entropy

MDPI

Article

Locally Differentially Private Heterogeneous Graph Aggregation with Utility Optimization

Zichun Liu *, Liusheng Huang, Hongli Xu and Wei Yang

School of Computer Science and Technology, University of Science and Technology of China, Hefei 230026, China
* Correspondence: lzc223@mail.ustc.edu.cn

Abstract: Graph data are widely collected and exploited by organizations, providing convenient services from policy formation and market decisions to medical care and social interactions. Yet, recent exposures of private data abuses have caused huge financial and reputational costs to both organizations and their users, enabling designing efficient privacy protection mechanisms a top priority. Local differential privacy (LDP) is an emerging privacy preservation standard and has been studied in various fields, including graph data aggregation. However, existing research studies of graph aggregation with LDP mainly provide single edge privacy for pure graph, leaving heterogeneous graph data aggregation with stronger privacy as an open challenge. In this paper, we take a step toward simultaneously collecting mixed attributed graph data while retaining intrinsic associations, with stronger local differential privacy protecting more than single edge. Specifically, we first propose a moderate granularity attributewise local differential privacy (ALDP) and formulate the problem of aggregating mixed attributed graph data as collecting two statistics under ALDP. Then we provide mechanisms to privately collect these statistics. For the categorical-attributed graph, we devise a utility-improved PrivAG mechanism, which randomizes and aggregates subsets of attribute and degree vectors. For heterogeneous graph, we present an adaptive binning scheme (ABS) to dynamically segment and simultaneously collect mixed attributed data, and extend the prior mechanism to a generalized PrivHG mechanism based on it. Finally, we practically optimize the utility of the mechanisms by reducing the computation costs and estimation errors. The effectiveness and efficiency of the mechanisms are validated through extensive experiments, and better performance is shown compared with the state-of-the-art mechanisms.

Keywords: data privacy; local differential privacy; graph aggregation; heterogeneous graph

Citation: Liu, Z.; Huang, L.; Xu, H.; Yang, W. Locally Differentially Private Heterogeneous Graph Aggregation with Utility Optimization. *Entropy* **2023**, *25*, 130. https://doi.org/10.3390/e25010130

Academic Editor: Boris Ryabko

Received: 17 November 2022
Revised: 24 December 2022
Accepted: 4 January 2023
Published: 9 January 2023

1. Introduction

Graph data are widely spread in people's lives from policy formation and market decisions to medical care and social interactions, whose exploitation and utilization are crucial to improve the overall quality of data-driven services. However, the heavy dependence of these services on personal graph data brings up serious concerns about the abuses of their private information. In recent years, a number of organizations have been exposed for abusing and compromising personal data privacy [1–3], and these incidents have caused huge financial and reputational damage to both organizations and their users. In order to avoid these negative outcomes, some countries and regions have actively enacted relevant laws to provide legislative authorities for privacy protection, such as GDPR [4] and CCPA [5]. Therefore, devising privacy protection mechanisms, which reveal overall valuable statistical information without violating the privacy of individual data, has become a top priority for organizations and research fields nowadays.

Due to its rigorous theoretical guarantees, differential privacy (DP) [6,7] has become the de facto standard of privacy preservation. DP mechanisms utilize a centralized trustworthy data curator to collect individual private data, and ensure that the overall output statistics does not reveal individual private information by adding calibrated noise to

aggregated results. As the scale of Web-enabled distributed devices grows, localized version of DP has been recently proposed to further reduce the risk of privacy breaches. Local Differential Privacy(LDP) [8] relies on no hypothetically trustworthy third-party data curator as in the conventional centralized DP, and provides on-device data perturbation and out-device purified statistics with rigorous privacy guarantee. Many companies have employed LDP based services, such as Apple [9,10], Google [11,12] and Microsoft [13].

LDP studies have been conducted in various fields, such as categorical data frequency publication [8,14–16] and numerical data mean estimation [13,17,18]. Subsequent research studies expand the scope to more complex data types, such as itemset release on set-valued data [19–21], decomposed distribution estimation on multidimensional data [22,23], and related data collection on key-value data [24,25]. However, studies on heterogeneous graph data are still scarce, which is a widely exploited data type in real-world applications, and data service providers wish to aggregate these heterogeneous graph data to analyze individual usage patterns and use them to improve the quality of services such as commodity recommendation [26,27], marketing [28] and pandemic tracking [29]. Consider heterogeneous social network as an example, each user (node) interacts through the social services belonging to multiple organizations or parties, and such communication linkages (edges) thus carry different numerical attributes, such as contacting frequencies and time intervals, and these attributes are potentially characterized as linkage weights. Users/organizations may also label part of edges as friendship, coworker-ship, kinship, political preference and sexual relationship. Accordingly, these attributed linkages represent the user engagement and usage frequency of corresponding social services, which is widely used in user profiling and recommendation systems. Another example is social–financial networks, where the users in one social network also have financial transactions. The social linkages between users may be attributed as friendship, coworkership and family, while the financial linkages between users contain fund transfer amount/time and trade amount/time. Aggregating the social–financial graph data is vital in marketing. As various social services provide location tracking systems, the so-called geosocial networks are also an important application of heterogeneous graphs. While part edges of the geosocial network are attributed social linkages, the geographical edges with trajectory distance and tracking time form a graph-based trajectory network. Combining and collecting these geosocial graphs provides significant pandemic tracking services.

Recently, graph data aggregation mechanisms under LDP constraint have been studied. By collecting perturbed degrees of pure graph data, Ref. [30] proposes to generate synthetic graph and [31] manages to aggregate subgraph statistics with extended privacy definition. Ref. [32] broadens the research scope to graph with node attributes. However, existing research studies mostly protect single edge privacy with edge-based LDP, while users in heterogeneous graph may require stronger privacy guarantee such as protecting a group of equally sensitive attributed edges from the statistical aggregation (e.g., protect all the sexual or political relations), and existing mechanisms may be insufficient to satisfy the potential heterogeneous graph privacy demands. Furthermore, existing mechanisms generally focus on single-attributed graph such as the weighted graph or the categorical-attributed graph, and leaves the heterogeneous graph aggregation challenge unresolved, which is to simultaneously collect mixed attributed graph data and intrinsic associations (between attributes and edges) while providing desirable utility.

In this paper, we take a step toward aggregating heterogeneous graph data with stronger local differential privacy protecting more than single edge. First of all, we characterize the two conventional variants of LDP definitionfor heterogeneous graph, integrate their characteristics and propose a fine-grained privacy definition with trade-offs between preservation strength and estimation accuracy. Under the moderate LDP definition, the problem of aggregating heterogeneous graph is addressed through two incremental stages, which are collecting categorical-attributed and heterogeneous graphs. For the former, we design a PrivAG mechanism to simultaneously sample and perturb subsets of encoded attribute and degree vectors, while retaining the relations reside within them. For the

latter, we present an optimal binning scheme to segment and merge mixed attributed data, which serves as a preceding subtask for subsequent mechanism. Lately, we extend PrivAG mechanism to uniformly aggregate heterogeneous graph, and further devise optimization techniques targeting user-side randomization and server-side estimation, achieving better privacy–utility tradeoff.

The main contributions of this paper are summarized as follows:

- We propose an attribute-wise local differential privacy (ALDP) notion with moderate granularity between conventional node-based LDP and edge-based LDP, trading-off privacy and utility between them, and formulate the problem of aggregating heterogeneous graph data under the ALDP notion as collecting attribute frequency and attribute-degree distribution.
- We apply padding and truncating for categorical-attributed graphs to handle the large data domain, and encode graph data as corresponding attribute and degree vectors. Then a utility-improved PrivAG mechanism is proposed to privately and simultaneously aggregate subsets of attribute and degree data.
- We present an adaptive binning scheme (ABS) to dynamically segment weighted edges and simultaneously collect mixed attributed data in the same process, reducing the computation cost to local devices and the estimation error caused by inconsistent distribution.
- We extend the privacy field to handle heterogeneous graphs and devise optimization techniques for user-side randomization and server-side estimation. The adaptive binning scheme and optimization techniques are integrated into the extended PrivHG mechanism.
- We validate the effectiveness and efficiency of our mechanisms based on extensive experiments, which are shown to have better performance than the state-of-the-art mechanisms.

The remainder of this paper is organized as follows. Section 2 introduces two conventional variants of LDP definition in graph data, and proposes a moderate attributewise local differential privacy. Section 3 formulates the problem of analyzing heterogeneous graph with ALDP and presents straightforward approaches. Section 4 proposes PrivAG mechanism for collecting the categorical-attributed graph. Section 5 designs an adaptive binning scheme to extend the privacy field to heterogeneous graph, and provides optimization techniques for extended PrivHG mechanism. Section 6 shows the extensive experimental results of PrivHG and baseline mechanisms. Section 7 reviews related literature. Finally, Section 8 concludes the paper.

2. LDP in Graph

In this section, two variants of local differential privacy in graph data are briefly introduced with their pros and cons. Then an eclectic notion is proposed to better trade-off privacy and utility for heterogeneous graph data in local settings.

Since its inception [6], differential privacy (DP) has become the standard for preserving private data. By introducing the concept of neighboring databases that only differ in one record, a randomized mechanism $\mathcal{M}$ under differential privacy constraint can guarantee statistical indistinguishability for these two databases D and D'. Although differential privacy has been extensively developed, practical scenarios lead to new challenges in local settings, therefore local differential privacy (LDP) is proposed [8], which relies on no trustworthy data curator and protects individual privacy on local devices. The privacy definition in local settings is based on user's perspective of local private data. As for graph data, two variants of LDP are given in [30] with different perspective, and we review two LDP definitions on local graph as follows:

Definition 1 (Node-based Local Differential Privacy [30]). *A randomized mechanism $\mathcal{M}$ satisfies ϵ-node local differential privacy if and only if for any two neighboring graph G, G' differing in one node, and any $O \in range(\mathcal{M})$,*

$$Pr(\mathcal{M}(G) \in O) \leq exp(\epsilon) \cdot Pr(\mathcal{M}(G') \in O)$$

Definition 2 (Edge-based local Differential Privacy [30]). *A randomized mechanism $\mathcal{M}$ satisfies ϵ-node local differential privacy if and only if for any two neighboring graph G, G' differing in one edge, and any $O \in range(\mathcal{M})$,*

$$Pr(\mathcal{M}(G) \in O) \leq exp(\epsilon) \cdot Pr(\mathcal{M}(G') \in O)$$

Despite the conventional privacy definitions of node-based LDP and edge-based LDP, there are certain drawbacks if applying them to heterogeneous graphs. On the one hand, node-based local differential privacy is a very promising and rigorous one, but directly applying node-based notion may introduce excessive noises and reduce the utility vastly. On the other hand, users may require a stronger notion than edge-based local differential privacy by protecting several equally sensitive attributed edges together, for the reason that similarly attributed relations deserve similar protection. Considering the privacy demand and the nature of attributed graph data, we combine the characteristics of these two notions, and propose an eclectic notion as attributewise local differential privacy(ALDP).

Definition 3 (Attributewise Local Differential Privacy). *A randomized mechanism $\mathcal{M}$ satisfies ϵ-attributewise local differential privacy, if and only if for any two neighboring attributed local graph data G, G' differing in one attribute and related edges, and any $O \in Range(\mathcal{M})$*

$$Pr[\mathcal{M}(G) \in O] \leq exp(\epsilon) \cdot Pr[\mathcal{M}(G') \in O]$$

Through trading off the rigorousness of node-based LDP and utility of edge-based LDP, we define neighboring private data from attribute level, that is to say, two attributed local graphs are neighboring if one can be obtained from another by altering one certain attribute along with all related edges. Intuitively, the privacy budget ϵ in ALDP is split among a subset of edges, where ϵ in node-based LDP is split in all edges and ϵ in edge-based LDP is used as a whole. Both node-based LDP and edge-based LDP can be viewed as extreme cases of ALDP. In one extreme case, the whole graph has only one attribute, and altering it is equivalent to altering all the edges, then ALDP corresponds to node-based LDP. In another extreme case, each edge of the whole graph has a distinct attribute value, and altering one certain attribute is equivalent to altering only one edge, then ALDP corresponds to edge-based LDP. Besides the extreme cases, ALDP in the nonextreme case actually trade-offs between the two definitions, thus achieving better estimation accuracy than the former and providing stronger privacy protection strength than the latter. In this paper, we aim to analyze edge-attributed graph under ALDP.

Some useful properties [8] of differential privacy provide theoretical guarantees for the design of subsequent mechanisms, the allocation of privacy budgets, and the optimization of perturbation results.

