

Dual-Granularity Local Differential Privacy for Secure Distributed Graph Publishing in Decentralized Network Systems

Zhihua Chang^{1,*}

¹Information Technology Center, Zhejiang University, Hangzhou 310058, China

Abstract

Decentralized social, interaction, and edge-assisted graphs are commonly stored as local neighbor views rather than as a trusted centralized graph, making privacy-preserving collection and useful graph publication difficult. This paper presents DDP-DGP, an integrative dual-granularity local differential privacy framework for undirected and unweighted graphs. Each user perturbs coarse-grained degree information with a full-support, boundary-corrected random-jump mechanism and perturbs fine-grained neighbor indicators with bitwise randomized response. The fine-grained budget is explicitly divided across the $|V|-1$ indicators, providing a whole-neighbor-list guarantee by sequential composition; a degree-calibrated sampling rule then suppresses randomized-response densification. The publisher uses only privatized reports: inconsistent endpoint reports are reconciled probabilistically, communities are extracted from the privatized adjacency matrix, intra-community edges are generated with clipped Chung-Lu probabilities, and inter-community edges are sampled from counts computed exclusively from privatized data. The contribution is therefore positioned as a privacy-preserving integration and calibration strategy rather than a new differential-privacy primitive. Experiments on four public graph datasets evaluate degree distribution, modularity, eigenvector centrality, and average path length. The displayed results indicate improved structural fidelity for DDP-DGP under the tested privacy budgets.

Keywords: local differential privacy, decentralized graph publishing, dual-granularity privacy protection, synthetic graph generation

Received on 29 June 2026, accepted on 08 September 2026, published on 07 October 2026

Copyright © 2026 Zhihua Chang, licensed to EAI. This is an open access article distributed under the terms of the [CC BY-NC-SA 4.0](#), which permits copying, redistributing, remixing, transformation, and building upon the material in any medium so long as the original work is properly cited.

doi: 10.4108/eetsis.13779

1. Introduction

With the evolution of distributed social media, cloud-edge collaboration, decentralized recommendation systems, and cross-domain data-sharing platforms, graph-structured data are increasingly generated and stored across distributed nodes rather than centralized servers. Graphs represent entities as nodes and complex relationships as edges, but unprotected local views can disclose communication links, transaction relations, friendship ties,

and service interactions. Privacy-preserving decentralized online social-network frameworks have shown the importance of reducing risks associated with centralized data control [1]. In distributed graph publishing, each node typically maintains only its own information and direct-neighbor connections, while the publisher may be untrusted. Therefore, in addition to complying with data-protection requirements such as Regulation (EU) 2016/679 (GDPR) [2], a practical mechanism should protect sensitive relationships before they leave the user side.

*Corresponding author. Email: changzhihua_hz@163.com

Local Differential Privacy (LDP) provides a suitable protection paradigm for this setting because each user perturbs its local information before sending it to an untrusted aggregator [3]. Randomized response is a fundamental LDP mechanism for binary data [4], and it is naturally applicable to edge indicators in a neighbor list. Nevertheless, directly perturbing all adjacency indicators usually introduces many false edges in sparse graphs. In addition, graph differential privacy must consider whether the protected object is an edge or a node[5,6]. This paper focuses on edge-level local privacy because the existence of a relationship between two users or devices is the sensitive item in many distributed graph-sharing scenarios.

Traditional privacy-preserving methods for graph-structured data, such as anonymization techniques, are difficult to defend against inference attacks that exploit background knowledge. Differential privacy, by contrast, provides a rigorous and quantifiable privacy-preserving mechanism that is less dependent on assumptions about the attacker's background knowledge. However, many existing graph publishing and analysis schemes under differential privacy are designed for centralized graph data, while practical distributed graphs are often held as decentralized local views. In this setting, no single entity has access to the complete graph structure. Each user only maintains sensitive direct-neighbor relations, making unconstrained collection of the whole graph infeasible and prone to privacy leakage. Moreover, due to the locality of subgraph views, the extractable features are limited, so centralized graph privacy-preserving methods cannot be directly adapted. Applying differential privacy to distributed graph publishing mainly faces two challenges: first, local subgraphs must be collected with a strict LDP guarantee; second, the synthetic graph should preserve global and local topology under strong perturbation. Fine-grained neighbor-list perturbation can retain local details but may introduce many false edges, whereas coarse-grained degree perturbation provides stable statistics but may lose community structure.

This paper focuses on undirected and unweighted decentralized network graphs and on edge-sensitive local views. We study Decentralized Graph Publishing based on Dual-granularity Differential Privacy (DDP-DGP). A closely related high-level dual-granularity graph-publishing pipeline was reported by Gao et al. [7]. Accordingly, this manuscript does not claim that pipeline itself as a new differential-privacy primitive; instead, it focuses on revised privacy accounting, a full-support boundary-corrected degree kernel, explicit degree-calibrated sampling and endpoint reconciliation, and clarified privacy-preserving reconstruction.

- (i) A dual-granularity local-privacy collection mechanism is specified with explicit neighboring-input definitions and budget accounting. Coarse-grained node degrees are protected by a full-support boundary-corrected random-jump kernel, while fine-grained neighbor indicators are protected by bitwise randomized response with $\epsilon_b = \epsilon_2/(|V|-1)$.

