, the value of which is dependent on its own graph G_i. Given any ELV G_i owned by v_i, the local sensitivity of triangle counting is the maximum change of the k-star counting result induced by adding an edge connected to v_i as:

$$LS_{f_{k\star}}(v_i) = \binom{d(v_i)}{k-1} + \max_{v_j \in N(v_i)} \binom{d(v_j)}{k-1} \quad (5)$$

where $N(v_i)$ denotes a set of nodes that neighbors node v_i.

While calibrating the magnitude of contribution to $LS_{f_{k\star}}(v_i)$ may seem more reasonable, it is not directly feasible to compute the information contribution of v_i as $LS_{f_{k\star}}(v_i)/\sqrt{\mathbf{v}/2}$. This is because the contribution magnitude itself may contain information about v_i, and thus v_i may actually contribute more information than $LS_{f_{k\star}}(v_i)/\sqrt{\mathbf{v}/2}$. Therefore, it is necessary to establish an upper bound for $LS_{f_{k\star}}(v_i)$, denoted as $LS^*_{f_{k\star}}(v_i)$.

We introduce a method for computing a probabilistic upper bound of any value x, given a noisy version of x injected with Laplace noise:

Lemma 1: [15] Let x be any real value, and $x^* = x + Lap(\lambda)$ for some $\lambda > 0$. Then, with $1 - \delta$ probability,

$$x^* + \lambda \cdot \log(\frac{1}{2\delta}) \geq x \tag{6}$$

By Lemma 1, each data owner contribute ϵ_0 information to derive an upper bound of $d(v_i)$ by the following equation:

$$d^*(v_i) = d(v_i) + Lap(\frac{2}{\epsilon_0}) + \frac{2}{\epsilon_0} \cdot \log(\frac{1}{2\delta}) \tag{7}$$

The information contribution is $\epsilon_0 + LS^*_{f_{k\star}}(v_i)/\sqrt{\mathbf{v}/2}$ for each v_i, where $LS^*_{f_{k\star}}(v_i)$ is obtained by combining Equation 5 with Equation 7.

C. k-clique Counting

To begin, we will focus on triangle counting, which is the simplest form of a k-clique (where $k = 3$), and then we will expand our approach to handle k-cliques with $k > 3$.

Triangle Counting. We need to consider the worst case, an edge (v_i, v_j) in the n-node graph can appear in $n-2$ triangles and each triangle is reported three times by its three nodes, when both v_i and v_j are connected to all other nodes in the whole data graph. Therefore, we have $DS_{f_\triangle}(v_i) = 3(n-2)$ derived from the worst-case scenario regardless of the structure of the actual graph G.

It is workable to have a more accurate assertion of information contribution by giving a two-phase method as follows. For each data owner v_i, adding (resp. removing) an edge (v_i, v_j) will increase (resp. decrease) $LS_{f_\triangle}(v_i)$ by $3c(v_i)$ since each triangle is reported by three times, where $c(v_i)$ is the maximum number of common neighbors that she shares with others in her local view, i.e.,

$$c(v_i) = \max_{v_j \in V_i \bigwedge j \neq i} |N(v_i) \cap N(v_j)| \tag{8}$$

where $N(v_i)$ denotes a set of nodes that neighbors node v_i, and V_i is the node set of v_i's ELV G_i. Therefore, the local sensitivity $LS_{f_\triangle}(v_i) = 3c(v_i)$. First, for each owner v_i, we propose to avoid directly collecting $\{c(v_1), ..., c(v_n)\}$ (as it has a high sensitivity with n), but let each owner v_i report an upper bound $d^*(v_i)$ of her degree $d(v_i)$ as adding (resp. removing) an edge (v_i, v_j) only increases (resp. decreases) $d(v_i)$ and $d(v_j)$ by one respectively. The rationale is that $d(v_i) \geq c(v_i)$ holds for any v_i, and hence, we can use a probabilistic upper bound of $d(v_i)$ in place of $c(v_i)$.

To select owners with the top-h' largest node degree, Lemma 1 is exploited to derive the upper bound of the degree. Specifically, for any $d(v_i), 1 \leq i \leq n$, we have:

$$d^*(v_i) = d(v_i) + Lap(\frac{2}{0.5\epsilon_0}) + \frac{2}{0.5\epsilon_0} \cdot \log(\frac{h'+1}{\delta}) \tag{9}$$

where h' is a large integer that $h' < n$ and $0 < \delta < 1$.

This disadvantage, however, is that $d(v_i)$ could be a rather loose upper bound of $c(v_i)$. This motivates us to develop a hybrid approach that combines both $c^*(v_i)$ and $d^*(v_i)$.

Algorithm 2 and 3 show the pseudo-code of the proposed solution, which is divided into two parts: Phase 1 and Phase 2. All nodes participate in the first phase, and only a selected few participate in the second phase. In Phase 1, we determine the set of owners selected in the second phase. Then we measure each participant's information contribution and give the corresponding answer of the query in Phase 2.

Algorithm 2: Phase 1

Input: The data graph G, a parameter for initial information contribution α, invalidation probability δ, a large number h'

Output: the number h of nodes in Phase 2

1 $\epsilon_0 = \frac{\alpha}{\sqrt{\mathbf{v}/2}}$;
2 **for** $i = 1$ **to** n **do**
3 $\quad d^*(v_i) = d(v_i) + Lap(\frac{2}{0.5\epsilon_0}) + \frac{2}{0.5\epsilon_0} \cdot \log(\frac{h'+1}{\delta})$;
4 **end**
5 Sort $\{v_i\}$ into $\{v_{[1]}, ..., v_{[n]}\}$ by $d^*(v_i)$ in descending order;
6 **for** $i = 1$ **to** h' **do**
7 $\quad$ **if** $\frac{i}{0.5\epsilon_0} \cdot \log(\frac{h'+1}{\delta}) \geq d^*(v_{[i+2]})$ **then**
8 $\quad\quad$ **break;**
9 $\quad$ **end**
10 **end**
11 $h = \lceil i/2 \rceil$;
12 $Sed = \{v_{[i]} | 1 \leq i \leq h\}$;

Phase 1: Determine which nodes to select for Phase 2. First, we obtain a probabilistic degree upper bound $d^*(v_i)$ for every owner v_i (lines 2-5), and we identify the set Sed of owners whose degree upper bounds are the largest (lines 6-12). Here, we partition the initial information ϵ_0 into two equal halves, where one portion is allocated for Phase 1 and the other portion is reserved for use in Phase 2. We also divide the probability δ into $2h'+2$ parts, where h' is a specified number indicating the maximum number of nodes to do in Phase 2. After that, we calculate the probabilistic upper bound of the actual degree $d(v_i)$, denoted by $d^*(v_i)$. Then, it uses a heuristic to decide $h \leq h'$, the number of nodes who participate in Phase 2, and obtains the set Sed.

Phase 2: Derive the contribution to the query. Intuitively, for any $v \in Sed$, using $d^*(v_i)$ as an upper bound of $c(v_i)$ is likely to be ineffective, since $c(v)$ could be much smaller than $d^*(v)$. Therefore, for each $v \in Sed$, we derive $c^*(v)$ as an alternative upper bound of $c(v)$, instead of relying solely on $d^*(v)$.