Theorem 1 (Sequential Composition [8]). *If randomized mechanism $\mathcal{M}_i$ satisfies ϵ_i-local differential privacy for $i = 1, \ldots, k$, then the sequential composition $\mathcal{M} = (\mathcal{M}_1, \ldots, \mathcal{M}_k)$ on private data G satisfies $\sum_1^k \epsilon_i$-local differential privacy.*

Theorem 2 (Parallel Composition [8]). *If randomized mechanism $\mathcal{M}_i(G_i)$ satisfies ϵ_i-local differential privacy for $i = 1, \ldots, k$, then the parallel composition $\mathcal{M} = (\mathcal{M}_1(G_1), \ldots, \mathcal{M}_k(G_k))$ on private data G satisfies $\max \epsilon_i$-local differential privacy.*

Theorem 3 (Postprocessing [8]). *If randomized mechanism $\mathcal{M}$ satisfies ϵ-local differential privacy, and f is a randomized mapping function, then $f \circ \mathcal{M}$ satisfies ϵ-local differential privacy.*

3. Problem Definition and Naive Approach

3.1. Problem Definition

Consider an edge-attributed graph as an undirected graph $G = (V, E, A)$, where V represents nodes in the graph, $E = \{e_{u,v} | u, v \in V\}$ represents edges, and each edge between two nodes is related to one attribute a_j from the universal attribute set A. Without lose of simplicity, in this paper we assume that each local graph may have several attributes but each edge of the graph has only one attribute. A graph with multidimensional attributes is beyond the scope of this paper, and we leave it for our future work. We assume that there are totally $|V|$ nodes, $|E|$ edges and $|A|$ attributes in graph G, which are all publicly known. Beyond the global parameters, local graph data G^i is stored on each individual i's device, and is considered as private. These private data include linked edges E^i and possessed attributes A^i. Take Figure 1 as an example to encode local graph data, user u holds four attributes from the universal attribute set (friend, coworker, kin, political, sexual), so the attribute vector of u is represented as $(1, 1, 0, 1, 1)$ with kinship as 0 as in upper right of Figure 1. There is one edge attributed as friend, which means the degree of attribute friend is 1, thus the first vector in lower right of Figure 1 is set to $(1, 0, 0, 0)$, so are other degree vectors. As for attributes not exist in graph, that rows are simply set to 0. The main notations are listed in Table 1.

Table 1. Notations.

Symbol	Meaning
$G(V, E, A)$	attributed graph
G^i	local graph of i-th user
A^i	possessed attribute set of i-th user
m	categorical attribute domain size $\|A_c\| = m$
w	numerical attribute domain size $\|A_n\| = w$
ℓ	maximum attributes each user have $\|A^i\| \leq \ell$
a_j	the jth attribute from $\mathcal{A}$
$deg^i(a_j)$	number of edges in G^i have attribute $a_j \in A$
θ	maximum degree bound
v_a	attribute vector
u_d	degree vectors
ϕ^a	frequency of attribute a
ψ^d	degree distribution of d

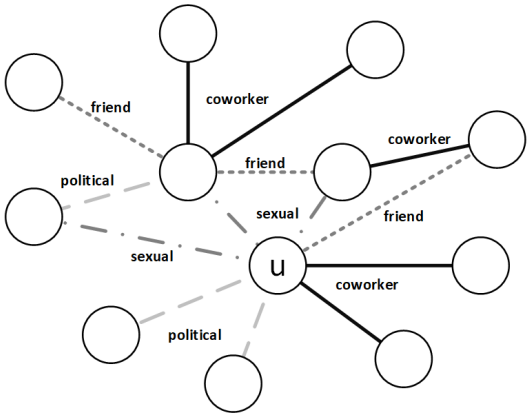
The objective of this paper is to provide tools for data curators to analyze heterogeneous local graphs, while satisfying ϵ-local differential privacy. Precisely speaking, through collecting perturbed attribute vector and attribute-degree vectors, we focus on estimating two fundamental statistics:

- Attribute frequency estimation. The attribute frequency ϕ^j is the ratio of users who possessed certain attribute a_j among whole users in the graph (e.g., fraction of users who installed certain social App among all Appstore users):

$$\phi^j = \frac{\#\{G^i | \exists a_j \in A^i\}}{n} \tag{1}$$

- Attribute-Degree distribution estimation. The attribute-degree distribution ψ_j^d is attributed version of degree distribution. Formally speaking, ψ_j^d is the number of nodes that have exactly d edges with attribute a_j:

$$\psi_j^d = \frac{\#\{G^i | \exists a_j \in A^i \text{ and } deg^i(a_j) = d\}}{n \cdot \phi^j} \tag{2}$$



encoded local graph of user u

friend	coworker	kin	political	sexual
1	1	0	1	1

	deg=1	deg=2	deg=3	deg=4
friend	1	0	0	0
coworker	0	1	0	0
kin	0	0	0	0
political	0	1	0	0
sexual	0	0	1	0

Figure 1. An example of attributed local graph G^u, right half is encoded vectors of G^u.

3.2. Data Preprocessing

Considering the practical flexibility, analyzing edge-attributed graph data still needs precaution before getting down to algorithm details. For attribute estimation, domain of real-world graph attributes can be enormous, and each user may possess several edge attributes in their local graph data. For simplicity, we assume, as in recent work [19,20], that the number of edge attributes in each user's local graph is fixed by parameter ℓ. As for degree estimation, when user opting out one attribute under attributewise local differential privacy, related edges will also be altered simultaneously, thus brings the high sensitivity of graph analysis, which in the worst case may reach the maximum degree, $|V| - 1$. Therefore, method to neutralize the effect of high sensitivity and retain better utility should be considered. Existing research studies mainly limit the magnitude of noises by projecting the original graph into a bounded graph with maximum degree equals θ.

In this paper, we first fix number of possessed attributes in each user's local graph, i.e., $|A^i| = \ell$. If a user has more than ℓ edge attributes in her local graph, she randomly sample ℓ attributes from the origin graph, together with related edges, forming a new graph with fixed ℓ edge attributes. For user with less than ℓ attributes, $\ell - |A^i|$ dummy items are padded to her graph, which are ignored by data curator in analyzing process. Then, for each attribute a_j possessed in user's local graph G_i, we set the maximum number of related edges in local graph as θ. When the number of edges with attribute a_j exceeds the given parameter, we truncate extra edges and bound the degree with θ. After the preprocessing of padding and truncating, we get the resulting attributed local graph $\bar{G}^i$. With the bounded local graph, we can further compare the allocation of privacy budget among different privacy notions: for attribute frequency estimation, budget ϵ_a in ALDP and edge LDP is used as a whole, while it is split as ϵ_a/ℓ for each attribute in node LDP; for degree estimation, budget ϵ_d in edge LDP is used as a whole, while it is split as ϵ_d/θ for each edge in ALDP and node LDP. In summary, ALDP strike a balance between edge LDP and node LDP.

3.3. Naive Approach

The first intuitive approach is adopting Laplace mechanism under ALDP, each user preprocesses her local graph into a bounded one, calculates numerical statistics and perturbs the statistics with Laplace noises. Specifically, given the initial parameters (including attribute set size ℓ, maximum degree θ and privacy budget ϵ), each user first encodes the

bounded graph as a numerical vector, in which each bit representing correlated number of edges with certain attribute, then adds Laplace noises sampled from $Lap(\frac{\Delta}{\epsilon})$. In the bounded local graph, changing one attribute will change the numerical vector at most $2\theta + 1$ bits, thus the sensitivity bound is $\Delta = 2\theta + 1$. Based on the Laplace approach, attribute-degree distribution can be estimated by aggregating the perturbed vectors from all users, and attribute frequency estimation can be derived from attributes with nonzero degree. However, the error of estimation is related to θ and the results can be highly inaccurate with a large θ, and when estimating attribute frequency solely, the sensitivity should be 2 instead of $2\theta + 1$, thus extra noises are added to the origin data.

Another naive approach to solve the problem is applying Randomized Response [33], and separately perturbing attribute possession and degree distribution by flipping two different coins. In particular, given the initial parameters, each user first encodes local graph as an attribute vector ϕ^a and degree vectors ψ^d, where ϕ^a is a binary vector indicating local attribute possession and ψ^d consists of m one-hot vectors denoting the degrees of corresponding attribute. By preprocessing and encoding local graphs, each user splits privacy budget ϵ in two parts ϵ_1 and ϵ_2 to perturb attribute and degree vectors respectively. In the attribute perturbation phase, the user splits ϵ_1 into 2ℓ parts and invokes GRR [11], which is an enhanced version of Randomized Response, to perturb bits in attribute vector with flipping probability: $p = \frac{1}{exp(\frac{\epsilon_1}{2\ell})+1}$. In the degree perturbation phase, the user splits ϵ_2 into 2θ parts, and perturbs the bits in degree vectors with probability: $p = \frac{1}{exp(\frac{\epsilon_2}{2\theta})+1}$. After the local perturbation, data curator collects perturbed vectors from all users and performs an unbiased estimation of attribute frequency and attribute-degree distribution. We regard this *GRR* approach as a baseline to our problem.

By observing these two approaches, some hurdles can be found. The conventional Laplacian mechanism is easy to implement, however, the noises added to origin data is θ-related, and the choice of θ is empirical and relies on specific graph data. Invoking Randomized Response twice as in *GRR* approach is a remedy to the problem, but pays the price of utility degrading by splitting privacy budget too fractionally. Furthermore, the attribute frequency and its degree distribution in one local graph should be correlated, and these two naive approaches fail to capture this property. In the next chapter, we tackle these hurdles in our PrivAG mechanism.

4. PrivAG Mechanism

4.1. General Mechanism

In order to tackle the aforementioned shortage of intuitive approach, we manage to reduce the fragmentation of privacy budget and retain the correlation between attribute possession and degree in PrivAG. The main idea of the mechanism is to first output an randomized attribute subset of fixed size k, where k relies on given parameters, and then accordingly perturb degree vectors based on the result of randomized attribute data. Specifically, PrivAG is comprised of two components, randomization and estimation component:

Randomization component. This component includes two phases that separately randomize attribute and degree vectors. One previously observed hurdle of naive approach is tiny split privacy budget and excessive noises, and the key coping idea for attribute randomization, which is inspired by the recent work [19], is to locally sample an attribute subset of size k as a whole to reduce introduced noises, without splitting privacy budget ϵ_1. As for degree randomization, inspired by research studies [34,35], we take OUE [34] as building block for degree randomization in this paper, which eliminates the effect of θ on the variance and transmits bit 1s and 0s differently. Note that the OUE method is replaceable in our mechanism, and GRR [33] or other methods could be an alternative for extremely sparse graphs with $\theta < 3exp(\epsilon_2) + 2$. After the randomization, additional postprocessing is executed to sustain the correlation between attribute and degree. The algorithmic detail of randomization component is presented in next subsection.

Estimation component. To reduce computational cost, data curator first broadcasts needed parameters to every user in the initializing phase of PrivAG, including public parameters and set size k in randomization component. k is calculated based on other public parameters, and the optimal k^* can be derived to further maximize utility, through treading off the theoretical error bounds of attribute frequency and degree distribution estimation. After each user invokes randomization component, data curator collects the perturbed data, and make an unbiased estimation about attribute and degree. Next, we present the two components in detail.

4.2. Data Randomization

Attribute randomization. In this phase, each user i encodes attribute set $\bar{A}^i \subseteq \bar{G}^i$ as a binary vector $v_a^i = (v_{a_1}^i, \ldots, v_{a_m}^i)$, then samples and outputs k elements from v_a^i with noisy probability consuming privacy budget ϵ_1. Denote the output as $\tilde{v}_k^i = (\tilde{v}_{a_1}^i, \ldots, \tilde{v}_{a_k}^i)$, which is one possible result from output domain $\mathbf{v}_k$ of all k-sized attribute vectors. This randomization phase is implemented based on the general Exponential Mechanism, which outputs element of maximum utility score u with the probability proportional to $exp(\frac{\epsilon u}{2\Delta u})$.