- (ii) A mathematically explicit degree-calibrated sampling rule uses only privatized degree and neighbor-list outputs to reduce false-edge inflation. The rule includes clipping, empty-list handling, and an expected post-sampling degree characterization, and is followed by an explicit probabilistic reconciliation rule for inconsistent endpoint reports.
- (iii) A community-aware post-processing strategy reconstructs the released graph using only privatized data. It applies Louvain partitioning, clipped Chung-Lu probabilities for intra-community edges, and inter-community edge sampling from privatized aggregate counts. The reported experiments assess structural utility; downstream learning performance is treated as an open empirical question rather than inferred directly from structural metrics.

Different from centralized graph publishing, decentralized network systems introduce privacy risks during local-view collection as well as during graph release. DDP-DGP therefore separates privacy-sensitive user-side randomization from publisher-side post-processing and makes the privacy boundary explicit. This separation is also important for interpreting comparisons with centralized-DP graph generators.

2. Related Work

2.1. Data Security and Privacy in Distributed Graph Publishing

Data security and privacy protection are fundamental requirements in decentralized graph publishing. In decentralized online social networks, recommendation systems, mobile interaction networks, and edge-assisted collaboration platforms, each participant often holds only a local ego-network rather than a complete graph. Uploading raw neighbor lists to an untrusted publisher can reveal private relationships before publication. Accordingly, this work studies privacy-preserving local-view collection and synthetic graph publication under LDP. Existing studies provide foundations in local privacy [3, 4], graph differential privacy [5, 6], and closely related dual-granularity graph publication [7]. Within EAI Endorsed Transactions on Scalable Information Systems, Sun and Sun [8] studied privacy-preserving publication of large-scale rating data, while Rehman et al. [9] investigated large-social-graph summarization and identified privacy, compression, and transmission efficiency as practical motivations. These studies are relevant to scalable privacy-aware data publication and graph processing, although neither provides an LDP graph-publication mechanism. Broader work on private graph release and decentralized graph analysis is reviewed below. Centralized and local models are distinguished throughout because their trust assumptions and privacy accounting are not interchangeable.

2.2. Privacy Protection for Graph Data

As a type of non-Euclidean data, graph-structured data lacks translational invariance and features complex connections between nodes, posing severe challenges to privacy protection. Hay et al. [5] provided an early differentially private method for estimating graph degree distributions. Since then, significant progress has been made in differential privacy-based graph protection. Unlike graph mining and analysis tasks that release only specific statistics, synthetic graph publishing under differential privacy is a more general paradigm applicable to multiple downstream analysis tasks. In distributed graph scenarios, it is necessary to preserve rich structural features as much as possible while protecting sensitive local views. Therefore, the trade-off between data privacy and graph utility has become a key challenge in synthetic graph publishing.

Iftikhar et al. [10] proposed the edge differential privacy synthetic graph publishing method dK-Microaggregation, which implements dK-degree model microaggregation via distance constraints to reduce noise introduction. Gao and Li [11] designed graph reconstruction methods by integrating dK-1, dK-2, and dK-3 sequences, enhancing the richness of graph structural information to improve synthetic graph utility. Huang et al. [12] proposed PBCN based on clustering and randomized perturbation and introduced an adjacency-degree privacy metric. Kang et al. [13] introduced personalized differential privacy and combined dimensionality reduction with segmented sampling to achieve hierarchical data protection. Gao et al. [14] proposed a social network graph generation method that uses the Louvain algorithm to discover community information and implements personalized differential privacy based on community average weight ratios. Zhang et al. [15] proposed APDP, which achieves personalized privacy control based on user attribute values and the TOPSIS method. Li et al. [16] proposed ACGDP for Internet of Vehicles data by combining anonymization with graph differential privacy. Yuan et al. [17] proposed PrivGraph, which groups nodes into communities under privacy guarantees and generates edges differently inside and outside communities to mitigate the impact of noise on graph utility. Hou et al. [18] proposed PPDU for dynamic graph publishing with local differential privacy, and Gao et al. [19] designed PGU-DNDP to capture update sequences of network snapshot graphs and reduce excessive noise through partial graph updates and downsampling.

For decentralized graphs, LDP is commonly adopted because users perturb local data before collection [3]. Direct randomized response over neighbor indicators can serve as a simple baseline in decentralized graph synthesis [20]; in sparse graphs, false-positive inflation can substantially densify the reported graph, a limitation emphasized in closely related dual-granularity publication work [7]. Qin et al. [20] proposed LDPGen for synthetic decentralized social graphs; Wei et al. [21] proposed AsgLDP for joint structure and attribute protection; and Lin et al. [22] studied private learning on decentralized

graphs. More recent work has investigated locally private community detection [23, 24], locally private graph embedding [25], continuous publication of weighted graphs [26], and communication-efficient graph statistics under LDP [27]. Gao et al. [7] recently reported a closely related dual-granularity graph-publication pipeline combining random-jump degree perturbation, randomized-response neighbor perturbation, degree-guided sampling, community discovery, and intra-/inter-community reconstruction. We therefore treat the high-level pipeline as prior work and limit our claims to the revised formalization and implementation details in Section 3.