Note that in this case, the amount of owners $v \in Sed$ is $O(|Sed|)$ instead of $O(n)$, since we do not request $c^*(v_i)$ for any $v_i \notin Sed$. Finally, we combine $d^*(v_i)$ and $c^*(v_i)(v_i \in Sed)$ to get an improved upper bound of $LS^*_{f_\triangle}(v_i)$. Each node in Sed calculates $c^*(v_i)$ as their probabilistic upper bound of common neighbor counts, and get their final upper bound $LS^*_{f_\triangle}(v_i)$ (lines 3-4). For other nodes not in Sed, we consider $d^*(v_i)$ to derive their final upper bound $LS^*_{f_\triangle}(v_i)$ (line 6).

Algorithm 3: Phase 2

Input: Query $Q = (f_\triangle, \mathbf{v})$, variance of noise $\mathbf{v}$, initial information contribution ϵ_0, invalidation probability δ, a large number h', ELVs of involved participants $\{G_1, ..., G_n\}$

Output: Contribution $\{\epsilon_1, ..., \epsilon_n\}$

1 **for** $i = 1$ **to** n **do**
2 **if** $v_i \in Sed$ **then**
3 $c^*(v_i) = c(v_i) + Lap(\frac{h}{0.5\epsilon_0}) + \frac{h}{0.5\epsilon_0} \cdot \log(\frac{h'+1}{\delta})$;
4 $LS^*_{f_\triangle}(v_i) = 3\min\{c^*(v_i), d^*(v_i)\}$;
5 **else**
6 $LS^*_{f_\triangle}(v_i) = 3d^*(v_i)$;
7 **end**
8 $\epsilon_i = \epsilon_0 + \frac{LS^*_{f_\triangle(v_i)}}{\sqrt{\mathbf{v}/2}}$;
9 **end**
10 **Return** $\{\epsilon_1, ..., \epsilon_n\}$;

That means:

$$LS^*_{f_\triangle}(v_i) = \begin{cases} 3\min\{c^*(v_i), d^*(v_i)\} & v_i \in Sed \\ 3d^*(v_i) & v_i \notin Sed \end{cases}$$

We can get $\epsilon_i = \epsilon_0 + LS^*_{f_\triangle}(v_i)/\sqrt{\mathbf{v}/2}$, which is the information contribution for v_i (line 8).

In Two-phase, we propose a more appropriate method to estimate the information contribution for each data owner individually and give a corresponding answer to the query given by the buyer.

Extension. The previous subsection discusses an algorithm to obtain an estimation of information contribution when giving a triangle counting query. In this subsection, we extend the algorithm to estimate each data owner's information contribution when answering $Q = (f_{k\mathbb{C}}, \mathbf{v})$, where $f_{k\mathbb{C}}$ is a k-clique counting query. A k-clique refers to a set of k nodes that are fully connected to each other. Note that triangle counting is a specific form of k-clique counting where $k = 3$.

The information contribution of each answer $f_{k\mathbb{C}}(G_i)$ injecting Laplace noise $Lap(\sqrt{\mathbf{v}/2})$ is bounded by $\epsilon_i \leq k\binom{n-2}{k-2}/\sqrt{\mathbf{v}/2}$, since the global sensitivity $DS_{f_{k\mathbb{C}}}(v_i) = k\binom{n-2}{k-2}$. This is because (1) adding or removing one edge e in G_i affects only those k-cliques where e is an edge, (2) there are at most $\binom{n-2}{k-2}$ such k-cliques, and (3) each k-clique is reported k times.

We apply an improved method to obtain a more accurate estimate of information contribution for counting k-clique. For this purpose, we consider an algorithm for computing a probabilistic upper bound of $LS_{f_{k\mathbb{C}}}(v_i)$ to release information no more than ϵ_0. First, we have:

$$LS_{f_{k\mathbb{C}}}(v_i) = \max_{v_j \in G_i, j \neq i} k \cdot \mathbb{C}(G_{i\cap j}, k-2) \quad (10)$$

where $G_{i\cap j}$ denotes the subgraph of G_i induced the common neighbors of v_i and v_j, and $\mathbb{C}(G_{i\cap j}, k-2)$ denotes the number of $(k-2)$-cliques in $G_{i\cap j}$. To explain, observe that if a k-clique is affected by the presence or absence of an edge (v_i, v_j), then (1) the k-clique must contain both v_i and v_j, and (2) apart from v_i and v_j, the remaining $k-2$ nodes in the clique must form a $(k-2)$-clique. There exists only $\mathbb{C}(G_{i\cap j}, k-2)$ such cliques, and each of them is reported by k data owners.

By Lemma 1, then we can compute $LS^*_{f_{k\mathbb{C}}}(v_i)$ as:

$$LS^*_{f_{k\mathbb{C}}}(v_i) = LS_{f_{k\mathbb{C}}}(v_i) + Lap(\lambda_0) + \lambda_0 \cdot \log(\frac{1}{2\delta_0}) \quad (11)$$

where $\lambda_0 = kn(\binom{n-2}{k-2} - \binom{n-3}{k-2})/\epsilon_0$, since adding or removing one edge in G_i may change $LS_{f_{k\mathbb{C}}}(v_i)$ by up to $kn(\binom{n-2}{k-2} - \binom{n-3}{k-2})$. By getting $LS^*_{f_{k\mathbb{C}}}(v_i)$ as an upper bound of the local sensitivity, we can calculate the information contribution $\epsilon_i = \epsilon_0 + LS^*_{f_{k\mathbb{C}}}(v_i)/\sqrt{\mathbf{v}/2}$.

Since adding or removing one edge in G_i may change each $LS_{f_{k\mathbb{C}}}(v_i)$ by up to $k(\binom{n-2}{k-2} - \binom{n-3}{k-2})$. This leads to an enormous amount of noise in $LS^*_{f_{k\mathbb{C}}}(v_i)$.

To address this issue, we apply a similar algorithm of Two-phase for counting triangles to derive an alternative upper bound of $LS^*_{f_{k\mathbb{C}}}(v_i)$.

Following Equation 10, because $\mathbb{C}(G_{i\cap j}, k-2) \leq \binom{c(v_i)}{k-2}$, if we are able to derive an upper bound of $c(v_i)$, then we can use $k\binom{c^*(v_i)}{k-2}$ as an upper bound of $LS_{f_{k\mathbb{C}}}(v_i)$. We compute such an upper bound $c^*(v_i)$ using the same method described in triangle counting. Regarding $k\binom{c^*(v_i)}{k-2}$ as an upper bound of $LS_{f_{k\mathbb{C}}}(v_i)$, the information contribution estimated is $\epsilon_i = \epsilon_0 + k\binom{c^*(v_i)}{k-2}/\sqrt{\mathbf{v}/2}$.