Given an input v_a^i and output domain $\mathbf{v}_k$ of all k-sized attribute vectors, we first define the essential utility function $u(v_a^i, \tilde{v}_k^i)$ to score the similarity between m-sized vector v_a^i and k-sized vector $\tilde{v}_k^i$ pairs. To keep the noisy probabilities stable, we define our utility function as a indicator function, indicating whether the ℓ_1 distance on the sampled k elements between v_a^i and $\tilde{v}_k^i$ is within k:

$$u(v_a^i, \tilde{v}_k^i) = [|v_a^i - \tilde{v}_k^i|_1 < k]$$

It can be derived that the sensitivity of utility function is 1:

$$\begin{aligned}\Delta u &= \max_{\tilde{v}_k^i \subseteq \mathbf{v}_k} \max_{||v_a^i - v_a^{i'}||_1 \leq 1} |u(v_a^i, \tilde{v}_k^i) - u(v_a^{i'}, \tilde{v}_k^i)| \\ &= \max_{\tilde{v}_k^i \subseteq \mathbf{v}_k} \max_{||v_a^i - v_a^{i'}||_1 \leq 1} |[|v_a^i - \tilde{v}_k^i|_1 < k] - [|v_a^{i'} - \tilde{v}_k^i|_1 < k]| = 1\end{aligned}$$

Through defining the low-sensitivity utility function $u(v_a^i, \tilde{v}_k^i)$, the noisy probability of outputting $\tilde{v}_k^i$ with input v_a^i is given by:

$$Pr[\mathcal{M}(v_a^i) = \tilde{v}_k^i] \propto exp(\frac{\epsilon_1 u(v_a^i, \tilde{v}_k^i)}{\Delta u}), \tilde{v}_k^i \subseteq \mathbf{v}_k$$

Substituting $\Delta u = 1$, the attribute randomization probability can be derived by aggregating all the proportional probabilities above:

$$Pr[\mathcal{M}(v_a^i) = \tilde{v}_k^i] = \frac{exp(\epsilon_1 u(v_a^i, \tilde{v}_k^i))}{\sum_{\tilde{v}_k \in \mathbf{v}_k} exp(\epsilon_1 u(v_a^i, \tilde{v}_k))}$$

The implementation of attribute randomization phase is illustrated in the Algorithm 1 from line 2 to line 14. During the line 2 and line 8, each user computes a series of probabilities, where Σ is the normalizer $\Sigma = \sum_{\tilde{v}_k \in \mathbf{v}_k} exp(\epsilon_1 u(v_a^i, \tilde{v}_k))$, and each p_i with $i \in [0, k]$ represents the probability that the number of selected origin attributes $a \in \bar{A}^i$ is exactly i, $p_i = Pr[\#\{a_j | a_j \in A^i \text{ and } \tilde{v}_k[a_j] = 1\} = i]$. Since u is an indicator function, the output domain of $\mathbf{v}_k$ contain $\binom{m+\ell}{k}$ outputs, and $u = 0$ when selecting k attributes from noninitial $m + \ell - \ell = m$ attributes, so Σ can be calculated with given parameters. The probabilities p_i is calculated iteratively. From line 8 to line 14, each user randomly generates a number k_s based on the previous probabilities, separately samples k_s attributes from $\bar{A}_i$ and samples $k - k_s$ attributes from the rest noninitial attributes set, and vectorization the union set as $\tilde{v}_k^i$, which is the perturbed attribute vector to be contributed.

Algorithm 1 Data Randomization Component (DRC).

Input: attributed local graph G^i, privacy budget ϵ, m, l, θ
Output: perturbed attribute–degree vectors $\hat{s} \in S$

//locally truncate and pad origin graph
$\bar{G}^i \leftarrow pre - processing(G^i)$
//attribute perturbation
$\Sigma \leftarrow \binom{m}{k} + exp(\epsilon_1)(\binom{m+\ell}{k} - \binom{m}{k})$
$p_0 \leftarrow \frac{\binom{m}{k}}{\Sigma}$
for $i \in [1, k]$ **do**
    $p_i \leftarrow p_{i-1} + \frac{exp(\epsilon_1)\binom{m}{i}\binom{\ell}{k-i}}{\Sigma}$
end for
$r_{att} \leftarrow random(0.0, 1.0)$
$k_s \leftarrow 0$
while $p_{k_s} \leq r_{att}$ **do**
    $k_s \leftarrow k_s + 1$
end while
$\tilde{v}_k \leftarrow vectorization(sample(k_s, A^i) \cup sample(k - k_s, A - A^i))$
for $a_j \in \tilde{v}_k$ *and* $a_j \notin G^i$ **do**
    $t \leftarrow random(1, \theta)$
    $u_t^j \leftarrow 1$
end for
//degree randomization
for $a_j \in \tilde{v}_k$ **do**
    for $t \in [1, \theta]$ **do**
        Perturbs as

$$Pr[\tilde{u}_t^j = 1] = \begin{cases} p_d = \frac{1}{2}, & u_t^j = 1 \\ q_d = \frac{1}{e^{\epsilon_2/k} + 1}, & u_t^j = 0 \end{cases}$$

    end for
end for
for $a_j \in A$ **do**
    $\tilde{u}^j \leftarrow \tilde{u}^j a_j$
end for
return $\tilde{v}_k$ and $\tilde{u}_d$

Degree randomization. As mentioned above, we adopt OUE to serve as our attribute-degree perturbation primitive. To be specific, for a individual's local graph G^i, after projected to $\bar{G}^i$, the number of edges with every attribute is known and limited, thus can be encoded as one-hot attribute-degree vector $u^j = [u_0^j, \ldots, u_\theta^j]$, where the subscript j stands for attribute $a_j \in A^i$ and only $deg^i(a_j)$-th bit is 1 in this degree vector of attribute a_j. For one-hot vectors like u_d, OUE takes noncomplementary probabilities for bits 1 and 0, bit 1 in u_d stays as 1 in $\tilde{u}_d$ with probability $p = 1/2$, in the meantime, bits 0 in u_d are flipped with probability q. The general randomization process can be sketched as:

$$Pr[\mathcal{M}(\tilde{u}_i = 1)] = \begin{cases} p, u_i = 1 \\ q, u_i = 0 \end{cases}$$

One shortage of baseline approach is that it fails to capture the intrinsic correlation between attribute and its degree, for example when a_j is perturbed as 0 after attribute randomization phase, the related degree vector u_d^j should also be 0 after perturbation. After the attribute randomization phase in PrivAG, there are k selected attributes as a whole to be perturbed as 1, with the rest bits in attribute vector v_a as 0, thus there should be k related

vectors u_d with nonzero degree. Degree randomization process is executed k times for each selected attribute in $\tilde{v}_k^i$, with split privacy budget ϵ_2/k, and for the rest attributes not in $\tilde{v}_k^i$, the degree vector is postprocessed to stay 0 with attributes simultaneously. Furthermore, parameter k take a role in both phases, and the optimal k is determined by estimation of both phases in theoretical analysis section. Therefore, the correlation between attribute and degree is retained. As shown from line 18 to line 24, in degree randomization phase, each user splits privacy budget as ϵ_2/k, and utilizes each share to flip one attribute-degree vector of selected attribute from attribute randomization phase.

By combining the two phases, the Data Randomization Component(DRC) is presented in Algorithm 1, in the initializing stage, each user gets public parameters from data curator, preprocesses local graphs, and divides privacy budget as $\epsilon = \epsilon_1 + \epsilon_2$ for subsequent randomization. In the final stage, each user multiples the perturbed degree vectors with the related attribute vector value, ensuring that the two phases are perturbed simultaneously.

4.3. Distribution Estimation

In this subsection, we present the complete PrivAG framework, including attribute frequency and attribute-degree distribution estimation component. In Algorithm 2, each user executes DRC on her edge-attributed local graph, and contributes the sanitized results to data curator. After collecting the perturbed data from all users, data curator aggregates the results and accordingly infers the attribute frequency ϕ^a and attribute-degree distribution ψ^d. The thorough estimation phase of PrivAG framework is given below.

Algorithm 2 PrivAG.

Input: local graphs G, privacy budget ϵ
Output: attribute frequency ϕ^a, attribute–degree distribution ψ^d
1: //user-side randomization
2: each user locally perturbs G^i by DRC, and report $\tilde{v}_k^i$ and $\tilde{u}_d^i$
3: //count bits 1 in the randomized vectors
4: $c_j \leftarrow count(\tilde{v}_k)$
5: $d_t(a_j) \leftarrow count(\tilde{u}_d)$
6: //estimate from recorded counts
7: **for** $j \in [1, m]$ **do**
8: $\quad \phi^j = \frac{c_j/n - q_a}{p_a - q_a}$
9: **end for**
10: **for** $j \in [1, m]$ *and* $t \in [0, \theta]$ **do**
11: $\quad \psi_j^t = \frac{d_t(a_j)/n - q_d}{p_d - q_d}$
12: **end for**
13: **return** ϕ^a and ψ^d

Attribute frequency. In this phase, data curator aggregates bit 1s in perturbed vector $\tilde{v}_k$ from n individuals as a counting vector $c_j = \#\{\tilde{v}_k^i | \tilde{v}_k^i[a_j] = 1\}$ and calibrates the frequency. During the calibration process, two probabilities are critical, which we denote as p_a and q_a. For a user i and an attribute $a_j \in A$, if a_j both appears in the origin vector of A^i and the perturbed result $\tilde{v}_k$, the probability is denoted as p_a:

$$\begin{aligned} p_a &= Pr[\tilde{v}_k[j] = 1 | v_j = 1] \\ &= \frac{exp(\epsilon_1)\binom{m+\ell-1}{k-1}}{\binom{m}{k} + exp(\epsilon_1)(\binom{m+\ell}{k} - \binom{m}{k})} \end{aligned} \tag{3}$$

Similarly, if a_j is beyond user's possessed attributes $a_j \notin A^i$, but the perturbed result $\tilde{v}_k$ contains a_j, then the probability is denoted as q_a:

$$\begin{aligned} q_a &= Pr[\tilde{v}_k[j] = 1 | v_j = 0] \\ &= \frac{\binom{m-1}{k-1} + exp(\epsilon_1)(\binom{m+\ell-1}{k-1} - \binom{m-1}{k-1})}{\binom{m}{k} + exp(\epsilon_1)(\binom{m+\ell}{k} - \binom{m}{k})} \end{aligned} \tag{4}$$

By calculating these probabilities p_a, q_a and the counting vector c_a, the unbiased estimation of attribute frequency a_j is:

$$\phi^j = \frac{c_j/n - q_a}{p_a - q_a} \tag{5}$$

Attribute-Degree distribution. The estimation of attribute-degree distribution is pretty similar to the previous phase. Data curator first aggregates bits 1s in vectors $\tilde{u}_d$ contributed by n individuals as a counting vector $d_t(a_j) = \#\{\tilde{u}_d^i | \tilde{u}_t^i[a_j] = 1\}$. By combining the two important probabilities already given in Algorithm 1: $p_d = \frac{1}{2}$ and $q_d = \frac{1}{1+e^{\epsilon_2/k}}$. Then data curator estimates the attribute-degree distribution as:

$$\psi_j^t = \frac{d_t(a_j)/n_j - q_d}{p_d - q_d} \tag{6}$$

With split privacy budget $\epsilon = \epsilon_1 + \epsilon_2$, the above mechanism PrivAG satisfies ϵ-attributewise local differential privacy. Upon analysis, the categorical-attributed graph with PrivAG mainly has two kinds of errors, relating to two estimation objectives. Next, we theoretically analyze these two errors and optimize key parameter to reduce estimation variance.

Error analysis on attribute frequency estimation. Based on the probabilities calculated in the previous subsection, the variance of an attribute $a_j \in A$ frequency is:

$$Var[\phi^j] = \frac{nq_a(1-q_a)}{(p_a - q_a)^2} \tag{7}$$

Error analysis on attribute-degree distribution estimation. Similarly, the variance of an attribute $a_j \in A$ frequency is:

$$Var[\psi_j^t] = \frac{nq_d(1-q_d)}{(p_d - q_d)^2} \tag{8}$$

5. PrivHG: Extending to Heterogeneous Graph

The aforementioned PrivAG mechanism is efficient and effective to perform analysis tasks for categorical-attributed graph. In this section, the privacy field is generalized from categorical attribute to heterogeneous attributes, such as the heterogeneous social networks, social–financial networks and geosocial networks, and an enhanced version of PrivAG mechanism (denoted as PrivHG) is presented to aggregate two statistics ϕ^a and ψ^d of local heterogeneous graph. The estimation accuracy and computation overhead of PrivHG are further optimized both in user-side randomization component and server-side estimation component.

The premise of extending PrivAG from categorical-attributed graph to heterogeneous graph is to collect categorical and numerical attribute possessions $\#a|a \in A$ and mixed-attributed edge degrees $\{deg(a)|a \in A\}$. An available approach is to separately collect these mixed statistics: leave categorical-attributed data to PrivAG and aggregate numerical-attributed data with hierarchy-based approach. The hierarchy approach commonly constructs an additional hierarchical structure and perturbs private data with multiple privacy granularity. Despite the additional computation cost of the hierarchical data structure building process, the limited privacy budget ϵ will be allocated proportionately between PrivAG

and hierarchy-based mechanisms when separately randomizing categorical-attributed and numerical-attributed data, which is also inefficient and impractical. Therefore, it is inappropriate to apply hierarchy-based approach for numerical data of heterogeneous graph in PrivHG mechanism. Another way is to apply binning-based approach to segment continuous numerical attributes $a \in A_n$ into discrete r intervals with binning scheme $B = (b_1, b_2, \ldots, b_r)$, and deal with them equally as categorical data. By applying binning-based approach in PrivHG mechanism, categorical-attributed and numerical-attributed statistics $a|a \in A_c \cup B$ and $\{deg(a)|a \in A_c \cup B\}$ can be aggregated under the same process, and privacy budget ϵ is utilized as a whole. In the following, PrivHG adopts binning-based approach as a building block to collaboratively analyze private heterogeneous graph along with PrivAG, a resizing binning technique is further designed in PrivHG to handle the large domain problem of heterogeneous graph, which reduces the aggregation and estimation error compared with straightforward application of binning-based approach.