2.3. Differential Privacy

As a rigorous privacy-preserving framework, differential privacy aims to prevent attackers from inferring specific individual information by adding noise or introducing random perturbation. The mathematical definitions related to differential privacy are given as follows.

Definition 1 ϵ -Differential Privacy [28]. For any two adjacent datasets D and D' differing by only one individual, and any output set $S \in \text{Range}(M)$, the algorithm M satisfies the inequality in Equation (1):

$$\Pr[M(D) \in S] \leq e^\epsilon \Pr[M(D') \in S] \quad (1)$$

Then M is said to satisfy ϵ -differential privacy. Here, the parameter ϵ is the privacy budget, which regulates the strength of privacy protection: the smaller ϵ is, the stricter the privacy protection provided.

Definition 2 (Global Sensitivity) [28]. Global sensitivity is a key parameter in differential privacy that measures the maximum possible change in the output of a function between adjacent datasets. For any query function $f: D \rightarrow \mathbf{R}^d$ defined on adjacent datasets D and D' , its global sensitivity is defined in Equation (2):

$$\Delta f = \max_{D, D'} \|f(D) - f(D')\| \quad (2)$$

Theorem 1 (Laplace Mechanism) [28]. The Laplace mechanism is designed for real-valued queries. It hides the influence of individual data records by adding noise sampled from the Laplace distribution to the query result. For a query function f , the mechanism M is expressed as $M(D) = f(D) + Y$, where Y follows a Laplace distribution with scale parameter $\Delta f/\epsilon$.

Differential privacy was originally designed for centralized scenarios, where the raw data from all parties is collected by a trusted data center, and a trusted data administrator performs randomized processing to ensure that the privacy risk of individual records is controlled within a certain level. However, in many practical applications, the data administrator may not be trustworthy. To address this issue, local differential privacy is proposed, which adds perturbation at the source of data generation.

Data owners do not need to trust the data publisher and can implement privacy protection locally.

Definition 3 (ϵ -Local Differential Privacy) [3, 29]. For any two input values t and t' , and any possible output value $y \in \text{Range}(M)$, if Equation (3) holds:

$$\Pr[M(t) = y] \leq e^\epsilon \Pr[M(t') = y] \quad (3)$$

Then M is said to satisfy ϵ -local differential privacy. Most current local differential privacy schemes are based on the randomized response mechanism, which introduces randomness to protect individuals' true responses.

Theorem 2 (Randomized Response Mechanism). Warner introduced randomized response for sensitive binary survey questions [4]. Under the modern ϵ -LDP parameterization, the true bit is reported with probability p and the opposite bit with probability $1-p$. When $p = e^\epsilon / (1 + e^\epsilon)$, the mechanism satisfies ϵ -local differential privacy [29].

Generalized randomized-response techniques were studied in survey methodology before the modern LDP formulation [30]. In current LDP, categorical and multi-valued data are commonly protected with mechanisms such as generalized randomized response, while continuous or large-domain data can be handled through discretization or locally private numerical mechanisms [29]. LDP also inherits differential-privacy composition properties; sequential composition applies when multiple private mechanisms depend on the same user record.

Property 1 (Sequential Composition) [28]. If mechanisms $M_1, \dots, M_n$ are applied to the same private input and M_i satisfies ϵ_i -differential privacy, then their joint output satisfies $(\sum_i \epsilon_i)$ -differential privacy. This conservative basic-composition rule is used below to account for the complete local report.

Property 2 (Parallel Composition) [28]. When differential privacy algorithms M_i are applied in parallel to disjoint datasets D_i , if each M_i satisfies ϵ_i -differential privacy, then the overall mechanism satisfies $\max\{\epsilon_i\}$ -differential privacy.

2.4. Graph Differential Privacy

As an extension of differential privacy in the field of graph data, graph differential privacy aims to protect the privacy of nodes or edges during the publication or analysis of graph data. Different from the organizational forms of categorical data or tabular data, graph data belongs to non-Euclidean data and lacks translational invariance, with its inherent information jointly constructed by a large number of nodes and edges. A graph can be represented as a binary tuple $G(V, E)$ consisting of a node set V and an edge set E , where V denotes the set of all nodes and their attributes in the graph, and E denotes the set of edges between nodes. When differential privacy is introduced into the domain of graph-structured data, node differential privacy and edge

differential privacy are naturally derived according to different protected objects.

Definition 4 (ϵ -Node Differential Privacy) [5, 6]. A mechanism M satisfies ϵ -node differential privacy if and only if for any two adjacent graphs $G_1(V_1, E_1)$ and $G_2(V_2, E_2)$ satisfying $|V_1 \oplus V_2| = 1$ and $E_1 \oplus E_2 = \{(u, v) \mid u \in (V_1 \oplus V_2) \text{ or } v \in (V_1 \oplus V_2)\}$, and for any output set $S \in \text{Range}(M)$, Equation (4) holds:

$$\Pr[M(G_1) \in S] \leq e^\epsilon \Pr[M(G_2) \in S] \quad (4)$$

Definition 5 (ϵ -Edge Differential Privacy) [5, 6]. A mechanism M satisfies ϵ -edge differential privacy if and only if for any two adjacent graphs $G_1(V_1, E_1)$ and $G_2(V_2, E_2)$ satisfying $|V_1 \oplus V_2| + |E_1 \oplus E_2| = 1$, and for any output set $S \in \text{Range}(M)$, Equation (5) holds:

$$\Pr[M(G_1) \in S] \leq e^\epsilon \Pr[M(G_2) \in S] \quad (5)$$

Edge differential privacy focuses on protecting the existence of individual edges from being disclosed, while node differential privacy aims to preserve the privacy of a single node and all its associated edges in a graph. In terms of privacy protection strength, node differential privacy provides a stronger guarantee, but it typically requires the injection of larger amounts of noise. The graph publishing method designed in this paper perturbs degree information and subgraph neighbor lists, and introduces random edge generation during the graph reconstruction process to protect the privacy of connections between nodes. Thus, this work focuses on achieving edge-level privacy protection for graphs.

3. The Proposed Method

3.1. Problem Formulation and Threat Model

Let $G=(V,E)$ be an undirected, unweighted graph with $N=|V|$ public pseudonymous node identifiers. User v_i holds a binary local neighbor vector $a_i=(a_{i1}, \dots, a_{iN})$, excluding a_{ii} , and degree $d_i = \sum_{j \neq i} a_{ij}$. The publisher knows the public

candidate identifier universe V and distributes N and the privacy parameters, but does not receive raw a_i or d_i . The publication objective is to generate a synthetic graph G' whose structural statistics approximate those of G while preventing raw edge indicators from being directly disclosed.

Threat model. The publisher is assumed to be honest-but-curious: it follows the specified protocol but may attempt to infer private relationships from received messages. The local randomizer executes on the user-controlled side and is trusted to use fresh randomness. The main analysis assumes non-colluding users and authenticated, stable pseudonymous node identifiers;

malicious parameter substitution, Sybil attacks, compromised local randomizers, and identifier privacy are outside the present scope. Stronger malicious or colluding adversaries require additional cryptographic or robustness mechanisms beyond the analyzed threat model.

Neighboring-input definitions. Two local neighbor vectors a_i and a_i' are edge-adjacent when their Hamming distance is one, meaning that exactly one incident-edge indicator changes. For the stronger whole-local-view notion, arbitrary vectors in $\{0,1\}^{N-1}$ may differ in as many as $N-1$ positions. DDP-DGP explicitly reports both guarantees: bitwise randomized response with ϵ_b protects one changed edge with ϵ_b -LDP, while sequential composition over all $N-1$ indicators yields $\epsilon_2 = (N-1)\epsilon_b$ for an arbitrary whole-neighbor-vector change. The degree randomizer is defined on the complete domain $\{0, \dots, N-1\}$.

For one collection round, the degree and neighbor-list mechanisms are applied to the same local view. Therefore, the complete local report satisfies $(\epsilon_1 + \epsilon_2)$ -LDP under the stronger whole-local-view notion and consequently also satisfies the same (looser) bound for edge adjacency; the tighter edge-adjacency bound is $\epsilon_1 + \epsilon_b$. If the graph is recollected over T rounds, basic sequential composition

gives cumulative loss no greater than $\sum_i (\epsilon_{1,i} + \epsilon_{2,i})$ for the

whole-local-view guarantee. Practical deployments should maintain this privacy ledger and stop collection when the allocated cumulative budget is exhausted.

3.2. Algorithm Framework

The overall DDP-DGP workflow is illustrated in Figure 1 and explicitly separates privacy-sensitive user-side randomization from publisher-side post-processing. Each user receives N and budgets ϵ_1 and ϵ_2 . The degree mechanism produces a privatized degree $\hat{d}_i$, while each of the $N-1$ neighbor indicators is randomized with $\epsilon_b = \epsilon_2 / (N-1)$. The privatized degree calibrates a sampling step applied only to the noisy neighbor list. Users transmit $\hat{d}_i$ and a sparse representation of the sampled noisy list. The publisher then reconciles the two endpoint reports for each undirected pair, applies Louvain community detection [31] to the resulting privatized graph, and reconstructs intra-community and inter-community edges. The intra-community model follows an expected-degree Chung-Lu construction [32, 33].