V. Arbitrage-free for Pricing Function

We first introduce a fundamental and desirable property of pricing functions, namely arbitrage-free. We must ensure the pricing function $\pi(Q)$ is arbitrage-free to avoid the buyer attempting to purchase other combinations of queries to answer $Q = (f, \mathbf{v})$ with a cheaper price. To achieve this objective, it should be arbitrage-free not only to the graph statistic query but also to the variance of noise $\mathbf{v}$. The proposed $\pi(Q)$ is a monotonically decreasing function with respect to $\mathbf{v}$ and cannot decrease faster than $1/\mathbf{v}$. Moreover, $\pi(Q)$ is only considered arbitrage-free if it satisfies the semi-norm for f.

Before investigating arbitrage-free, we first establish the key concept of the determinacy relation for computing $Q(G)$. A similar concept has been studied in randomized query/view answering from the database community [19].

We give the formal definition of query determinacy as follows.

Definition 5 (Determinacy): The determinacy relation is a *relation* between a query $Q = (f, \mathbf{v})$ and a multi-set of queries $S = \{Q_1, ..., Q_k\}$, denoted $S \to Q$, and defined by the following rules:

1) *Summation*: $\{(f_1, \mathbf{v}_1), ..., (f_k, \mathbf{v}_k)\} \to (f_1 + ... + f_k, \mathbf{v}_1 + ..., +\mathbf{v}_k)$;
2) *Scalar multiplication*: $\forall c \in \mathbb{R}$, $(f, \mathbf{v}) \to (cf, c^2\mathbf{v})$;
3) *Relaxation*: $(f, \mathbf{v}) \to (f, \mathbf{v}')$, where $\mathbf{v} \leq \mathbf{v}'$;
4) *Transitivity*: If $S_1 \to Q_1, ..., S_k \to Q_k$ and $\{Q_1, ..., Q_k\} \to Q$, then $\bigcup_{i=1}^k S_k \to Q$.

Based on the query determinacy relation, we formally define arbitrage-free.

Definition 6: A pricing function $\pi(Q)$ is arbitrage-free if $\forall i \geq k$, $\{Q_1, ..., Q_k\} \rightarrow Q$ implies:

$$\pi(Q) \leq \sum_{i=1}^{k} \pi(Q_i) \tag{12}$$

The intuition behind the above definition is that if there exists arbitrage in the pricing function $\pi(\cdot)$, e.g., $\pi(Q) > \sum_{i=1}^{k} \pi(Q_i)$, then the data buyer would never pay the full price of the query Q. Instead, he would turn to buy a cheaper set of queries $\{Q_1, ..., Q_k\}$.

We divide an arbitrage-free pricing function into two parts, namely the variance of noise $\mathbf{v}$ and the query f, and conquer each part step by step.

First, we consider the constraint of the variance $\mathbf{v}$ of noise. By Definition 5(3) we know that π is monotonically decreasing in $\mathbf{v}$. The next lemma shows that it cannot decrease faster than $1/\mathbf{v}$.

Lemma 2: For any arbitrage-free pricing function $\pi(f, \mathbf{v})$ that depends on two independent parts f and $\mathbf{v}$, it can not decrease faster than $1/\mathbf{v}$.

Proof. Suppose the contrary: there exists a query f and a sequence of variance $\{\mathbf{v}_j | j \in \{1, ..., +\infty\}\}$ such that $\lim_{j\to\infty} \mathbf{v}_j = +\infty$ and $\lim_{j\to\infty} \mathbf{v}_j \pi(f, \mathbf{v}) = 0$. Select $j_0 > 1$ such that $\mathbf{v}_{j_0} > 1$ and $\mathbf{v}_{j_0}\pi(f, \mathbf{v}_{j_0}) < \pi(f, 1)/2$. Then we can answer the query $(f, 1)$ by requesting $\lceil \mathbf{v}_{j_0} \rceil$ times the same query $(f, \mathbf{v}_{j_0})$ and average their answers. For these $\lceil \mathbf{v}_{j_0} \rceil$ queries, we pay:

$$\lceil \mathbf{v}_{j_0} \rceil \pi(f, \mathbf{v}_{j_0}) \leq (\mathbf{v}_{j_0}+1)\pi(f, \mathbf{v}_{j_0}) < 2\mathbf{v}_{j_0}\pi(f, \mathbf{v}_{j_0}) < \pi(f, 1)$$

which implies that we have arbitrage, a contradiction.

We continue to consider the other part of an arbitrage-free pricing function $\pi(f, \mathbf{v})$, namely the query f. For that, we assume that π is inversely proportional to $\mathbf{v}$. In other words, π decreases at a rate $1/\mathbf{v}$, which is the fastest rate allowed by Lemma 2. Set $\pi(f, \mathbf{v}) = h^2(f)/\mathbf{v}$, for some positive function h that depends only on f. It can be shown that π is arbitrage-free iff h is a semi-norm in [12], [20]. Recall that a semi-norm is a function h that satisfies the following properties.

- *Homogeneity:* For any $c \in \mathbb{R}$ and any graph statistic query f, $h(cf) = |c|h(f)$.
- *Subadditivity:* For any query f_1 and f_2, $h(f_1 + f_2) \leq h(f_1) + h(f_2)$.

Furthermore, we utilize semi-norm to design our basic arbitrage-free pricing function:

Theorem 2 (Basic Arbitrage-free Pricing function): Let $\pi(f, \mathbf{v}) = h(f))^2/\mathbf{v}$ be the pricing function for some positive function $h(f)$ that only depends on f. Then, $\pi(f, \mathbf{v})$ is arbitrage-free iff $h(f)$ is a semi-norm.

We next consider how to construct more arbitrage-free pricing functions by combining basic/existing ones. We resort to a general class of non-decreasing and subadditive functions. We recall that a function $\Gamma : \mathbb{R}^\phi \rightarrow \mathbb{R}$ over $\forall \mathbf{y}, \mathbf{z} \in \mathbb{R}^\phi$ is non-decreasing, if $\mathbf{y} \leq \mathbf{z}$, $\Gamma(\mathbf{y}) \leq \Gamma(\mathbf{z})$. Besides, it is subadditive, if $\Gamma(\mathbf{y} + \mathbf{z}) \leq \Gamma(\mathbf{y}) + \Gamma(\mathbf{z})$.

Theorem 3 (Composite Arbitrage-free Pricing Functions): Let $\Gamma : \mathbb{R}^\phi \rightarrow \mathbb{R}$ be a non-decreasing and subadditive function. For any set of arbitrage-free pricing functions $\{\pi_1(Q), ..., \pi_\phi(Q)\}$, the composite pricing function $\pi(Q) = \Gamma(\pi_1(Q), ..., \pi_\phi(Q))$ is also arbitrage-free.

We give some typical examples of composite arbitrage-free pricing functions as follows. If $\pi_1(Q), ..., \pi_\phi(Q)$ are arbitrage-free, then

- *Linear Combination:* $\forall c_1, ..., c_\phi \geq 0$, $\sum_{j=1}^{\phi} c_j \pi_j(Q)$;
- *Geometric Mean:* $\sqrt{\Pi_{j=1}^{\phi} \pi_k(Q)}$;
- *Maximum:* $\max(\pi_1(Q), ..., \pi_\phi(Q))$;
- *Power:* $\pi(Q)^c$ for $0 \leq c \leq 1$;
- *Logarithmic:* $\log(\pi(Q) + 1)$;
- *Cut-off:* $\min(\pi(Q), c)$ for $c \geq 0$;
- *Sigmoid:* $\tanh(\pi(Q))$, $\arctan(\pi(Q))$, $\frac{\pi(Q)}{\sqrt{\pi(Q)^2+1}}$.

are arbitrage-free as well.