Despite the intuitive outline of the extended PrivHG mechanism, there are still limitations on the details of aggregating heterogeneous graph data, leaving room for following improvement.

- In the initialization process of PrivHG mechanism, the binning scheme $B = (b_1, b_2, \ldots, b_r)$ is directly applied to the local data to aggregate statistics, thus determines the estimation accuracy brought by subsequent truncation and perturbation processes. Since the heterogeneous graph data are potentially distributed unevenly but truncated uniformly with maximum degree parameter θ, different binning scheme $B \in \mathcal{B}$, which groups data with different granularity of sparsity, affects the gap between truncation range $[1, \theta]$ and actual data range $[min(deg(b_i)), max(deg(b_i))]$ for each bin $b_i \in B$, therefore having an impact on the accuracy of subsequent attribute-bins-related estimation for $a_j \in b_i$. To be more specific, considering two extreme cases: if the binning scheme B is too fine, most bins $b_i \in B$ contain only sparse attributed edges and aggregation of these sparse data falls far below the truncation threshold $max(deg(b_i)) \ll \theta$, then excessive θ-related noises are introduced into these sparse bins and associated attributed graph during perturbation process; On the other hand, a too coarse binning scheme B groups numerous attributed edges together, then the aggregated statistics of these large bins $b_i \in B$ may exceed the truncation parameter θ too much $max(deg(b_i)) \gg \theta$, resulting in enormous error due to excessive attributed edges being truncated. Note that this unevenly distributed but uniformly truncated limitation applies for both categorical-attributed and numerical-attributed data in heterogeneous graphs. Therefore, finding optimal binning scheme in unified PrivHG mechanism is critical, and perturbation with inappropriate binning scheme could suffer from high randomization error with sparse data and high truncation error with dense data.
- During the randomization process of heterogeneous graph, the intrinsic correlations between attributed edges need to be reflected in the simultaneous randomization of attributes (bins) and degrees, and retained in the estimation of attribute frequency ϕ^a and degree distribution ψ^d. Especially, if a nonpossessed attribute $(a_j = 0)$ is perturbed as a possessed one on $(a_j = 1)$ when perturbing a private local graph, a fake attributed degree $deg(a_j)$ needs to be generated as a counterpart; On the contrary, if a possessed attribute $(a_j = 1)$ is perturbed as a nonpossessed one on $(a_j = 0)$, related degree $deg(a_j)$ is set to 0. The fake degree $deg(a_j)$ in PrivAG mechanism is randomly generated from range $[1, \theta]$ without prior knowledge, which skews the estimation results of degree distribution ψ^d.
- The sampling size k of randomized data set is determined on the server side without considering local devices' capabilities, which lead to $O(k * \theta)$ computation and communication overhead on the user side. A large k represents that lots of data needs to be sampled, randomized and contributed from each user, which means much burden to user's device. However, practical local devices have various capabilities, and imposing heavy burden to the low-capability local devices in turn brings difficulties to data collection.

- The randomization strength and estimation accuracy in PrivAG mechanism is controlled by the privacy budget ϵ. When ϵ is split many times among heterogeneous graphs, the outliers generated by data randomization component may obscure graph data characteristic and have a relatively huge impact on the estimation results, but PrivAG mechanism is lack of corresponding techniques to correct randomization outliers and neutralize estimation variance.

The overview of PrivHG mechanism is shown in Figure 2, which mainly extends the privacy field to heterogeneous graph data and optimizes the above limitations. Taking the real-world applications of heterogeneous social network and social–financial network as examples, the brief process of running PrivHG mechanism can be summarized as: First, during the initialization phase, the whole heterogeneous social graph or social-financial network is divided as two user groups. Users in group 1 preprocess the numerical attributes (e.g., contacting time intervals in heterogeneous social network or fund transfer amount in social–financial network) according to the binning scheme, and encode their local graph data as illustrated in Figure 1. Then the numerical-attributed data (e.g., contacting time intervals and fund transfer amount) and categorical-attributed data (e.g., social linkage type and financial activity type) are equally randomized and collected with randomization mechanism. After the data curator aggregates the statistics, a generalized optimal binning scheme is output, covering mixed attributes with minimal estimation error. In the following phase, the optimal binning scheme and necessary parameters are informed to user group 2, and each user preprocesses and randomizes his/her local data with optimization techniques. Finally, these randomized vectors are aggregated by the server, then unbiased estimations about the data distribution of heterogeneous social network or social–financial network are generated. Specifically, the techniques of extending to PrivHG mainly include the following components: Adaptive Binning Scheme is firstly proposed to find an optimal binning scheme $B_o \in \mathcal{B}$ for mixed attributes based on a portion of the heterogeneous graph data (Section 5.1). The binning B_o in ABS strikes a balance between truncation and perturbation error, ensuring that the final aggregated statistics are approximately around the threshold $max(deg(b_i)) \approx \theta$ for $b_i \in B_o$. Then, the byproducts of Adaptive Binning Scheme enable subsequent optimizations. During the perturbation process, the sample set size k is chosen by trading off communication overhead and estimation accuracy (Section 5.2), and correlated fake degrees are calibrated based on the estimated data distribution in ABS rather than random values (Section 5.2). Finally, considering the heterogeneous graph data properties, the aggregated statistics are corrected by filtering out the outliers (Section 5.3).

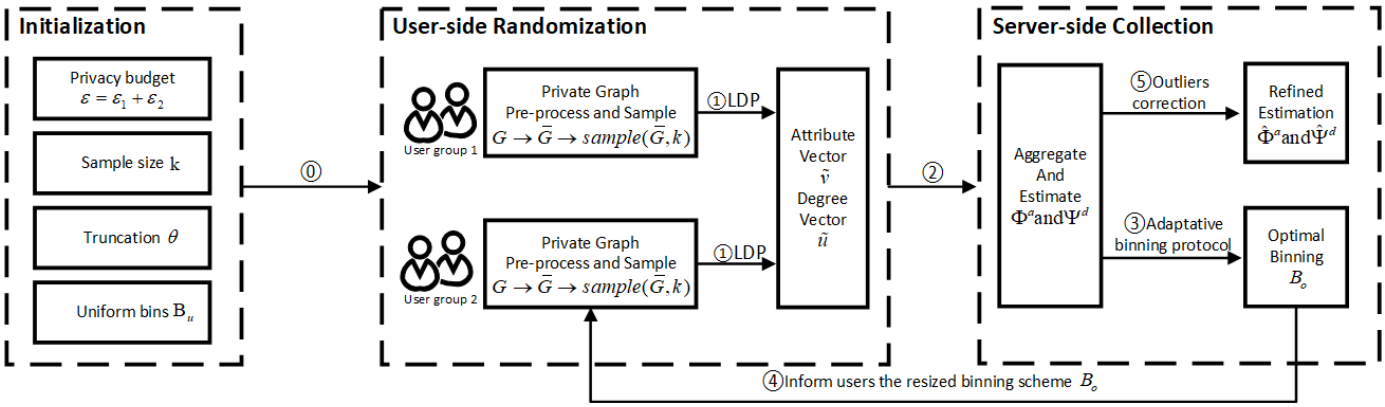


Figure 2. Overview of PrivHG mechanism.

5.1. Adaptive Binning Scheme

This subsection elucidates the process of finding the optimal binning scheme B_o in Figure 2. As previously stated, binning schemes are designed to discretize numerical-attributed data, so that heterogeneous graphs can be perturbed uniformly with PrivHG mechanism. As different binning schemes influence the final estimation accuracy differently,

the intuition to find a proper binning scheme requires keeping aggregated statistic of each bin as close to the truncation threshold as possible $max(deg(b_i)) \approx \theta$ for $b_i \in B_o$, in the mean time minimizing both the estimation error from perturbations and the truncation error from binning, and reducing the dependence on background knowledge of data distribution.

Two basic binning schemes for heterogeneous graph data are uniform binning and geometric binning. Uniform binning is pretty straightforward and intuitive. For numerical attribute range $a \in [1, w]$, uniform binning divides it into bins with equal width, $B = b_i | i \in (1, r)$ where $b_i = (1 + (i-1) * \delta, 1 + i * \delta)$ and $\delta = \frac{w-1}{r}$. Geometric binning is another feasible scheme, where the bins are covered by a geometric series δ^i and the width of bins varies from narrow to wide, which mimics the long tail distribution nature of some graph data. Formally speaking, the $[1, w]$ interval is divided geometrically as $b_i = (1 + \delta^{i-1}, 1 + \delta^i)$, where $i \in (1, r)$ and δ is a predefined parameter controlling the variations of bin width. However, there are drawbacks when applying the two basic binning schemes. First of all, finding the parameter δ that controls the width of bins in the binning schemes requires practical experience, and to guarantee finding the optimal parameter is a nontrivial effort. Second, the two binning schemes rely on certain data distribution to achieve accurate estimation and they may perform poorly in other scenarios, for example, uniform binning suffers from the unevenly distributed but uniformly truncated problem, and geometric binning suffers from nongeometric data distributions. Third, once the two binning schemes are defined, they are only suitable for the covered graph and not applicable to other graphs. Last but not least, uniform binning scheme may cover a set of categorical attributes and geometric binning scheme may cover a set of numerical attributes, but the PrivHG mechanism requires a unified scheme to cope with mixed attributes of heterogeneous graphs. As a precursor subtask of the PrivHG mechanism, we propose Adaptive Binning Scheme (ABS), which integrates the merits of above schemes and allows PrivHG to be conveniently extended to the mixed-attributed data of heterogeneous graph.

As shown in Algorithm 3, ABS first divides numerical attribute as discrete intervals with basic binning scheme (we take uniform binning $B_u = (1 + (i-1) * \delta, 1 + i * \delta)$ where $i \in (1, r)$ and $\delta = \frac{b-1}{r}$ in PrivHG for simplicity, while geometric or other binning is alternative), and aggregates both numerical-attributed and categorical-attributed data of heterogeneous graph. Then ABS estimates the error and cost variations of resizing and merging the bins for all possible binning schemes $B \in \mathcal{B}$, and finds the binning scheme with minimum overall cost. Finally the optimal binning B_o is distributed to subsequent subtasks. Comparing with uniform and geometric binning scheme, the benefits of ABS are evident: 1. The large domain problem of heterogeneous graph and predefined binning is neutralized in ABS by combining sparse bins and reducing overall bin counts. 2. Attributed data that are comparatively below the maximum degree threshold θ are collected simultaneously, trading off truncation error and perturbation noises. 3. ABS is feasible for both numerical- and categorical-attributed graph data, which collects heterogeneous data under one mechanism and avoids overdivision of privacy budget. 4. Byproduct of ABS provides access to subsequent optimization techniques of PrivHG, which is illustrated in the following subsection.