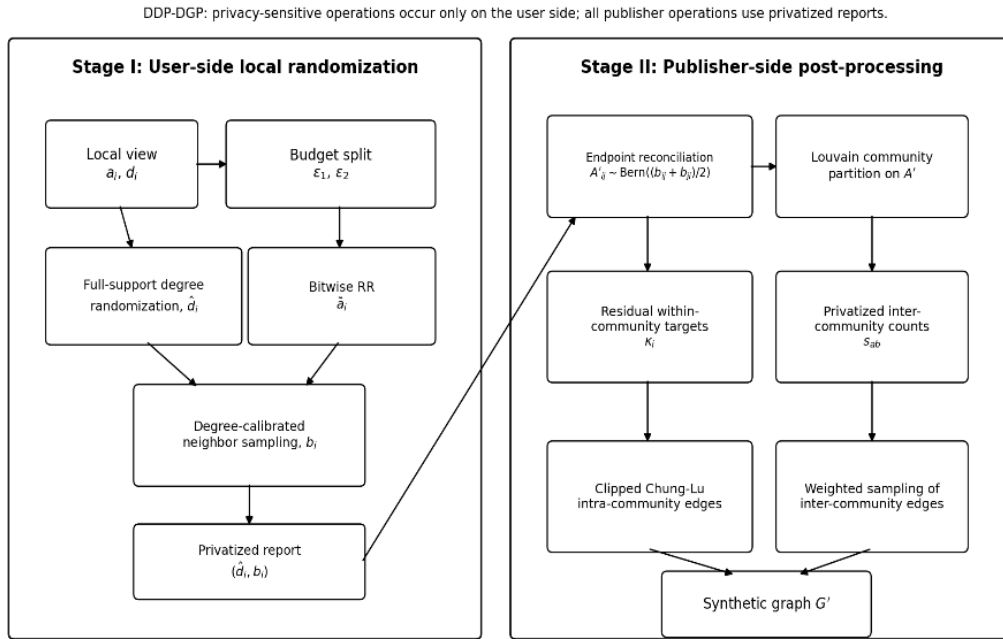


Figure 1. Overview of DDP-DGP with user-side randomization and publisher-side post-processing

3.3. Algorithm Implementation

Coarse-grained degree randomization. Building on prior random-jump degree perturbation [7, 21], we redefine the mechanism on the complete degree domain $D = \{0, \dots, N-1\}$. Choose an integer radius $0 \leq r \leq (N-1)/2$ and define

$L(d) = \min\{\max(d-r, 0), N-(2r+1)\}$ and $J_r(d) = \{L(d), \dots, L(d)+2r\}$. Thus, $J_r(d)$ always contains d and has exactly $2r+1$ elements, including at the boundaries $d=0$ and $d=N-1$. For any output $y \in D$, the

mechanism uses three strictly positive weights: e^{ϵ_1} at $y=d$, $e^{\epsilon_1/2}$ for the other points in $J_r(d)$, and 1 outside $J_r(d)$.

$$\Pr[M_{\text{deg}}(d) = y] = w_d(y) / Z_r, \quad Z_r = e^{\epsilon_1} + 2r \cdot e^{\epsilon_1/2} + N - 2r - 1 \quad (6)$$

Here $w_d(y) = e^{\epsilon_1}$ if $y=d$, $w_d(y) = e^{\epsilon_1/2}$ if $y \in J_r(d) \setminus \{d\}$, and $w_d(y)=1$ otherwise. The normalization Z_r is independent of d and all outputs have nonzero probability. Hence, for any d, d' and y , $w_d(y) / w_{d'}(y) \leq e^{\epsilon_1}$, proving ϵ_1 -LDP. Because clipping and boundary correction do not in general preserve exact unbiasedness, no unbiasedness property is asserted for the perturbed degree.

Radius selection. Rather than using a privacy-leakage surrogate whose derivation was not established, r is selected by a public, data-independent expected squared-error criterion. Because N and ϵ_1 are public parameters, the discrete search consumes no privacy budget. Under a uniform public prior over the admissible degree domain, the objective is:

$$r^* = \arg \min_{0 \leq r \leq (N-1)/2} \frac{1}{N} \sum_{d=0}^{N-1} \sum_{y=0}^{N-1} (y-d)^2 \Pr[M_{\text{deg}}(d) = y] \quad (7)$$

If an application has a public, non-sensitive degree prior, that prior can replace the uniform weighting in Equation (7). The mechanism itself remains unchanged. This formulation resolves boundary support, normalization, and radius-selection ambiguities by specifying the degree kernel directly over the full output domain.

Fine-grained neighbor-list perturbation. Let ϵ_2 be the total budget assigned to the $N-1$ neighbor indicators and set $\epsilon_b = \epsilon_2 / (N-1)$. Each a_{ij} , $j \neq i$, is independently perturbed by binary randomized response with parameter ϵ_b :

$$\Pr[\tilde{a}_{ij} = a_{ij}] = \frac{e^{\epsilon_b}}{1 + e^{\epsilon_b}}, \quad \Pr[\tilde{a}_{ij} \neq a_{ij}] = \frac{1}{1 + e^{\epsilon_b}} \quad (8)$$

Degree-calibrated sampling. Let $m_i = \sum_{j \neq i} \tilde{a}_{ij}$ be the degree of the randomized neighbor vector and let $t_i = \text{clip}(\text{round}(\hat{d}_i), 0, N-1)$. If $m_i=0$, set $\rho_i=0$. Otherwise set $\rho_i = \min\{1, t_i/m_i\}$. Each position with $\tilde{a}_{ij}=1$ is independently retained with probability ρ_i , yielding b_{ij} ; zero entries remain zero. Conditional on $(\tilde{a}_i, \hat{d}_i)$, the expected sampled degree is $E\left[\sum_j b_{ij} \mid \tilde{a}_i, \hat{d}_i\right] = \rho_i m_i = \min\{t_i, m_i\}$.

Thus the rule cannot create new positive indicators, handles an empty noisy list explicitly, clips invalid degree targets,

and suppresses RR-induced densification using only privatized quantities.

Because the sampling rule depends only on $\hat{d}_i$ and $\tilde{a}_i$, it is differential-privacy post-processing and consumes no additional budget. Algorithm 1 explicitly lists the clipping, empty-list handling, retention-probability computation, and sampling operations.