We note that given basic arbitrage-free pricing functions, the first five composite arbitrage-free pricing functions set an infinite price for the unperturbed answer, i.e., the variance of noise $\mathbf{v} = 0$. However, these functions are impractical in the data market, since the data owner v_i tends to sell the unperturbed answer for price p_i, which is a high but finite price. Nerveless, we can turn to apply some bounding functions for composition, e.g., cut-off and sigmoid functions.

Based on these proprieties of arbitrage-free pricing functions, we can prove that $\pi(Q)$ is arbitrage-free. Observing that $\pi(Q) = c\sum_{i=1}^{n} \mu_i(Q)$ is a linear combination of $\mu_i(Q)$ for each data owner v_i, it suffices to prove that $\mu_i(Q)$ is arbitrage-free by Theorem 3 (Linear combination). It further suffices to prove the arbitrage-free of ϵ_i by Theorem 3 (Linear combination and Sigmoid). Analogous to Theorem 3 (Geometric Mean and Linear combination), we can construct $\psi_i(Q) = (DS_f(v_i))^2/\mathbf{v}$. Then it can be proved that $\psi_i(Q)$ is arbitrage-free as follows.

Theorem 4: $\psi_i(Q)$ is arbitrage-free.

Proof. First, we can check that $\psi_i(Q)$ decreases as $1/\mathbf{v}$, which satisfies an arbitrage-free function for the variance $\mathbf{v}$ in Lemma 2. Then we set $\psi_i(Q) = h_i^2(f)/\mathbf{v}$ where $h_i(f) = DS_f(v_i)$, it suffices to prove that $h_i(f)$ is semi-norm by Theorem 2.

We will prove that $h_i(f) = DS_f(v_i)$ is semi-norm. Observing that $h_i(f)$ satisfies the homogeneity property, we will prove $h_i(f) = DS_f(v_i)$ satisfies the subadditive property. We stipulate that $E_x \cap E_y = \emptyset$ and $E_x \cup E_y = E$, where E_x, E_y and E denotes the edge set of G_x, G_y and G respectively. It means that G_x and G_y do not have common edges and they can construct the whole graph G. Then, we have $f_1(G) = f(G_x)$, $f_2(G) = f(G_y)$ and $f_3(G) = f(G) = f_1(G) + f_2(G)$.

Because $DS_{f_3}(v_i) \leq DS_{f_1}(v_i) + DS_{f_2}(v_i)$, it is obvious that $h_i(f_3) \leq h_i(f_1) + h_i(f_2)$, which completes our proof.

TABLE I: Dataset Statistics

Dataset	# nodes	# edges	Avg. degree	# 3-star	# triangle
Facebook	4,039	88,234	43.69	9,314,849	4,836,030
HepPh	12,008	118,521	19.74	15,280,441	10,075,497
DBLP	317,080	1,049,866	6.62	21,780,889	6,673,155
Email	36,692	183,831	10.02	25,566,893	2,181,132

VI. Experiment

The goal of the experimental section is three-fold: (1) validate that the answer's accuracy is monotone with respect to the information contribution. (2) show that the framework can generate diverse prices for buyers, while data owners can receive appropriate compensations according to their information contribution. (3) demonstrate that our methods exhibits significantly better performance than some baselines while satisfying the online query requirements.

A. Experimental Settings

Datasets. We used four different sets of publicly available real-world datasets from *Stanford Large Network Dataset Collection* [21]: a social network *Facebook* [22], a citation network *HepPh* [23], a collaboration network *DBLP* [24] and a communication network *Email* [25]. All graphs are converted into undirected graphs by ignoring edge directions. Summary statistics of datasets are provided in Table I.

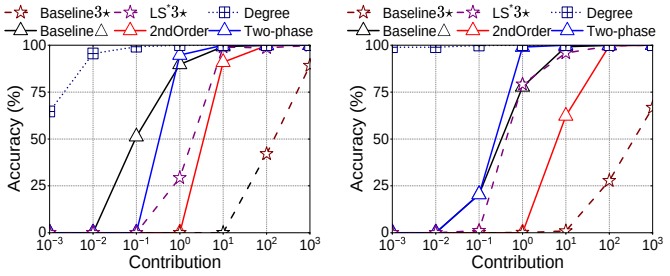
Algorithms. To justify the performance of our pricing method, we compare it with the state-of-art methods: (1) Baseline [16], a noisy pricing framework for quantifying information by using global sensitivity for the subgraph counting. (2) 2nd Order [6], an information contribution quantification framework by giving the second-order local sensitivity with a simple upper bound for k-clique counting. We implemented the proposed algorithms: Degree, LS'k-star and Two-phase for calculating the information contribution and answers of node degrees, k-stars and k-cliques respectively.

All experiments were run on a machine with an Intel(R) Core(TM) i7-9700 CPU at 3.00GHz with 16GB RAM. Each experiment was run 10 times and the average result is reported.

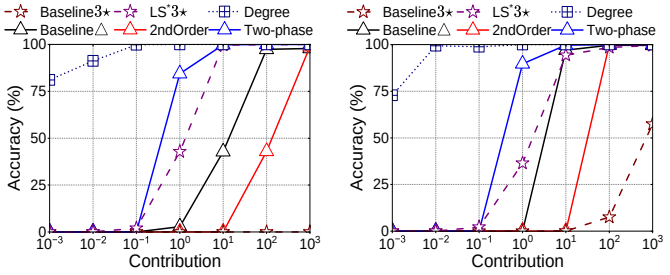
B. Experimental Results

Exp-1: Accuracy influenced by information contribution. Before investigating economic properties, we first evaluate how accuracy changes when the information contribution ϵ increases. Here, we define $\epsilon = \max_i \epsilon_i$. We changed the feature ϵ varies from 10^{-3} to 10^3 by exponential growth. We conducted an experiment utilizing all the mentioned methods to calculate the total node degrees, 3-star counts and triangle counts. The accuracy was reported as $1 - \frac{|f(G)-Q(G)|}{|f(G)+Q(G)|}$, where $f(G)$ represents the unperturbed results and $Q(G)$ represents the perturbed results. The experimental results are illustrated in Figure 4.

We observe that as the value of ϵ increases, the accuracy of the results improves, often exhibiting rapid growth within certain intervals. To illustrate this, we present Figure 4(a), where Two-phase achieves an accuracy of approximately 94.8% at $\epsilon = 1$, while the accuracy drops to nearly zero at



(a) Accuracy (*Facebook*) (b) Accuracy (*HepPh*)



(c) Accuracy (*DBLP*) (d) Accuracy (*Email*)

Fig. 4: Accuracy influenced by the information contribution.