Based on the essential objective of adaptive binning scheme is to ensure that the maximum aggregated degree of each bin is as close to the truncation parameter θ as possible, while ensuring the overall cost of executing ABS as small as possible. We formalize the ABS objective as minimizing the following three components of overall cost under the constraint of predefined parameter θ:

Binning Resize Cost. This component captures the cost of binning and truncating processes for the resulting estimation, which mainly introduced by the resizing from basic bins to optimal bins. For aforementioned basic bins with fixed bin size for single attribute $a \in A$, if the correlated maximum degree is below the truncation threshold $max(deg(a)) < \theta$, then vacant bits $[u_{max(deg(a))+1}, \ldots, u_\theta]$ in the encoded degree vector u are randomized as outliers after executing the perturbation mechanism with probability $q_d = \frac{1}{1+e^{\epsilon_2/k}}$, which further reduces the estimation accuracy. The larger deviation between

correlated maximum degree and parameter $|\theta - max(deg(a))|$, the more vacant bits are randomized as outliers, therefore the higher estimation error is. By merging sparse and low-degree attributed data (basic bins), binning scheme B enables that the aggregated maximum degree of merged attributes/bins ($b \in B$) is close to truncation threshold $max(deg(b)) \approx \theta$, which reduces the error of perturbing the vacant bits in the related degree vectors. Due to the reduction of vacant bits, binning part of overall resizing cost in general is a negative value. However, under extreme circumstances, some merged bins may also lead to extra data being truncated. For the merged attributes/bins $a \in b$, if the related degree exceed θ, then extra truncating cost is denoted by degrees$\sum_{a \in b} deg(a) - theta$. On the other hand, if the maximum aggregated degree of merged bin $max(deg(b))$ is still below the truncating parameter θ, then additional truncating cost of resizing this bin is 0. Summing the binning costs and truncating costs up, the overall resizing cost is given as below, where B is a binning scheme, q_d is the probability of randomizing vacant bits as 1 and ϵ_2 is privacy budget for degree randomization.

$$\begin{aligned} RC(B, u_d, \theta, \epsilon_2, k) &= BC(B, u_d, \theta, \epsilon_2, k) + TC(B, u_d, \theta) \\ &= \sum_{b_i \in B} (|\theta - max(deg(b_i))| q_d - \sum_{a \in b_i} |\theta - max(deg(a))| q_d) \\ &+ \sum_{b_i \in B} max(\sum_{a \in b_i} deg(a) - \theta, 0) \\ &= \sum_{b_i \in B} (\frac{1}{1 + exp(\epsilon_2/k)} (|\theta - max(deg(b_i))| - \sum_{a \in b_i} |\theta - max(deg(a))|) \\ &+ max(\sum_{a \in b_i} deg(a) - \theta, 0)) \end{aligned} \tag{9}$$

Attribute Randomization Cost. This component captures the cost brought by binning schemes for the attribute randomization. After ABS merging bins with attribute $a \in A$ (or basic bin $b \in B$) on the server side, the possession of local attributes is replaced by the possession of local merged bins, thus $\bar{v}_i = 1$ if $\exists v_a = 1$ and $a \in b_i$, and $\bar{v}_i = 0$ if $\forall v_a = 0$ and $a \in b_i$. During local randomization component, if a merged bin $b_i \in B$ is possessed by user, the indicating bit in binning vector is set to 1 $\bar{v}_i = 1$, which is equivalent to all corresponding attribute bits being estimated as 1 $v_a = 1$ for $a \in b_i$, while these attributes may not all be possessed by local user and the actual value of these bits may be 0. Attribute randomization cost comes from the difference between indicating bits in resized binning vector $(\bar{v}_1, \ldots, \bar{v}_i, \ldots, \bar{v}_r)$ for merged bins $b_i \in B$ and indicating bits in attribute vector $(v_1, \ldots, v_j, \ldots, v_m)$ for attributes $a_j \in A$, which is formalized as $AC(B, \bar{v}_b, v_a)$.

$$AC(B, \bar{v}_b, v_a) = \sum_{b_i \in B} \sum_{a_j \in b_i} |\bar{v}_i - v_j| = \sum_{b_i \in B} \bar{v}_i (|b_i| - \sum_{a_j \in b_i} v_j) \tag{10}$$

Degree Estimation Cost. This component captures the cost brought by binning scheme for the degree estimation. When aggregating attributed degrees based on the binning scheme B in ABS, the degree of each attribute a is estimated as the average degree of related bin $b_i \in B$ for $a \in b_i$. Furthermore, if the merged degrees of bins exceed truncation parameter θ, extra edges are truncated on local devices, then the aggregated degree of each bin $b_i \in B$ is the minimum value of parameter θ and sum of attributed degree $deg(b_i) = \sum_{a \in b_i} deg(a)$; therefore, the estimated degrees of the including attributes are replaced by the statistical average $\hat{deg}(a) = \frac{deg(b_i)}{|b_i|}$ for $a \in b_i$. Degree Estimation Cost comes from this deviation.

$$DC(B, u_d, \theta) = \sum_{b_i \in B} \sum_{a \in b_i} |deg(b_i) - deg(a)| = \sum_{b_i \in B} \frac{|b_i| - 1}{|b_i|} min(\sum_{a \in b_i} deg(a), \theta) \tag{11}$$

Combining these three components, the objective function of overall binning scheme cost can be summarized as following, and an optimized binning scheme is found by solving this Minimum Binning Scheme Cost Problem.

$$\min \sum_{b_i \in B} (RC(b_i, u_d, \theta, \epsilon_2, k) + AC(b_i, \bar{v}_b, v_a) + DC(b_i, u_d, \theta))$$
$$s.t. \begin{cases} u_d, \bar{v}_i, v_j \in \{0,1\} d \in [1, \ldots, \theta] \\ i \in [1, \ldots, r] \\ j \in [1, \ldots, m] \\ 1 \leq k \leq r \leq m \\ \epsilon_2 = \epsilon/2 \end{cases} \tag{12}$$

Algorithm 3 Adaptive Binning Scheme (ABS).

Input: Local graphs G, attribute frequency ϕ^a, attribute-degree distribution ψ^d, privacy budget ϵ, basic binning B_u.
Output: Optimized binning scheme B_o with minimal overall cost.

```
//compute cost for all possible binning schemes
for B ∈ 𝓑 do
    //merge basic bins a ∈ B_u with degree truncation
    for b_i ∈ B and a ∈ b_i do
        v̄_i = 1 if ∃ v_a = 1 and a ∈ b_i
        deg(b_i) = min(∑_{a∈b_i} deg(a), θ)
        d̂eg(a) = deg(b_i) / |b_i|
        //compute three cost components for merged bins
        RC(b_i) = (1/(1+exp(ε))(|θ − max(deg(b_i))| − ∑_{a∈b_i} |θ − max(deg(a))|) + max(∑_{a∈b_i} deg(a) − θ, 0))
        AC(b_i) = v̄_i(|b_i| − ∑_{a_j∈b_i} v_j)
        DC(b_i) = (|b_i|−1)/|b_i| min(∑_{a∈b_i} deg(a), θ)
    end for
end for
//solving the objective function
B_o = argmin_{B∈𝓑} ∑_{b_i∈B} (RC(b_i) + AC(b_i) + DC(b_i))
return B_o
```

The pseudocode of Adaptive Binning Scheme is presented in Algorithm 3, which mainly computes the overall cost for each possible binning scheme in universal set $\mathcal{B}$ and outputs the optimized one. Due to its independence on background knowledge, ABS relies on the estimation of noisy graph data, where each user locally counts the statistics based on uniform binning B_u of heterogeneous attributes and randomly perturbs graph data with privacy budget ϵ (Note that uniform binning B_u in ABS is alternative and other reasonable binning scheme is applicable). After local private graph being perturbed with binning and truncating processes, data curator correspondingly collects the estimation of binning vectors $(\bar{v}_1, \ldots, \bar{v}_i, \ldots, \bar{v}_r)$ and degrees $deg(b_i)$ for each bin $b_i \in B$, then the overall cost of a binning scheme $B \in \mathcal{B}$ is calculated according to Equations (9)–(11). Finally, the optimal binning scheme B_o is obtained with dynamic programming by solving Minimum Binning Scheme Cost Problem in Equation (12). Because ABS is executed on the server side, it brings no computational overhead to local devices.

5.2. Randomization Optimization

On the basis of aforementioned optimal binning scheme B_o generated by ABS, minimal overall binning cost is achieved when aggregating heterogeneous graph data. One direct benefit is that ABS generally scales the domain size of graph attributes from $|A| = m$ down to $|B_o| = r$ and reduces the storage and communication burden reduction on local devices.

Furthermore, this subsection continues to provide optimizing strategies for user-side local randomization of the PrivHG mechanism, which mainly contains two parts sampling subset size and fake degree generation.

Sampling subset size. During the data randomization component of PrivHG, k-sized subset of attribute and degree data are sampled, randomized and contributed, with privacy budget ϵ split among these k pairs of data. Therefore, altering k strictly affects estimation accuracy, budget usage and communication overhead. Under the circumstance that privacy budget and communication resources are sufficient, theoretically optimal parameter k_o can be selected by taking estimation accuracy into account and minimizing the variance.

$$k_o = argmin(\sum_{j\in[1,r]} Var[\phi^j] + \sum_{j\in[1,r],t\in[1,\theta]} Var[\psi_j^t])$$

Although the derivation of a closed-form optimal k_o is almost impossible, due to the complexity of computing variances $Var[\phi^j]$ and $Var[\psi_j^t]$, k_o can still be selected from thorough computation based on public parameters. Before distributing parameters for PrivHG, data curator first computes all $Var[PrivHG] = \sum_{j\in[1,r]} Var[\phi^j] + \sum_{j\in[1,r],t\in[1,\theta]} Var[\psi_j^t]$ of every possible $k \in [1, m]$, and select one with minimal variance as the k_o. However, under circumstances where privacy budget or communication resources are limited, a large k will bring about much difficulties in practical execution of PrivHG. Therefore, a feasible approach is to sacrifice a minor proportion of estimation accuracy in exchange for a communication overhead reduction and overall privacy budget utilization by fixing $k = 1$, which is denoted as k_e-PrivHG.

Deployment of PrivHG on heterogeneous graph with k_o or k_e is pretty empirical. Hardware resource constraint is a viable standard as stated above. Another feasible standard is based on the sparsity of heterogeneous graph data, because the performance of randomization relies on data sparsity and practical heterogeneous graphs may have pretty different data distributions. When perturbing sparse graph data, the aggregation and estimation are usually inaccurate, so the optimal k_o-PrivHG is picked to improve the estimation accuracy. When perturbing dense graph data, diminution of sampling size with k_e-PrivHG is reasonable, which reduces communication overhead and utilizes privacy budget as a whole. The general principle is that deploying k_o-PrivHG on small and sparse graph data and picking k_e-PrivHG otherwise. In the experiment section, We reasonably pick these two mechanisms for different datasets, and leave the fine-grained contextual-dependent selection of k-PrivHG for heterogeneous graph as future work.

Fake degree generation. Due to the intrinsic correlation within heterogeneous graph data, the perturbation of attributes and degrees should remain correlated, otherwise the information loss results in inaccurate estimates. PrivHG ensures that degree randomization follows the result of attribute randomization. Specifically, there are four possible cases for randomizing indicating bit of merged bins $\bar{v}_b \to \tilde{v}_b$: $1 \to 0, 1 \to 1, 0 \to 0, 0 \to 1$. When an indicating bit of merged bin is perturbed to $\tilde{v}_b = 0$ ($\bar{v}_b = 1$ or $\bar{v}_b = 0$), the corresponding aggregated binning degree $deg(b)$ should be set as 0 (equivalent to set degree vector $\tilde{u}^b = [0, \ldots, 0]$) regardless of perturbed degree value, otherwise the correlation between them will be violated. When $\bar{v}_b = 1$ is randomized as $\tilde{v}_b = 1$, degree bit $\tilde{u}_{deg(b)}$ is normally randomized and retained (The corresponding randomization in Algorithm 4 is achieved by multiplying two randomized vectors $\tilde{u}^b = \tilde{u}^b \cdot \tilde{v}_b$). In the case of $\bar{v}_b = 0$ and $\tilde{v}_b = 1$, the corresponding aggregated binning degree needs to satisfy $deg(b) \neq 0$, but local user has no related degree data to be randomized, therefore PrivAG randomly generates a fake degree from $[1, \theta]$ as $deg(b)$, which skews the estimation of attributed degree distribution. With the help of Algorithm 3, the fake degree generation is further refined in PrivHG, therefore neutralizing the skewing effect on the estimation. For $\bar{v}_b = 0$ being randomized as $\tilde{v}_b = 1$, the generation range of fake degree is scaled to $[min(deg(b)), max(deg(b))]$ instead of $[1, \theta]$ to prevent outliers being generated, and the generation probability is set to the estimated

frequency of degrees ψ_j^t instead of equal probabilities $\frac{1}{\theta}$ to reduce the skewness. $deg(b)$ and ψ_j^t can be inferred from the postprocessed statistics of ABS without violating ϵ-ALDP.

5.3. Estimation Optimization

This subsection corresponds to the last step in Figure 2 and provides optimization techniques for server-side aggregation and estimation. Similar to Algorithm 1, PrivHG aggregates bit 1s in perturbed attribute and degree vectors, and makes an unbiased estimation based on the aggregation. Since the estimated statistics should follow the characteristic of heterogeneous graph data, two postprocessing approaches are further proposed in PrivHG to filter out the aggregated outliers and correct the final statistical estimation.