Algorithm 1. User-side dual-granularity data collection

Input: ϵ_1 ; total neighbor-list budget ϵ_2 ; $\epsilon_b = \epsilon_2 / (N-1)$; $N=|V|$; local neighbor vector a_i .
 Output: privatized degree $\hat{d}_i$ and sampled noisy neighbor indicators b_i .

1. Compute $d_i = \sum_{j \neq i} a_{ij}$ and select r^* by Equation (7) from public N and ϵ_1 .
2. Compute $L(d_i) = \min\{\max(d_i - r^*, 0), N - (2r^* + 1)\}$ and $J_r(d_i) = \{L, \dots, L + 2r^*\}$.
3. Sample $\hat{d}_i$ from the full-support kernel in Equation (6).
4. For every $j \neq i$, perturb a_{ij} by Equation (8) with ϵ_b to obtain $\tilde{a}_{ij}$.
5. Set $m_i = \sum_{j \neq i} \tilde{a}_{ij}$ and $t_i = \text{clip}(\text{round}(\hat{d}_i), 0, N-1)$.
6. Set $\rho_i = 0$ if $m_i = 0$; otherwise $\rho_i = \min\{1, t_i/m_i\}$.
7. For each $\tilde{a}_{ij}=1$, draw $z_{ij} \sim \text{Bernoulli}(\rho_i)$ and set $b_{ij} = z_{ij}$; if $\tilde{a}_{ij}=0$, set $b_{ij}=0$.
8. Transmit $\hat{d}_i$ and the sparse index set $\{j: b_{ij}=1\}$; do not transmit a_i or d_i .

Publisher-side reconciliation. After receiving $(\hat{d}_i, b_i)$ from all users, the publisher constructs a symmetric privatized adjacency matrix A' . For each unordered pair $i < j$, the two endpoint reports can disagree. DDP-DGP uses probabilistic averaging: $A'_{ij} = A'_{ji}$ is sampled from $\text{Bernoulli}((b_{ij} + b_{ji})/2)$, and $A'_{ii} = 0$. Agreements are preserved, whereas a one-sided report is retained with probability $1/2$. Consequently, $E[A'_{ij} \mid b_{ij}, b_{ji}] = (b_{ij} + b_{ji})/2$; unlike OR or AND reconciliation, this rule does not systematically bias every disagreement toward density inflation or sparsification. It is pure post-processing of local-DP outputs.

Community and inter-community statistics. Louvain community detection [31] is applied only to A' , producing $C = \{C_1, \dots, C_K\}$. For each node i , define the target $t_i = \text{clip}(\text{round}(\hat{d}_i), 0, N-1)$, the privatized external degree $\delta_i^{\text{out}} = \sum_{j \in C(i)} A'_{ij}$, and the residual within-community target $\kappa_i = \text{clip}(t_i - \delta_i^{\text{out}}, 0, |C(i)| - 1)$. For each pair of distinct communities C_a and C_b , define $s_{ab} = \sum_{u \in C_a} \sum_{v \in C_b} A'_{uv}$.

Because A' is already privatized, s_{ab} is post-processing and requires no extra privacy budget; defensively, it is clipped to $[0, |C_a||C_b|]$.

Intra-community reconstruction follows a valid Chung-Lu expected-degree construction [32, 33]. For each unordered pair of distinct nodes $i < j$ within the same community C , an edge is drawn once using the clipped probability in Equation (9). Self-loops are excluded and each unordered pair is considered only once, so duplicate edges cannot occur.

$$p_{ij}^{\text{in}} = \begin{cases} \min \left\{ 1, \frac{\kappa_i \kappa_j}{\sum_{u \in C} \kappa_u} \right\}, & i \neq j, v_i, v_j \in C, \sum_{u \in C} \kappa_u > 0, \\ 0, & \sum_{u \in C} \kappa_u = 0. \end{cases} \quad (9)$$

When clipping in Equation (9) does not bind, the expected within-community degree is approximately κ_i ; the external target δ_i^{out} is handled separately, so the combined construction is calibrated toward t_i rather than double-counting the same degree target. For inter-community pair (C_a, C_b) , exactly s_{ab} distinct node pairs are sampled without replacement. Candidate pair (u, v) receives weight $\delta_u^{\text{out}} \delta_v^{\text{out}}$; if all weights are zero, uniform sampling is used. This construction uses only privatized A' and $\hat{\mathbf{d}}$ and therefore introduces no new access to raw aggregate edge counts.

Algorithm 2 summarizes the complete publisher-side procedure and makes the reconciliation rule, s_{ab} definition, clipping, and intra/inter-community sampling explicit.

Algorithm 2. DDP-DGP graph publication

Input: $\epsilon_1, \epsilon_2, N=|V|$, and user-side local neighbor vectors a_i (used only inside Algorithm 1).
 Output: synthetic graph G' .