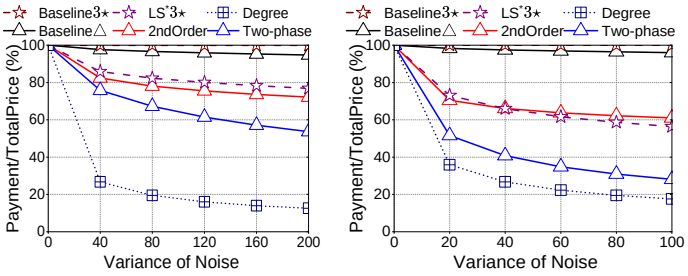
$\epsilon = 0.1$. We provide an explanation for this phenomenon based on the relationship between the variance of Laplace noise ($\mathbf{v}$) and ϵ:

$$\epsilon = \frac{\max_i DS_f(v_i)}{\sqrt{\mathbf{v}/2}} \Rightarrow \mathbf{v} = 2(\frac{\max_i DS_f(v_i)}{\epsilon})^2 \qquad (13)$$

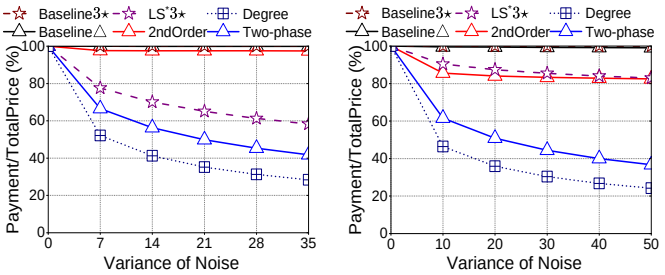
which follows from Theorem 1. As the information contribution ϵ increases, the amount of noise added decreases. Consequently, this reduction in noise results in more accurate answers. Furthermore, when ϵ is too small or too large, the perturbation or the true result completely dominates. Additionally, we compare the accuracy of our three methods using the same contribution settings. Degree demonstrates the highest accuracy, followed by Two-phase, and LS'3⋆. This discrepancy can be attributed to the lower sensitivity of degree counting compared to 3-star and triangle counting. The results demonstrate that all methods examined in our analysis meet the fundamental pricing principle, namely, that higher accuracy of the answer can be achieved when owners provide more information.

Exp-2: Impact of variance on payment. To evaluate the impact of variance, we varied the variance from d_0 to $5d_0$, where d_0 is the average degree in a dataset. As shown in Figure 5, comparing the ratio of the payment $\pi(Q)$ and the total price (i.e., $\sum_{i=1}^{n} p_i$), buyers can expect to pay less as the noise added to data increases. This provides the advantage of reducing the cost of accommodating diverse requirements. Conversely, the data market is able to offer different levels of accuracy to buyers based on their affordability. We explain the reason through the arbitrage-free query pricing function:

$$\pi(Q) = \sum_{i=1}^{n} \frac{2p_i}{\Pi} \arctan(b_i \epsilon_i) \qquad (14)$$

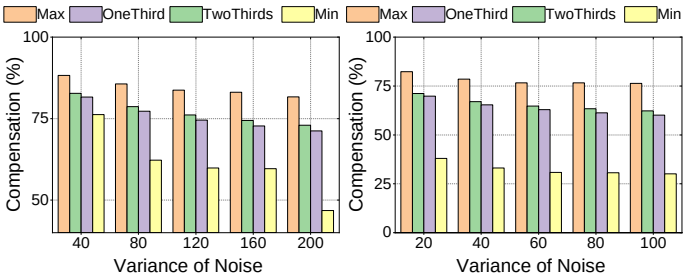


(a) Price (*Facebook*) (b) Price (*HepPh*)

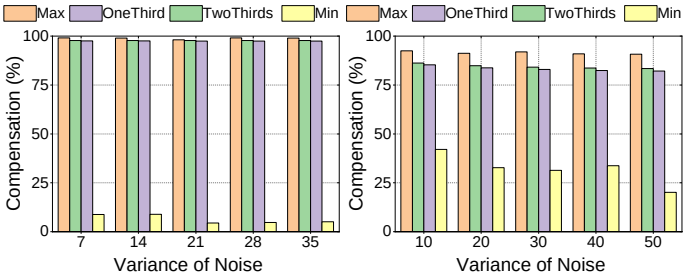


(c) Price (*DBLP*) (d) Price (*Email*)

Fig. 5: Impact of variance on payment.



(a) *Facebook* (b) *HepPh*

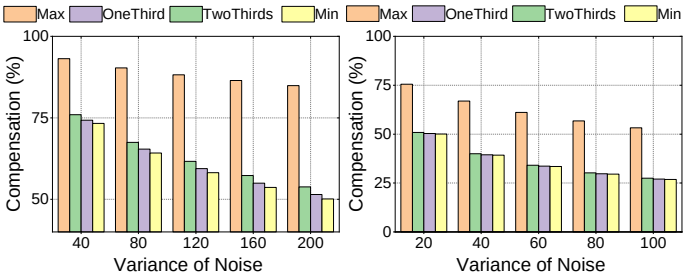


(c) *DBLP* (d) *Email*

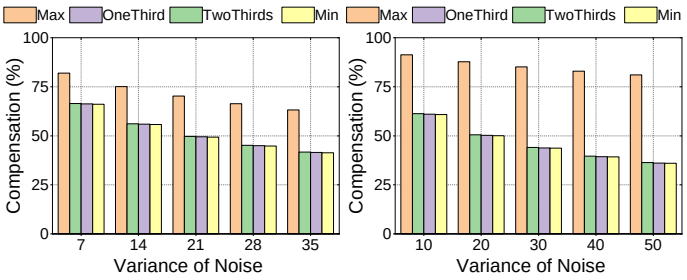
Fig. 6: Impact of variance on compensations (2nd Order).

When the noise $\mathbf{v}$ becomes larger, the information contribution ϵ_i becomes smaller. Consequently, the payment also decreases due to the monotonically increasing nature of the $\arctan(\cdot)$ function. Furthermore, we compare the payments of different methods. Specifically, we find that $LS^*_{f_\triangle}(v_i)$ (Two-phase) $< LS^*_{f_\triangle}(v_i)$ (2nd Order) $< DS_{f_\triangle}(v_i)$ (Baseline△). The payment of Two-phase experiences a significant decrease with an increase in variance, followed by 2nd Order, and then Baseline△. Similarly, the payment of LS'3⋆ decreases at a faster rate compared to Baseline3⋆ due to $LS^*_{f_{3\star}}(v_i)$ (LS'3⋆) $< DS_{f_{3\star}}(v_i)$ (Baseline3⋆). For instance, as shown in Figure 5(d) using the *Email* dataset, when $\mathbf{v} = 50$, the buyer only needs to pay 36.7% of the total price in Two-phase. In contrast, the buyer needs to pay 82.4% and 99.1% in 2nd Order and Baseline△, respectively. These results highlight that, compared to baselines, LS'3⋆ and Two-phase offer a greater degree of flexibility. Specifically, this approach enables the market to cater to a wider range of buyers with varying needs and budgets. These findings demonstrate the potential benefits of adopting LS'3⋆ and Two-phase in enhancing market efficiency and improving buyer satisfaction. Another observation is that, when comparing three different graph statistic queries, the payment of Degree decreases with an increase in variance, followed by Two-phase, and then LS'3⋆. Overall, our experimental results provide insights into the dynamics of payments concerning noise variance and information contribution across various methods.