Attribute Bin Frequency Estimation. The probabilities of Equations (3) and (4) are critical to make an unbiased estimation of attribute distribution ϕ^a. Since ABS resizes the domain size through aggregating attributes into bins, then the two binning randomization probabilities of $\hat{p}_b = Pr[\hat{v}_b = 1|\bar{v}_b = 1]$, and $\hat{q}_b = Pr[\hat{v}_b = 1|\bar{v}_b = 0]$ are derived as follows.

$$\begin{cases} \hat{p}_b = \dfrac{exp(\epsilon_1)\binom{r+\ell'-1}{k-1}}{\binom{r}{k} + exp(\epsilon_1)(\binom{r+\ell'}{k} - \binom{r}{k})} \\ \hat{q}_b = \dfrac{\binom{r-1}{k-1} + exp(\epsilon_1)(\binom{r+\ell'-1}{k-1} - \binom{r-1}{k-1})}{\binom{r}{k} + exp(\epsilon_1)(\binom{r+\ell'}{k} - \binom{r}{k})} \end{cases}$$

Based on the above probabilities, the expected counts of aggregated bins $\hat{C}_b$ is denoted as:

$$\mathbb{E}[\hat{c}_b] = \mathbb{E}[\#\{i|\hat{v}_b^i = 1, i \in [1,n], b \in [1,r]\}] = \phi^b n\hat{p}_b + (1-\phi^b)n\hat{q}_b \tag{13}$$

Then unbiased estimation of attribute bin frequency is:

$$\phi^b = \frac{\hat{c}_b - n\hat{q}_b}{n(\hat{p}_b - \hat{q}_b)} \tag{14}$$

The common way to optimize frequency estimation like ϕ^b is to clip it with range $[0,1]$. In PrivHG, a better lower bound is given based on the characteristic of heterogeneous graph. Assume an extreme case, where there is only one edge e_{xy} corresponding to a merged bin b in the whole heterogeneous graph, then at least two nodes x and y report attribute data $\hat{v}_b^x = 1$ and $\hat{v}_b^y = 1$, and the least aggregated bits count $\hat{c}_b$ for each merged bin b is 2, therefore the lower bound of ϕ^b should be $\frac{2}{n}$. The estimation of attribute distribution $\hat{\phi}^b$ is derived by clipping ϕ^b with range $[\frac{2}{n}, 1]$.

Binned Degree Frequency Estimation. Similar to the estimation in Section 4, binned degrees are estimated based on the aggregated bins in B_o. The expected counts of binned degree $\hat{d}_b^t$ is derived as follows, where $\hat{p}_d = \frac{1}{2}$ and $\hat{q}_d = \frac{1}{exp(\epsilon_2/k)+1}$

$$\mathbb{E}[\hat{d}_b^t] = \mathbb{E}[\#\{i|\hat{u}_b^t(i) = 1\}] = \psi_b^t n\phi^b\hat{p}_d + (1-\psi_b^t)n\phi^b\hat{q}_d \tag{15}$$

Then unbiased estimation of binned degree frequency is:

$$\psi_b^t = \frac{\hat{d}_b^t - n\phi^b\hat{q}_d}{n\phi^b(\hat{p}_d - \hat{q}_d)} \tag{16}$$

Since the attributed degree distribution cannot be negative, the $\hat{\psi}_b^t$ is first clipped with $[0,1]$ for each bin in $B = [b_1, \ldots, b_r]$ to eliminate negative influences of outliers. Then the estimations are further corrected based on the nature of graph data. Considering the one characteristic of graph edges that the total number of edges have an upper bound $\frac{n(n-1)}{2}$, which is the edge number of complete graph with n nodes. Similarly in the context of PrivHG, the maximum degree is truncated as θ for each bin in B, therefore the total number

of edges cannot exceed that of θ-complete graph with n_j nodes, where n_j is the number of binned nodes with $b_j \in B$ and can be derived by corresponding estimated attribute bin frequency as $n\phi^b$, then the upper bound of total edges is $\frac{n\phi^b\theta}{2}$. Since the lower bound of total edges is 1, the total edges $\sum_{t\in[1,\theta]} tn_{bt}\psi_b^t$ for each bin $b \in B$ is bounded as:

$$1 \leq \frac{\sum_{t\in[1,\theta]} tn_{bt}\psi_b^t}{2} \leq \frac{n\phi^b\theta}{2} \tag{17}$$

Given that $2 < \theta$, the refined estimation of binned degree frequency is derived by substituting Equation (17) into (16):

$$\hat{\psi}_b^t = \frac{\hat{d}_b^t - n\phi^b\hat{q}_d}{tn\phi^b(\hat{p}_d - \hat{q}_d)} \cdot max(\frac{2}{\sum_{t\in[1,\theta]}\psi_b^t t}, 1) \cdot min(\frac{\theta}{\sum_{t\in[1,\theta]}\psi_b^t t}, 1) \tag{18}$$

5.4. PrivHG Mechanism

In this subsection, we present the overall PrivHG mechanism based on aforementioned building blocks. The detailed pseudocode of PrivHG is listed in Algorithm 4.

Algorithm 4 PrivHG.

Input: local heterogeneous graphs G, privacy budget ϵ.
Output: attribute frequency $\hat{\phi}^a$, attribute-degree distribution $\hat{\psi}^d$.

```
//user-side randomization with basic binning
B_u = {b_i = (1 + (i − 1)δ, 1 + iδ) | i ∈ (1, r), δ = (w − 1)/r}
Ḡ′ ← pre − process(G′, B_u)
φ^a, ψ^d ← PrivAG(Ḡ′)
//server-side optimal binning scheme selection
B′_u ← B_u ∪ A_c
B_o ← ABS(Ḡ′, φ^a, ψ^d, ε_2/k, B′_u)
redistribute parameters B_o, k_e or k_o
//user-side randomization with optimal binning
Ḡ″ ← pre − process(G″, B_o)
v̂_b, û_b^t ← DRC(Ḡ″)
//server-side estimation with correction
c_j ← count(v̂_b)
d_b^t ← count(û_b^t)
for b ∈ [1, r] do
    estimate attribute bin frequency: φ^b = (ĉ_b − nq̂_b) / (n(p̂_b − q̂_b))
    clip φ^b with [2/n, 1]
end for
for b ∈ [1, r] and t ∈ [1, θ] do
    estimate binned degree frequency with refinement as:
    ψ̂_b^t = (d̂_b^t − nφ̂^b q̂_d) / (tnφ̂^b(p̂_d − q̂_d)) · min(θ / Σ_{t∈[1,θ]} ψ_b^t t, 1)
    clip ψ̂_b^t with [1/(nφ̂^b), 1]
end for
return φ̂^a and ψ̂^d
```

To elaborate, PrivHG mechanism first generalizes Algorithm 3 as a fundamental subtask to deal with mixed-attributed data in local graph. There are two conventional approaches regarding the execution of subtasks, one is to divide the privacy budget ϵ as several parts for each subtask to execute on the complete data set, and the other is to divide the user data while each subtask consuming the complete privacy budget ϵ to execute on a portion of data set. The former approach of dividing privacy budget ϵ leads to inaccurate estimations especially for heterogeneous graph. In contrast, executing subtasks separately

on divided data sets utilizes the full privacy budget and comparatively reduces the overall error, which has been adopted by several recent studies and is also applied in our PrivHG mechanism. The heterogeneous graph G is divided as two groups G' and G'', where ABS subtask is executed on G' to derive the optimized binning B_o and B_o is employed on G'' to make an unbiased estimation.

During the initialization phase of PrivHG, numerical-attributed data of G' is divided by a basic uniform binning B_u (other schemes like geometric binning is alternative), and each interval is treated equally as the categorical attributes. Based on B_u, Algorithm 2 aggregates statistics of preprocessed G'. Then on the server side, ABS outputs generalized optimal binning scheme B_o for mixed attributes by enlarging the input domain as $B_u \cup A_c$, which is the union set of numerical-attributed and categorical-attributed data. Generalized ABS makes no assumption about the attribute type, and solely optimizes Equation (12) based on the degrees of each merged bin $b_i \in B'_u$. In order to further mitigate the influences brought by noisy degree outliers, we choose to remove 5% marginal data when practically executing ABS in this paper. On the one hand, these marginal values may be biased outliers that are randomly generated from vacant vector bits, and the estimation error will be reduced if removing these outliers. On the other hand, even actual marginal data may account for a relatively small proportion of whole data due to the data distribution of graph, which have a minor impact on the results. In the following phase, the optimal binning scheme B_o and necessary parameters are informed to the other subset of heterogeneous graph G'', and each user preprocesses and randomizes local data as Algorithm 1, in which fake degree generation is calibrated as stated in Section 5.2 instead of randomly selected. Finally, these randomized vectors are aggregated by the server, then unbiased estimations with correction and refinement are made according to Section 5.3.

According to composition and postprocessing theorems, aggregating the two statistics of heterogeneous graph under PrivHG mechanism satisfies ϵ-attributewise local differential privacy, and proof of which is omitted due to the triviality.

6. Experimental Evaluation

In this section, we evaluate the estimation performance of proposed PrivHG and comparison mechanisms on extensive scenarios.

Evaluated Mechanisms. For attribute and degree distribution estimation on categorical-attributed graph, we utilize the generalized randomized response (GRR) mechanism to perturb local data as in [36,37], which is compared with PrivAG and PrivHG (PrivHG executes ABS solely on categorical-attributed data). For estimation on heterogeneous graph with mixed-attributed data, we combine GRR and basic binning scheme (both uniform binning B_u and geometric binning B_g schemes) to uniformly perturb heterogeneous data, which is denoted as BGRR, and PrivAG is also tentatively extended to heterogeneous graph with basic binning scheme. These two mechanisms are compared with PrivHG.

General Setting. The experiments are implemented on various synthetic Erdos–Renyi random graphs [38], which gives a general simulation about the real-world datasets. To be specific, we separately generate graphs with different attributes based on parameter m, w and n, and merge them together as a heterogeneous graph in each experiment epoch, the number of users/nods n is set to 5000, the categorical attribute domain size m ranges from 8 to 32, the numerical attribute range bound w varies from 10 to 20. To simulate different attributed data sparsity of heterogeneous graphs, the maximum number of synthetic edges for each attribute follows the Uniform/Gaussian distribution ($\mu = 0$ and $\sigma = 10$). During the data preprocess part, truncation parameter θ range from 10 to 50, and the privacy budget ϵ ranges from 0.005 to 5.0, with $\epsilon_1 = \epsilon_2 = \epsilon/2$. Each setting of the experiments runs 100 times, and the result are average of these experiments.

Performance Metrics. The performance of attribute and degree distribution estimation is evaluated by MSE (ℓ_2-norm error):

$$\|\hat{\phi}^a - \phi^a\|_2 = \mathbb{E}[\sqrt{\|\hat{\phi}^a - \phi^a\|^2}],\ \|\hat{\psi}^d - \psi^d\|_2 = \mathbb{E}[\sqrt{\|\hat{\psi}^d - \psi^d\|^2}] \tag{19}$$

where ϕ^a and ψ^d (resp. $\hat{\phi}^a$ and $\hat{\psi}^d$) are the true distribution of attributes and attribute-degrees (resp. estimated).

Influence of categorical attribute domain size. Figure 3 shows the estimation error of categorical-attributed graph aggregation, with different categorical attribute domain size m and privacy budget ϵ settings. It can be observed from the figures that the estimation error reduction grows larger as the domain size m increases, and PrivHG is less affected by domain size than other two mechanisms. In most settings, PrivHG outperforms GRR and PrivAG on both attribute frequency and attribute-degree distribution estimation.

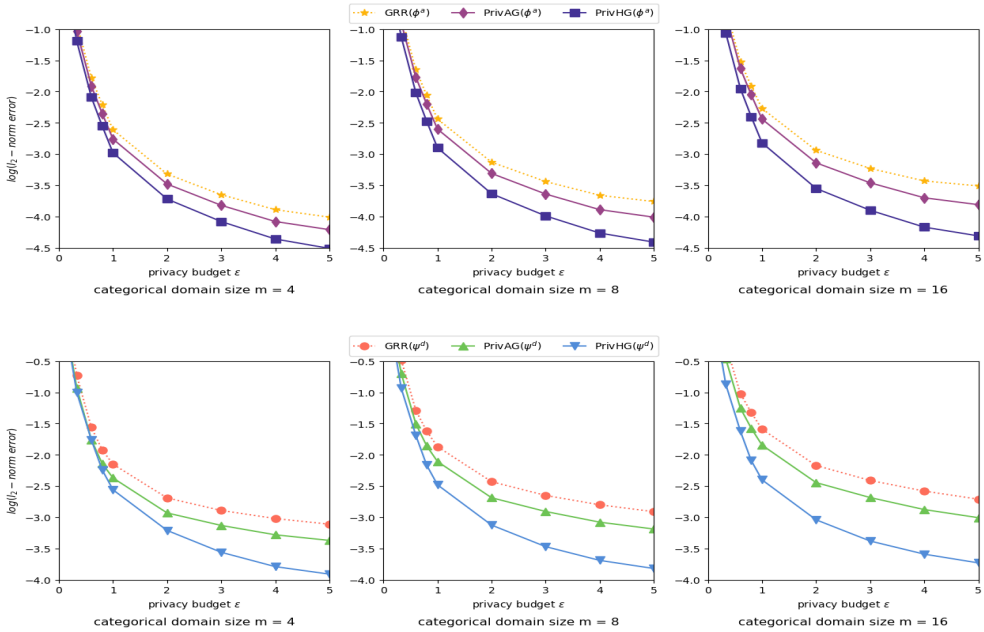


Figure 3. Categorical-attributed graph aggregation with different attribute domain size.

Influence of numerical attribute domain range. Figure 4 shows the results of heterogeneous graph aggregation with varied numerical attribute domain w and fixed categorical attribute domain size $m = 32$, where BGRR and PrivAG apply uniform binning scheme to deal with numerical-attributed data. PrivHG outperforms BGRR and PrivAG in most settings. As w increases, the reduction of attribute frequency estimation error among three mechanisms is minor, while the degree distribution estimation error of PrivHG decreases faster than other two mechanisms.