1. Publisher broadcasts $N, \epsilon_1, \epsilon_2$ and $\epsilon_b = \epsilon_2 / (N-1)$.
2. Each user executes Algorithm 1 locally and sends only $(\hat{\mathbf{d}}_i, b_i)$.
3. For every $i < j$, sample $A'_{ij} = A'_{ji} \sim \text{Bernoulli}((b_{ij} + b_{ji})/2)$; set $A'_{ii} = 0$.
4. Apply Louvain [31] to A' and obtain communities $C = \{C_1, \dots, C_K\}$.
5. For each i , compute $t_i, \delta_i^{\text{out}}$, and κ_i from privatized quantities.
6. For every unordered within-community pair $i < j$, sample one edge using Equation (9).
7. For each community pair $a < b$, compute s_{ab} from A' and clip it to $[0, |C_a||C_b|]$.
8. Sample exactly s_{ab} distinct cross-community pairs without replacement, weighted by $\delta_u^{\text{out}} \delta_v^{\text{out}}$; use uniform weights if all are zero.
9. Merge intra- and inter-community edge sets, remove no additional information, and return G' .

3.4. Privacy Analysis

Theorem 3 (privacy of the degree mechanism). For any $d, d' \in \{0, \dots, N-1\}$ and any output y , the full-support random-jump mechanism in Equation (6) satisfies $\Pr[M_{\text{deg}}(d) = y] / \Pr[M_{\text{deg}}(d') = y] \leq e^{\epsilon_1}$. Proof. The normalizer Z_r is constant for every input degree. Each weight $w_d(y)$ belongs to $\{1, e^{\epsilon_1/2}, e^{\epsilon_1}\}$, so the largest possible ratio between weights under two inputs is e^{ϵ_1} ; all probabilities are positive. Boundary degrees use a shifted fixed-cardinality window and therefore do not create support mismatch. Thus M_{deg} is ϵ_1 -LDP.

Theorem 4 (neighbor-list privacy). Each indicator is randomized with $\epsilon_b = \epsilon_2 / (N-1)$. If a_i and a'_i are edge-adjacent, only one input bit changes and the joint output ratio is bounded by e^{ϵ_b} ; hence the neighbor-list mechanism is ϵ_b -LDP for one-edge adjacency. For arbitrary local neighbor vectors with Hamming distance h , sequential composition gives $h\epsilon_b$ -LDP, and the worst case $h=N-1$ gives ϵ_2 -LDP. This distinction separates the one-edge guarantee from the stronger whole-neighbor-vector guarantee.

The degree and neighbor-list randomizers act on the same local view, so basic sequential composition [28] yields $(\epsilon_1 + \epsilon_2)$ -LDP for the full local report under arbitrary whole-vector changes. Under one-edge adjacency, the tighter bound is $\epsilon_1 + \epsilon_b$ because only one neighbor bit changes, although the looser $(\epsilon_1 + \epsilon_2)$ statement remains valid. Degree-calibrated sampling, endpoint reconciliation, Louvain partitioning, s_{ab} computation, Chung-Lu reconstruction, and inter-community sampling use only privatized outputs and are post-processing. For repeated collection rounds, cumulative whole-view privacy loss is bounded by the sum of the per-round $(\epsilon_1 + \epsilon_2)$ budgets.

3.5. Complexity Analysis

This section separates worst-case computational/storage bounds from implementation-dependent sparse costs. Let $N=|V|$, let $|E'|$ denote the number of edges retained in the reconciled privatized graph A' , and let $|E_G|$ denote the number of edges in the released graph.

Communication and user-side computation. Algorithm 1 touches $N-1$ candidate indicators, so the worst-case local computation is $O(N)$ per user and $O(N^2)$ across all users. Transmitting a dense randomized vector would also cost $O(N)$ bits per user and $O(N^2)$ overall. With sparse-index transmission, user i sends $\hat{\mathbf{d}}_i$ plus the retained index set $\{j: b_{ij}=1\}$; the realized payload is $O(1+|b_i|_0)$ and the total realized payload is $O(N + \sum_i |b_i|_0)$. This is a realized sparse cost, not a worst-case guarantee.

Publisher time. Reconciliation is $O(N^2)$ in a dense implementation but can be implemented over transmitted sparse candidates plus bookkeeping. Louvain is empirically near-linear in the number of stored edges for sparse graphs, although no strict worst-case linear guarantee is assumed. Naively considering all within-

community pairs costs $O\left(\sum_c |C_c|^2\right)$, and cross-

community candidate enumeration can also be quadratic; weighted sparse sampling or alias/reservoir implementations can reduce practical cost when candidate sets are sparse. Communication-efficient LDP graph statistics illustrate related scalability techniques[27].

Space complexity. A dense publisher-side adjacency matrix requires $O(N^2)$ space. Under an adjacency-list or sparse-edge representation, storing node metadata, the reconciled graph, and the released graph requires $O\left(N + |E'| + |E_G|\right)$ space, plus $O(K)$ community metadata.

These bounds explicitly account for stored edges rather than only node and community metadata. Empirical scaling in runtime, peak memory, transmitted bytes, and reconstruction time is not evaluated in the present study and remains an important direction for implementation-level validation.

4. Experimental Results and Analysis

4.1. Experimental Setup

This section evaluates DDP-DGP using the four-dataset experiments reported in Figures 2-7. All methods are implemented in Python 3.12 on Windows 11 with 16 GB RAM, and each plotted point is the mean of ten independent runs. The present study reports mean curves but does not include run-level dispersion estimates or hypothesis tests. Accordingly, between-method comparisons are interpreted descriptively, and inferential terms such as “statistically significant” are not used.