Exp-3: Compensations for data owners. Empirical evidence suggests that real-world graphs conform to a Power-Law degree distribution, indicating that the majority of information is contributed by a small subset of nodes, who deserve a higher percentage of the compensations. Also, as the increase



(a) *Facebook* (b) *HepPh*



(c) *DBLP* (d) *Email*

Fig. 7: Impact of variance on compensations (Two-phase).

of the amount of noise is added to the answer, the corresponding contribution of the data owner diminishes, resulting in a commensurate reduction in the associated compensation. We compared the compensation for data owners of three methods. Specifically, we evaluate the compensation for the most, ranked one-third, two-thirds, and the least impacted by noise. Here, the global sensitivity $DS_{f_\triangle}(v_i)$ of Baseline△ is the same for each owner in the certain variance, Therefore, the compensation is the total divided by the number of

owners. Thus, we only consider the trend of the compensation change of 2nd Order and Two-phase shown in Figure 6 and Figure 7. Both methods show that the data market offers each data owner individualized compensation, depending on their upper bound of the local sensitivity $LS^*_{f_\triangle}(v_i)$, and their compensation decreases with the increment of noise $\mathbf{v}$. As illustrated in Figures 6(d) and 7(d), for the *Email* dataset, 2nd Order and Two-phase exhibit maximum compensation to unperturbed payment ratios of 90.7% and 81.1%, respectively, when $\mathbf{v} = 50$. In contrast, at the same $\mathbf{v}$, 2nd Order and Two-phase exhibit one-third ratios of 83.4% and 36.4%, respectively.

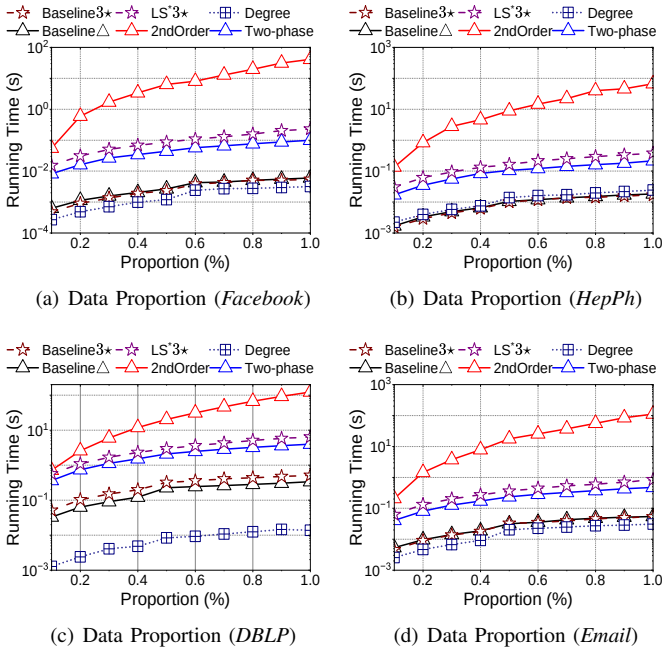
The results imply that Two-phase exhibits a steeper decrease in compensation for data owners who contribute less information than 2nd Order. The results show that Two-phase offers a more equitable approach to payment allocation than 2nd Order due to its reduced compensation for data owners who contribute less information, resulting in a lower price. The previous payment experiment indicates that Two-phase results in a significantly lower price compared to 2nd Order. Furthermore, the experiment reveals that although the compensation received by the primary contributor varies only slightly between Two-phase and 2nd Order, the compensation for data owners who provide less information is substantially lower in Two-phase compared to 2nd Order.

Another observation is that the compensation gaps between data owners are quite different in these two methods. In 2nd Order, the difference between the maximum and one-third/two-thirds ratios of the compensation obtained by the data owner is always maintained at about ten percent, which means that the gap does not widen for owners whose degrees are large. But the circumstance is completely different in Two-phase, since $LS^*_{f_\triangle}(v_i)$ of different data owners varies widely, their compensation ratios also vary greatly as a result, which provides a more reasonable query pricing strategy. For instance, shown in Figure 7(d), on *Email*, though $\mathbf{v} = 50$, the most significant contributor still receives 81.1% compensation compared to the price for an unperturbed answer. By contrast, there are 36.4%, 36.1%, and 35.9% for the one-third ranking, two-thirds ranking and the least compensation shown in Figure 7(d), receptively.

TABLE II: Time complexity and running time per owner (ms)

	Degree	Baseline3⋆	LS⁺3⋆	Baseline3△	2ndOrder	Two-phase
Time complexity	$O(n)$	$O(n)$	$O(n^2)$	$O(n)$	$O(n^3)$	$O(n^2 + h'n^2)$
Facebook	0.00079	0.0014	0.057	0.00060	11.065	0.025
HepPh	0.0020	0.0014	0.032	0.00062	5.403	0.016
DBLP	0.000043	0.0016	0.020	0.0011	0.388	0.012
Email	0.00084	0.0014	0.023	0.00098	3.107	0.013

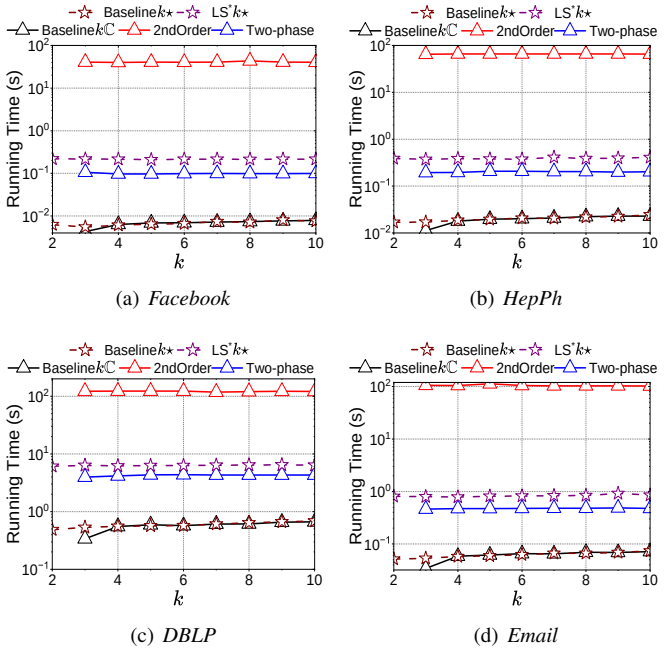
Exp-4: Performance of Algorithms. To test the performance of these algorithms, we present the time complexity of all mentioned methods in Table II. To further validate our analysis, we measured the running time of these algorithms and calculated the average time cost per data owner, as presented in Table II. Additionally, the total time cost for each type of graph statistics is illustrated in Figure 8. The running time of all methods grows with the increase of owner number. Specially,



(a) Data Proportion (*Facebook*) (b) Data Proportion (*HepPh*)

(c) Data Proportion (*DBLP*) (d) Data Proportion (*Email*)

Fig. 8: Impact of graph size on running time.



(a) *Facebook* (b) *HepPh*

(c) *DBLP* (d) *Email*

Fig. 9: Impact of k on running time.