Influence of truncation parameter. Figure 5 shows the results of heterogeneous graph aggregation with different truncation parameter θ and privacy budget ϵ settings. When θ grows larger, the degree estimation accuracy of BGRR and PrivAG degrades a lot due to excessive vacant bits being randomized as noises, but results of PrivHG have a relatively significant improvement. The error reduction of attribute estimation is slightly affected by truncation parameter θ between PrivHG and other mechanisms. In most cases, PrivHG outperforms BGRR and PrivAG on distribution estimation.

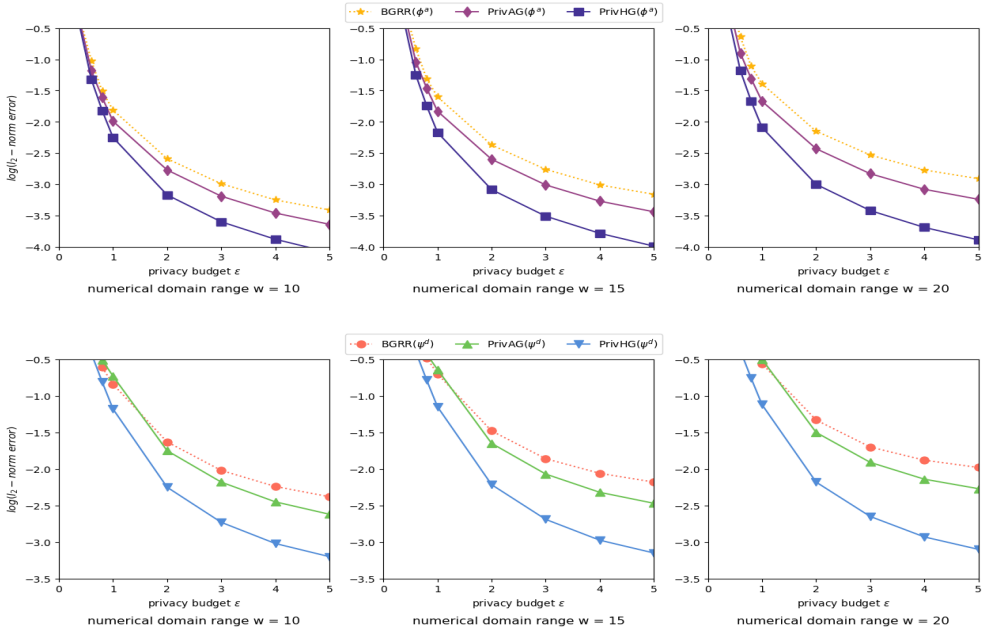


Figure 4. Heterogeneous graph aggregation with different numerical attribute range.

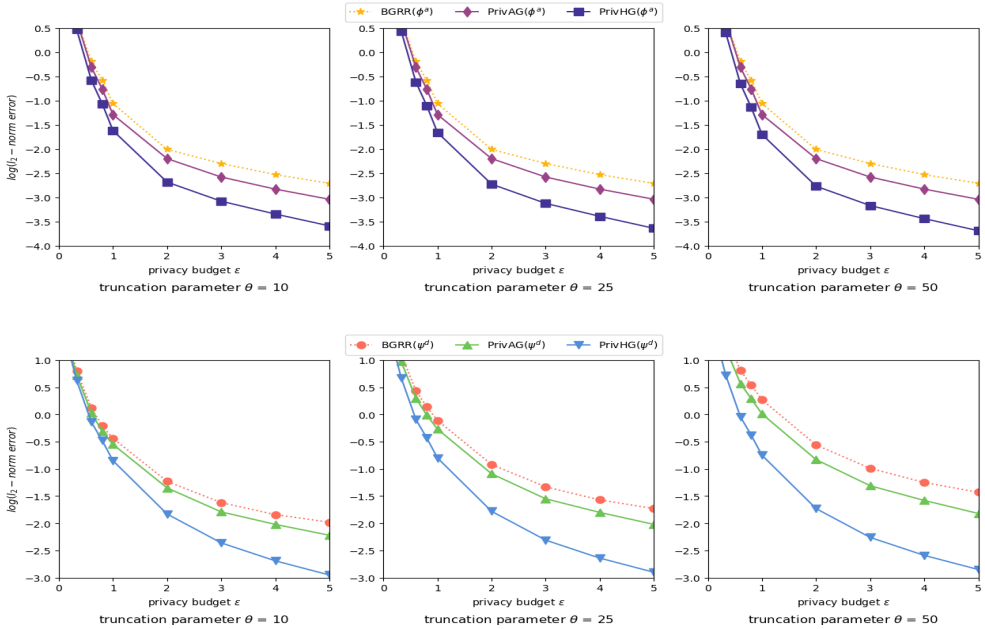


Figure 5. Heterogeneous graph aggregation with different truncation parameter.

Influence of data distribution and binning scheme. Figure 6 shows the results of heterogeneous graph aggregation with different data distribution/sparsity and various binning schemes. As can be summarized from these figures, the estimation error reduction between PrivHG and other two mechanisms is rather noticeable when the data distribution and binning scheme are dissimilar, which could be due to the reliance of BGRR and PrivAG on the consistency of intrinsic graph data distribution and binning scheme. In most settings, PrivHG is stable and outperforms BGRR and PrivAG on both attribute frequency and attribute-degree distribution estimation.

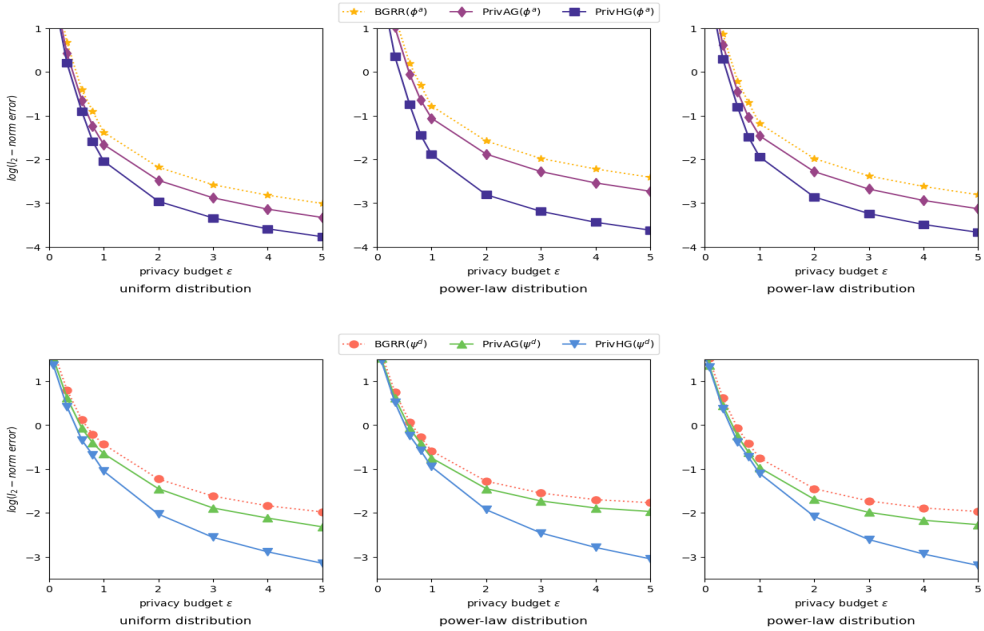


Figure 6. Heterogeneous graph aggregation with uniform binning and Uniform distribution (**left**), uniform binning and Gaussian distribution (**middle**), geometric binning and Gaussian distribution (**right**).

In summary, above experiments show that it is feasible to preserve privacy for heterogeneous graph data under ϵ-ALDP with high fidelity, and PrivHG mechanism significantly outperforms baseline mechanisms on statistical results by reducing 43% estimation error in average. Furthermore, the proposed PrivHG mechanism is well suited to deal with various heterogeneous graphs and does not rely on specific data sparsity or attribute binning scheme.

7. Related Work

The de facto Differential Privacy (DP) notion have form the theoretical basis of a considerable amount of research literature for past decade. By assuming a centralized and trustworthy data curator [15,39], several fundamental mechanisms achieving differential privacy constraint have been proposed to deal with numerical and categorical data, including Laplace mechanism in [6] and Exponential mechanism in [7]. However, under the gradually increasing risk of adversaries prying into personal privacy and the growing expectation to keep private data on personal devices, the emphasis of privacy-preservation studies has shifted from centralized settings to local settings.

Local Differential Privacy [40] ensures that private data are perturbed locally on each user's devices, thus avoiding the reliance on trustworthiness of data curator and broadening the applicable scenario of DP. A variety of studies protecting local differential privacy have been constantly emerging. The pioneer study of Randomized Response, which was proposed by [33], satisfies local differential privacy guarantee well, and many following studies are built on it. Its variants play an important role in the categorical data domain [8,11,14,41–43]. Later on, the study of LDP is expanded to more promising fields. Ref. [34] summarizes the characteristics of existing mechanisms and proposes OUE and OLH to better adapt to various novel scenarios. Ref. [20] presents a two-phase framework for aggregating set-valued data under local differential privacy, and [19] proposes a generalized mechanism PrivSet to perturb a sampled subset of set-valued data domain and provides optimized estimation guarantee. As for numerical data, Ref. [44] utilizes square wave and smoothing mechanism to maximum the estimation expectation of numerical data distribution, and [45] proposes an adaptive hierarchy-based mechanism to privately answer range query. Beyond the single datatype, Ref. [24] designs an iterative

mechanism PrivKVM to locally privately collect key-valued data, and retain the correlation between key-value pairs. Ref. [25] optimizes the estimation accuracy and communication cost of PrivKVM mechanism. These studies offer powerful tools for tackling our problem.

Due to its intrinsic complexity, preserving private graph requires additional concerns. According to the variations of privacy granularity, differential privacy for graph data can be generally divided into two groups [46]: node-based and edge-based, which provide protection either on edge-level privacy or on node-level privacy. Based on different privacy granularity, various problems are studied, such as publishing private degree frequency [47,48], aggregating graphic statistics [49,50] and synthetic graph generation [51,52]. Recently, graph data aggregation mechanisms under LDP constraint have been studied. Ref. [53] manages to aggregate node degrees and weighted edges based on 1-neighborhood graph in the local setting. By defining neighboring clusters, collecting neighboring degrees and refining the clusters, Ref. [30] proposes an iterative graph generation framework LDP-Gen to generate synthetic graphs. Ref. [31] introduces a novel privacy notion DDP for social networks, and provides a multiphased framework to aggregate subgraph statistics. Ref. [32] presents a graph generative framework AsgLDP, capturing node features and generating node-attributed graph. Ref. [37] extends the research fields to multiplex graphs and proposes to locally privately estimate clustering coefficients on them. However, these research studies of preserving local private graph data mainly focus on edge-based LDP for graph and neither of them provides stronger privacy guarantee while aggregating heterogeneous graph data.

8. Conclusions

In this paper, we study the heterogeneous graph aggregation with a unified, efficient and effective PrivHG mechanism under local differential privacy. We combine characteristics of two conventional LDP variants and propose a fine-grained privacy definition for locally private heterogeneous graph, which generally provides stronger privacy guarantee than edge-based LDP and higher estimation accuracy than node-based LDP. We design a unified mechanism PrivHG to aggregate two statistics of heterogeneous graph while protecting the fine-grained attributewise local differential privacy. Furthermore, we propose several optimization techniques for reducing the computation costs and estimation errors of PrivHG mechanism in practical application. The effectiveness and efficiency of the PrivHG mechanism are validated through extensive experiments.

We will investigate the application of PrivHG with other graph analysis tasks and extend the perturbation mechanisms for other correlated and heterogeneous data types for future work.