Baseline has an advantage over other methods in terms of running time, even though it outputs a relatively less accurate information contribution estimation. Two-phase runs much faster than 2nd Order because it only needs to calculate $c^*(v_i)$ for $v_i \in Sed$ and 2nd Order needs to calculate for all v_i where $1 \leq i \leq n$. Despite being slower than their corresponding Baseline, LS⁺3⋆ and Two-phase are capable of meeting the online requirements. As noted in [26], the response time for online services should be less than 6.7 seconds.

Finally, we examined the impact of k in Figure 9. The results show the performance of each method of calculating information contribution with respect to k in these four datasets. We observe that all methods' execution time remains virtually unchanged as the k increases. As illustrated in Section IV-B and IV-C, it can be seen that these methods for k-star (resp. k-clique) have similar processes with the 3-star (resp.triangle). We also note that Two-phase outperforms 2nd Order in terms of running time across all datasets. Although LS'3$\star$ and Two-phase exhibit slower performance than their Baseline, they still meet the online requirements.

VII. Related Work

We categorize the related work as follows.

Data market design. Data market design has gained increasing attention in recent years, especially from the database community. The pricing research in this field mainly focuses on query-based pricing [10], showing that the prices of a large class of SQL queries can be computed using ILP solvers. Based on the theoretical framework, a new pricing system, called QueryMarket [27], is developed for flexible query pricing, which can leverage query history to avoid double charging when queries purchased over time have overlapping information. More arbitrage-free pricing functions are designed for arbitrary query formats in [19]. The structure of arbitrage-free functions is characterized in both answer-dependent and instance-independent settings. Based on the above work, a novel pricing system, called QIRANA [28], is implemented to perform query-based data pricing for a large class of SQL queries (including aggregation) in real-time. Specific to private data, a classical framework is proposed in [12] to price linear queries by introducing arbitrage-free. However, subgraph counting is not a simple task as linear queries. An aggregate statistics pricing framework over private correlated data (ERATO) [16] takes data correlation into account, and further considers servicing pricing and privacy compensation in practical aggregate statistics. However, ERATO can only provide pricing for aggregate statistics. When it comes to graph statistics, it cannot adequately compensate data owners due to the difficulty in quantifying the interdependence of each owner's data.

Differential privacy over social graphs. A number of algorithms have been developed for analyzing graphs under differential privacy, classified into the centralized setting and decentralized setting. Due to the trustfulness of data holders, the major differences among the existing algorithms are the perturbation mechanism and statistics aggregation. In the first category, the global structure of a social graph is managed by a trusted party, who releases subgraph counts [6], [29], degree distribution [30], synthetic graphs [31], [32] and many other statistics [33], [34] on graphs under differential privacy guarantees. In this scenario, it can be further classified into two types [35]: Edge-DP [36], [37], [38] and Node-DP [39], [40], [41]. Edge-DP considers two neighboring graphs that differ in the only edge. By contrast, Node-DP considers two neighboring graphs that differ in one node and all its incident edges, which provides a stricter privacy guarantee. In the second category, an untrusted party needs to communicate with individual participants of the network, each of which has a limited local view of the whole social graph, and then combines information from different participants to estimate the global network properties. Typically, only Edge-DP is considered since the data aggregators are usually aware of the identities of participated clients. Decentralized Different Privacy is proposed in [15] to require each client to protect not only her own privacy but also the privacy of her neighbors. Recently, An enhanced privacy notion named edge relational local differential privacy is proposed in [42] and it ensures the probability of revealing the edge presence by one report bounded by the privacy budget given the observations of all other reports.

Privacy preserving subgraph counts. There have been a large number of algorithms developed for counting subgraphs thus far and these algorithms can generally be classified into random response and adding random noise. The first category of algorithms started with [43] and was followed by [17], [44]. An approach of collecting a neighbor list perturbed by the Random Response from each client and calibrating the triangle counts for a less biased estimation is proposed under local differential privacy [43] . In the second category, the LS-Based algorithm adds Cauchy noise with an expected magnitude proportional to the smooth sensitivity under centralized differential privacy [18]. By contrast, a two rounds collection algorithm first asks each noise to report the minimum scale necessary for injecting Laplace noise in the whole network and collects subgraph counts accordingly under decentralized differential privacy [15], [42], [17]. Except for the Laplace noise, quantifying the privacy loss caused by a query with a specific variance of noise using other methods is challenging due to their complex loss function compression [17]. Hence, in this paper, we limit the noise to follow a Laplace distribution.

VIII. Conclusion

With the development of data exchange today, data markets have emerged as intermediaries to facilitate this process. At the same time, there is a growing concern regarding graph analysis tasks in various applications. This paper focuses on developing a framework for pricing noisy graph statistics. Specifically, we introduce the concept of extended local views (ELVs) for data owners to publish their data. By accurately quantifying the relationship between noise and payment, we ensure a fair and reasonable pricing mechanism for both buyers and data owners. To measure the information contribution of data owners, we propose algorithms specifically designed for fundamental graph statistics, including node degrees and subgraph counts such as k-stars and k-cliques. Furthermore, we provide formal proofs demonstrating that the proposed pricing framework is arbitrage-free. In addition to theoretical analysis, we conduct comprehensive experiments to validate the effectiveness of our algorithms. Through these experiments, we demonstrate the practical viability of our framework in real-world scenarios.

REFERENCES

[1] J. M. Nolin, "Data as oil, infrastructure or asset? three metaphors of data as economic value," *Journal of Information, Communication and Ethics in Society*, vol. 18, no. 1, pp. 28–43, 2020.

[2] K. Schwab, A. Marcus, J. Oyola, W. Hoffman, and M. Luzi, "Personal data: The emergence of a new asset class," in *An Initiative of the World Economic Forum*. World Economic Forum Cologny, Switzerland, 2011.

[3] J.-P. Onnela, J. Saramäki, J. Kertész, and K. Kaski, "Intensity and coherence of motifs in weighted complex networks," *Physical Review E*, vol. 71, no. 6, p. 065103, 2005.

[4] A. Mrzic, P. Meysman, W. Bittremieux, P. Moris, B. Cule, B. Goethals, and K. Laukens, "Grasping frequent subgraph mining for bioinformatics applications," *BioData mining*, vol. 11, no. 1, pp. 1–24, 2018.

[5] M. Jia, M. Van Alboom, L. Goubert, P. Bracke, B. Gabrys, and K. Musial, "Encoding edge type information in graphlets," *Plos one*, vol. 17, no. 8, p. e0273609, 2022.

[6] V. Karwa, S. Raskhodnikova, A. Smith, and G. Yaroslavtsev, "Private analysis of graph structure," *Proceedings of the VLDB Endowment*, vol. 4, no. 11, pp. 1146–1157, 2011.

[7] B.-R. Lin and D. Kifer, "On arbitrage-free pricing for general data queries," *Proceedings of the VLDB Endowment*, vol. 7, no. 9, pp. 757–768, 2014.

[8] http://www.bigdataexchange.com/.

[9] http://www.qlik.com/us/products/qlik-data-market.

[10] P. Koutris, P. Upadhyaya, M. Balazinska, B. Howe, and D. Suciu, "Query-based data pricing," *Journal of the ACM (JACM)*, vol. 62, no. 5, pp. 1–44, 2015.

[11] C. Chen, Y. Yuan, Z. Went, G. Wang, and A. Li, "Gqp: A framework for scalable and effective graph query-based pricing," in *2022 IEEE 38th International Conference on Data Engineering (ICDE)*. IEEE, 2022, pp. 1573–1585.

[12] C. Li, D. Y. Li, G. Miklau, and D. Suciu, "A theory of pricing private data," *ACM Transactions on Database Systems (TODS)*, vol. 39, no. 4, pp. 1–28, 2014.

[13] J. G. Saw, M. C. Yang, and T. C. Mo, "Chebyshev inequality with estimated mean and variance," *The American Statistician*, vol. 38, no. 2, pp. 130–132, 1984.

[14] S. A. Ross, "The arbitrage theory of capital asset pricing," in *Handbook of the fundamentals of financial decision making: Part I*. World Scientific, 2013, pp. 11–30.

[15] H. Sun, X. Xiao, I. Khalil, Y. Yang, Z. Qin, H. Wang, and T. Yu, "Analyzing subgraph statistics from extended local views with decentralized differential privacy," in *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, 2019, pp. 703–717.

[16] C. Niu, Z. Zheng, F. Wu, S. Tang, X. Gao, and G. Chen, "Erato: trading noisy aggregate statistics over private correlated data," *IEEE Transactions on Knowledge and Data Engineering*, vol. 33, no. 3, pp. 975–990, 2019.

[17] J. Imola, T. Murakami, and K. Chaudhuri, "Locally differentially private analysis of graph statistics." in *USENIX Security Symposium*, 2021, pp. 983–1000.

[18] K. Nissim, S. Raskhodnikova, and A. Smith, "Smooth sensitivity and sampling in private data analysis," in *Proceedings of the thirty-ninth annual ACM symposium on Theory of computing*, 2007, pp. 75–84.

[19] S. Deep and P. Koutris, "The design of arbitrage-free data pricing schemes," *arXiv preprint arXiv:1606.09376*, 2016.

[20] L. Chen, P. Koutris, and A. Kumar, "Towards model-based pricing for machine learning in a data marketplace," in *Proceedings of the 2019 International Conference on Management of Data*, 2019, pp. 1535–1552.

[21] J. Leskovec and A. Krevl, "SNAP Datasets: Stanford large network dataset collection," http://snap.stanford.edu/data, Jun. 2014.

[22] J. McAuley and J. Leskovec, "Facebook," https://snap.stanford.edu/data/ego-Facebook.html, 2012.

[23] J. Leskovec, J. Kleinberg, and C. Faloutsos, "Hepph," https://snap.stanford.edu/data/ca-AstroPh.html, 2007.

[24] J. L. Jaewon Yang, "Dblp," https://snap.stanford.edu/data/com-DBLP.html, 2012.

[25] W. Cohen, "Email," https://snap.stanford.edu/data/ca-AstroPh.html, 2004.

[26] Browserstack, "How fast should a website load in 2023?" https://www.browserstack.com/guide/how-fast-should-a-website-load, 2023.

[27] P. Koutris, P. Upadhyaya, M. Balazinska, B. Howe, and D. Suciu, "Toward practical query pricing with querymarket," in *proceedings of the 2013 ACM SIGMOD international conference on management of data*, 2013, pp. 613–624.

[28] S. Deep and P. Koutris, "Qirana: A framework for scalable query pricing," in *Proceedings of the 2017 ACM International Conference on Management of Data*, 2017, pp. 699–713.

[29] J. Zhang, G. Cormode, C. M. Procopiuc, D. Srivastava, and X. Xiao, "Private release of graph statistics using ladder functions," in *Proceedings of the 2015 ACM SIGMOD international conference on management of data*, 2015, pp. 731–745.

[30] M. Hay, C. Li, G. Miklau, and D. Jensen, "Accurate estimation of the degree distribution of private networks," in *2009 Ninth IEEE International Conference on Data Mining*. IEEE, 2009, pp. 169–178.

[31] A. Sala, X. Zhao, C. Wilson, H. Zheng, and B. Y. Zhao, "Sharing graphs using differentially private graph models," in *Proceedings of the 2011 ACM SIGCOMM conference on Internet measurement conference*, 2011, pp. 81–98.

[32] M. Eliáš, M. Kapralov, J. Kulkarni, and Y. T. Lee, "Differentially private release of synthetic graphs," in *Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms*. SIAM, 2020, pp. 560–578.

[33] C. Borgs, J. Chayes, A. Smith, and I. Zadik, "Revealing network structure, confidentially: Improved rates for node-private graphon estimation," in *2018 IEEE 59th Annual Symposium on Foundations of Computer Science (FOCS)*. IEEE, 2018, pp. 533–543.

[34] J. Ullman and A. Sealfon, "Efficiently estimating erdos-renyi graphs with node differential privacy," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[35] C. Task and C. Clifton, "A guide to differential privacy theory in social network analysis," in *2012 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*. IEEE, 2012, pp. 411–417.

[36] Q. Qian, Z. Li, P. Zhao, W. Chen, H. Yin, and L. Zhao, "Publishing graph node strength histogram with edge differential privacy," in *Database Systems for Advanced Applications: 23rd International Conference, DASFAA 2018, Gold Coast, QLD, Australia, May 21-24, 2018, Proceedings, Part II 23*. Springer, 2018, pp. 75–91.

[37] C. Yang, H. Wang, K. Zhang, L. Chen, and L. Sun, "Secure deep graph generation with link differential privacy," *arXiv preprint arXiv:2005.00455*, 2020.

[38] C. Yang, H. Wang, K. Zhang, and L. Sun, "Secure network release with link privacy," 2020.

[39] W.-Y. Day, N. Li, and M. Lyu, "Publishing graph degree distribution with node differential privacy," in *Proceedings of the 2016 International Conference on Management of Data*, 2016, pp. 123–138.

[40] X. Jian, Y. Wang, and L. Chen, "Publishing graphs under node differential privacy," *IEEE Transactions on Knowledge and Data Engineering*, 2021.

[41] K. R. Macwan and S. J. Patel, "Node differential privacy in social graph degree publishing," *Procedia computer science*, vol. 143, pp. 786–793, 2018.

[42] Y. Liu, S. Zhao, Y. Liu, D. Zhao, H. Chen, and C. Li, "Collecting triangle counts with edge relationship local differential privacy," in *2022 IEEE 38th International Conference on Data Engineering (ICDE)*. IEEE, 2022, pp. 2008–2020.

[43] Q. Ye, H. Hu, M. H. Au, X. Meng, and X. Xiao, "Lf-gdpr: A framework for estimating graph metrics with local differential privacy," *IEEE Transactions on Knowledge and Data Engineering*, vol. 34, no. 10, pp. 4905–4920, 2020.

[44] Z. Qin, T. Yu, Y. Yang, I. Khalil, X. Xiao, and K. Ren, "Generating synthetic decentralized social graphs with local differential privacy," in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, 2017, pp. 425–438.