Author Contributions: Conceptualization, Z.L.; methodology, Z.L.; software, Z.L.; writing—original draft preparation, Z.L.; writing—review and editing, L.H., H.X. and W.Y.; visualization, Z.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: This paper is an extended version of our conference paper [54] entitled "PrivAG: Analyzing Attributed Graph Data with Local Differential Privacy" in the 26th IEEE International Conference on Parallel and Distributed Systems (ICPADS 2020).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Marketing Firm Exactis Leaked a Personal Info Database with 340 Million Records. 2018. Available online: https://www.wired.com/story/exactis-database-leak-340-million-records/ (accessed on 12 January 2021).
2. Facebook Security Breach Exposes Accounts of 50 Million Users. 2018. Available online: https://www.nytimes.com/2018/09/28/technology/facebook-hack-data-breach.html (accessed on 12 January 2021).
3. Marriott Hacking Exposes Data of Up to 500 Million Guests. 2018. Available online: https://www.nytimes.com/2018/11/30/business/marriott-data-breach.html (accessed on 12 January 2021).
4. Voigt, P.; Von dem Bussche, A. The eu general data protection regulation (gdpr). In *A Practical Guide*, 1st ed.; Springer International Publishing: Cham, Switzerland, 2017; Volume 10, pp. 10–5555.
5. Goldman, E. An Introduction to the California Consumer Privacy Act (CCPA). Santa Clara Univ. Legal Studies Research Paper 2020. Available online: https://ssrn.com/abstract=3211013 (accessed on 12 January 2021).
6. Dwork, C.; McSherry, F.; Nissim, K.; Smith, A.D. Calibrating Noise to Sensitivity in Private Data Analysis. In Proceedings of the TCC, New York, NY, USA, 4–7 March 2006.
7. McSherry, F.; Talwar, K. Mechanism Design via Differential Privacy. In Proceedings of the 48th Annual IEEE Symposium on Foundations of Computer Science (FOCS'07), Providence, RI, USA, 21–23 October 2007; pp. 94–103.
8. Duchi, J.C.; Jordan, M.I.; Wainwright, M.J. Local privacy and statistical minimax rates. In Proceedings of the 2013 51st Annual Allerton Conference on Communication, Control, and Computing (Allerton), Monticello, IL, USA, 2–4 October 2013; p. 1592.
9. Learning with Privacy at Scale. 2017. Available online: https://machinelearning.apple.com/research/learning-with-privacy-at-scale (accessed on 12 January 2021).
10. Tang, J.; Korolova, A.; Bai, X.; Wang, X.; Wang, X. Privacy loss in apple's implementation of differential privacy on macos 10.12. *arXiv* **2017**, arXiv:1709.02753.
11. Erlingsson, Ú.; Korolova, A.; Pihur, V. RAPPOR: Randomized Aggregatable Privacy-Preserving Ordinal Response. *arXiv* **2014**, arXiv:1407.6981.
12. Fanti, G.; Pihur, V.; Erlingsson, Ú. Building a RAPPOR with the unknown: Privacy-preserving learning of associations and data dictionaries. *arXiv* **2015**, arXiv:1503.01214.
13. Ding, B.; Kulkarni, J.; Yekhanin, S. Collecting telemetry data privately. *Adv. Neural Inf. Process. Syst.* **2017**, *30*, 3574–3583.
14. Kairouz, P.; Bonawitz, K.; Ramage, D. Discrete distribution estimation under local privacy. In Proceedings of the International Conference on Machine Learning, PMLR, New York, NY, USA, 20–22 June 2016; pp. 2436–2444.
15. Li, C.; Hay, M.; Miklau, G.; Wang, Y. A data-and workload-aware algorithm for range queries under differential privacy. *arXiv* **2014**, arXiv:1410.0265.
16. Kairouz, P.; Oh, S.; Viswanath, P. Extremal mechanisms for local differential privacy. *Adv. Neural Inf. Process. Syst.* **2014**, *27*, 492–542.
17. Duchi, J.C.; Jordan, M.I.; Wainwright, M.J. Minimax optimal procedures for locally private estimation. *J. Am. Stat. Assoc.* **2018**, *113*, 182–201. [CrossRef]
18. Nguyên, T.T.; Xiao, X.; Yang, Y.; Hui, S.C.; Shin, H.; Shin, J. Collecting and analyzing data from smart device users with local differential privacy. *arXiv* **2016**, arXiv:1606.05053.
19. Wang, S.; Huang, L.; Nie, Y.; Wang, P.; Xu, H.; Yang, W. PrivSet: Set-Valued Data Analyses with Locale Differential Privacy. In Proceedings of the IEEE INFOCOM 2018—IEEE Conference on Computer Communications, Honolulu, HI, USA, 15–19 April 2018; pp. 1088–1096.
20. Qin, Z.; Yang, Y.; Yu, T.; Khalil, I.M.; Xiao, X.; Ren, K. Heavy Hitter Estimation over Set-Valued Data with Local Differential Privacy. In Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, Vienna, Austria, 24–28 October 2016.
21. Wang, T.; Li, N.; Jha, S. Locally Differentially Private Frequent Itemset Mining. In Proceedings of the 2018 IEEE Symposium on Security and Privacy (SP), San Francisco, CA, USA, 20–24 May 2018; pp. 127–143.
22. Ren, X.; Yu, C.M.; Yu, W.; Yang, S.; Yang, X.; McCann, J.A.; Philip, S.Y. LoPub: High-dimensional crowdsourced data publication with local differential privacy. *IEEE Trans. Inf. Forensics Secur.* **2018**, *13*, 2151–2166. [CrossRef]
23. Wang, N.; Xiao, X.; Yang, Y.; Zhao, J.; Hui, S.C.; Shin, H.; Shin, J.; Yu, G. Collecting and analyzing multidimensional data with local differential privacy. In Proceedings of the 2019 IEEE 35th International Conference on Data Engineering (ICDE), Macao, China, 8–11 April 2019; pp. 638–649.
24. Ye, Q.; Hu, H.; Meng, X.; Zheng, H. PrivKV: Key-Value Data Collection with Local Differential Privacy. In Proceedings of the 2019 IEEE Symposium on Security and Privacy (SP), San Francisco, CA, USA, 19–23 May 2019; pp. 317–331.
25. Gu, X.; Li, M.; Cheng, Y.; Xiong, L.; Cao, Y. {PCKV}: Locally Differentially Private Correlated {Key-Value} Data Collection with Optimized Utility. In Proceedings of the 29th USENIX Security Symposium (USENIX Security 20), Boston, MA, USA, 12–14 August 2020; pp. 967–984.
26. Yang, J.; Leskovec, J. Defining and evaluating network communities based on ground-truth. *Knowl. Inf. Syst.* **2015**, *42*, 181–213. [CrossRef]
27. Wang, Z.; Liao, J.; Cao, Q.; Qi, H.; Wang, Z. Friendbook: A Semantic-Based Friend Recommendation System for Social Networks. *IEEE Trans. Mob. Comput.* **2015**, *14*, 538–551. [CrossRef]

28. Chen, W.; Wang, C.; Wang, Y. Scalable influence maximization for prevalent viral marketing in large-scale social networks. In Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Washington, DC, USA, 24–28 July 2010.
29. Al-garadi, M.; Khan, M.; Varathan, K.D.; Mujtaba, G.; Al-Kabsi, A.M. Using online social networks to track a pandemic: A systematic review. *J. Biomed. Inform.* **2016**, *62*, 1–11. [CrossRef] [PubMed]
30. Qin, Z.; Yu, T.; Yang, Y.; Khalil, I.M.; Xiao, X.; Ren, K. Generating Synthetic Decentralized Social Graphs with Local Differential Privacy. In Proceedings of the ACM Conference on Computer and Communications Security, Dallas, TX, USA, 30 October–3 November 2017.
31. Sun, H.; Xiao, X.; Khalil, I.; Yang, Y.; Qin, Z.; Wang, H.; Yu, T. Analyzing subgraph statistics from extended local views with decentralized differential privacy. In Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security, London, UK, 11–15 November 2019; pp. 703–717.
32. Wei, C.; Ji, S.; Liu, C.; Chen, W.; Wang, T. AsgLDP: Collecting and Generating Decentralized Attributed Graphs With Local Differential Privacy. *IEEE Trans. Inf. Forensics Secur.* **2020**, *15*, 3239–3254. [CrossRef]
33. Warner, S. Randomized response: A survey technique for eliminating evasive answer bias. *J. Am. Stat. Assoc.* **1965**, *60*, 63–69. [CrossRef] [PubMed]
34. Wang, T.; Blocki, J.; Li, N.; Jha, S. Locally Differentially Private Protocols for Frequency Estimation. In Proceedings of the USENIX Security Symposium, Vancouver, BC, Canada, 16–18 August 2017.
35. Zhang, Z.; Wang, T.; Li, N.; He, S.; Chen, J. CALM: Consistent Adaptive Local Marginal for Marginal Release under Local Differential Privacy. In Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security, Toronto, ON, Canada, 15–19 October 2018.
36. Ye, Q.; Hu, H.; Au, M.; Meng, X.; Xiao, X. Towards Locally Differentially Private Generic Graph Metric Estimation. In Proceedings of the 2020 IEEE 36th International Conference on Data Engineering (ICDE), Dallas, TX, USA, 20–24 April 2020; pp. 1922–1925.
37. Liu, Z.; Xu, H.; Huang, L.; Yang, W. Estimating Clustering Coefficient of Multiplex Graphs with Local Differential Privacy. In Proceedings of the WASA, Nanjing, China, 25–27 June 2021.
38. Gilbert, E.N. Random Graphs. *Ann. Math. Statist.* **1959**, *30*, 1141–1144. [CrossRef]
39. Johnson, N.; Near, J.P.; Song, D. Towards practical differential privacy for SQL queries. *Proc. Vldb Endow.* **2018**, *11*, 526–539. [CrossRef]
40. Raskhodnikova, S.; Smith, A.; Lee, H.K.; Nissim, K.; Kasiviswanathan, S.P. What can we learn privately. In Proceedings of the 54th Annual Symposium on Foundations of Computer Science, Berkeley, CA, USA, 26–29 October 2013; pp. 531–540.
41. Pastore, A.; Gastpar, M. Locally differentially-private distribution estimation. In Proceedings of the 2016 IEEE International Symposium on Information Theory (ISIT), Barcelona, Spain, 10–15 July 2016; pp. 2694–2698.
42. Bassily, R.; Smith, A. Local, private, efficient protocols for succinct histograms. In Proceedings of the Forty-Seventh Annual ACM Symposium on Theory of Computing, Portland, OR, USA, 14–17 June 2015; pp. 127–135.
43. Chen, R.; Li, H.; Qin, A.K.; Kasiviswanathan, S.P.; Jin, H. Private spatial data aggregation in the local setting. In Proceedings of the 2016 IEEE 32nd International Conference on Data Engineering (ICDE), Helsinki, Finland, 16–20 May 2016; pp. 289–300.
44. Li, Z.; Wang, T.; Lopuhaä-Zwakenberg, M.; Li, N.; Škoric, B. Estimating numerical distributions under local differential privacy. In Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data, Portland, OR, USA, 14–19 June 2020; pp. 621–635.
45. Du, L.; Zhang, Z.; Bai, S.; Liu, C.; Ji, S.; Cheng, P.; Chen, J. AHEAD: Adaptive Hierarchical Decomposition for Range Query under Local Differential Privacy. In Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security, Virtual, 15–19 November 2021; pp. 1266–1288.
46. Hay, M.; Li, C.; Miklau, G.; Jensen, D.D. Accurate Estimation of the Degree Distribution of Private Networks. In Proceedings of the 2009 Ninth IEEE International Conference on Data Mining, Miami, FL, USA, 6–9 December 2009; pp. 169–178.
47. Kasiviswanathan, S.P.; Nissim, K.; Raskhodnikova, S.; Smith, A.D. Analyzing Graphs with Node Differential Privacy. In Proceedings of the TCC, Tokyo, Japan, 3–6 March 2013.
48. Day, W.Y.; Li, N.; Lyu, M. Publishing Graph Degree Distribution with Node Differential Privacy. In Proceedings of the SIGMOD Conference, San Francisco, CA, USA, 26 June–1 July 2016.
49. Zhang, J.; Cormode, G.; Procopiuc, C.M.; Srivastava, D.; Xiao, X. Private release of graph statistics using ladder functions. In Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data, Victoria, Australia, 31 May–4 June 2015; pp. 731–745.
50. Raskhodnikova, S.; Smith, A. Lipschitz extensions for node-private graph statistics and the generalized exponential mechanism. In Proceedings of the 2016 IEEE 57th Annual Symposium on Foundations of Computer Science (FOCS), New Brunswick, NJ, USA, 9–11 October 2016; pp. 495–504.
51. Leskovec, J.; Chakrabarti, D.; Kleinberg, J.M.; Faloutsos, C.; Ghahramani, Z. Kronecker Graphs: An Approach to Modeling Networks. *J. Mach. Learn. Res.* **2008**, *11*, 985–1042.
52. Lu, W.; Miklau, G. Exponential random graph estimation under differential privacy. In Proceedings of the KDD, New York, NY, USA, 24–27 August 2014.

53. Liu, Q.; Wang, G.; Li, F.; Yang, S.; Wu, J. Preserving privacy with probabilistic indistinguishability in weighted social networks. *IEEE Trans. Parallel Distrib. Syst.* **2016**, *28*, 1417–1429. [CrossRef]
54. Liu, Z.; Huang, L.; Xu, H.; Yang, W.; Wang, S. PrivAG: Analyzing attributed graph data with local differential privacy. In Proceedings of the 2020 IEEE 26th International Conference on Parallel and Distributed Systems (ICPADS), Hong Kong, China, 2–4 December 2020; pp. 422–429.