4.2. Experimental Data and Evaluation Metrics

Four public graph datasets are adopted in the experiments. email-Eu-core, Facebook, and Wiki-vote originate from the Stanford Network Analysis Platform (SNAP), while Chameleon is a Wikipedia webpage graph used in the DGL/Wikipedia-network benchmark family. All graphs are evaluated as undirected and unweighted: self-loops are removed, duplicate edges are merged, directed edges are symmetrized when necessary, and density is calculated as $2|E|/(|V|(|V|-1))$. Table 1 reports the processed undirected counts used in the experiments. The reproducibility scope of the present experimental report is limited to the stated preprocessing protocol and processed graph statistics; exact source-file identifiers, preprocessing scripts, processed edge lists, and random-seed files are not included here.

Table 1. Dataset information

Dataset	Nodes	Edges	Graph Density	Graph Type
email-Eu-core	1005	16064	0.02500	Communication Network
Chameleon	2277	31371	0.01200	Webpage Graph
Facebook	4039	88234	0.01082	Social Network
Wiki-vote	7115	100762	0.00205	Voting Network

The comparison labels are standardized to match Figures 3-6: RNL denotes direct randomized-response neighbor-list perturbation; PRNL denotes an author-implemented personalized structural neighbor-list reference inspired by AsgLDP [21], and PRNL is not an algorithm name defined in [21]; LDPGen follows the decentralized LDP graph-generation framework of Qin et al. [20]; and PrivGraph_dis denotes a distributed structural adaptation inspired by PrivGraph [17]. Importantly, the original PrivGraph is a centralized-DP publication method, so PrivGraph_dis is treated as a structural reconstruction reference rather than as a privacy-equivalent LDP baseline. The privacy accounting used for DDP-DGP is $\epsilon = \epsilon_1 + \epsilon_2$. RNL, PRNL, and LDPGen are interpretable as local-DP baselines only when configured under the same total local budget and preprocessing protocol; utility comparisons involving PrivGraph_dis should therefore be read as structural comparisons rather than as comparisons under an identical trust model.

Degree-distribution metric. KL divergence is computed on integer degree bins $k=0, \dots, k_{\max}$, where $k_{\max}$ is the largest degree observed in either the original or synthetic graph for the compared pair. To avoid undefined log terms, additive smoothing $\alpha=10^{-12}$ is applied to every bin of both histograms, followed by renormalization to probability distributions P and Q before Equation (10) is evaluated. Smaller KL divergence indicates closer degree-distribution agreement.

$$D_{\text{KL}}(P \| Q) = \sum P(x) \log \frac{P(x)}{Q(x)} \quad (10)$$

Community-structure metric. Modularity Q is computed using the Louvain partition with resolution 1.0 and the same random-seed schedule for paired runs. The reported modularity relative error is $\left| Q_{\text{syn}} - Q_{\text{orig}} \right| / \max\left(\left| Q_{\text{orig}} \right|, 10^{-12}\right)$, which prevents division by a numerically near-zero original modularity. Equation (11) defines the modularity statistic itself; the relative-error convention is stated here to make implementation reproducible.

$$Q = \frac{1}{2|E|} \sum_{i,j} \left[A_{ij} - \frac{k_i k_j}{2|E|} \right] \delta(c_i, c_j) \quad (11)$$

Node-importance metric. Eigenvector centrality is computed component-wise for connected components with at least two nodes; isolated nodes are assigned zero. Each component vector is L2-normalized before concatenation and the full vector is then L2-normalized over the common node-identifier order. The reported error is the mean absolute difference between the original and synthetic centrality vectors. This convention avoids ambiguity when the graph is disconnected.

$$x_i = \frac{1}{\lambda} \sum_{j \in N(i)} A_{ij} x_j \quad (12)$$

Connectivity metric. Average path length is computed on the largest connected component (LCC) of each graph, using the mean of shortest-path distances over unordered node pairs in that LCC. LCC size is treated as a companion diagnostic because perturbation can change component membership. Equation (13) is therefore interpreted over the selected LCC rather than over unreachable node pairs in the full graph. Because run-level LCC sizes are not

reported, the path-length comparison is interpreted at the level of the displayed mean curves.

$$L = \frac{1}{n(n-1)} \sum_{i \neq j} d(i, j) \quad (13)$$

The total local budget is $\epsilon = \epsilon_1 + \epsilon_2 \in \{0.5, 1, 1.5, 2, 2.5, 3, 3.5\}$. Unless otherwise stated, $\epsilon_1 : \epsilon_2 = 1 : 9$ and each neighbor bit uses $\epsilon_b = \epsilon_2 / (N-1)$. This allocation gives the whole-neighbor-vector ϵ_2 bound established in Section 3.4. Each plotted point is the mean of ten runs. Standard deviations, confidence intervals, and p-values are not reported; consequently, the analysis is descriptive rather than inferential.

4.3. Experimental Result Analysis

Figure 2 examines privacy-budget allocation on the Chameleon dataset. It displays the four structural metrics for $\epsilon_1 : \epsilon_2$ ratios from 9:1 to 1:9. Corresponding allocation-sensitivity experiments are not reported for email-Eu-core, Facebook, or Wiki-vote; therefore, the allocation-sensitivity conclusion is restricted to Chameleon.

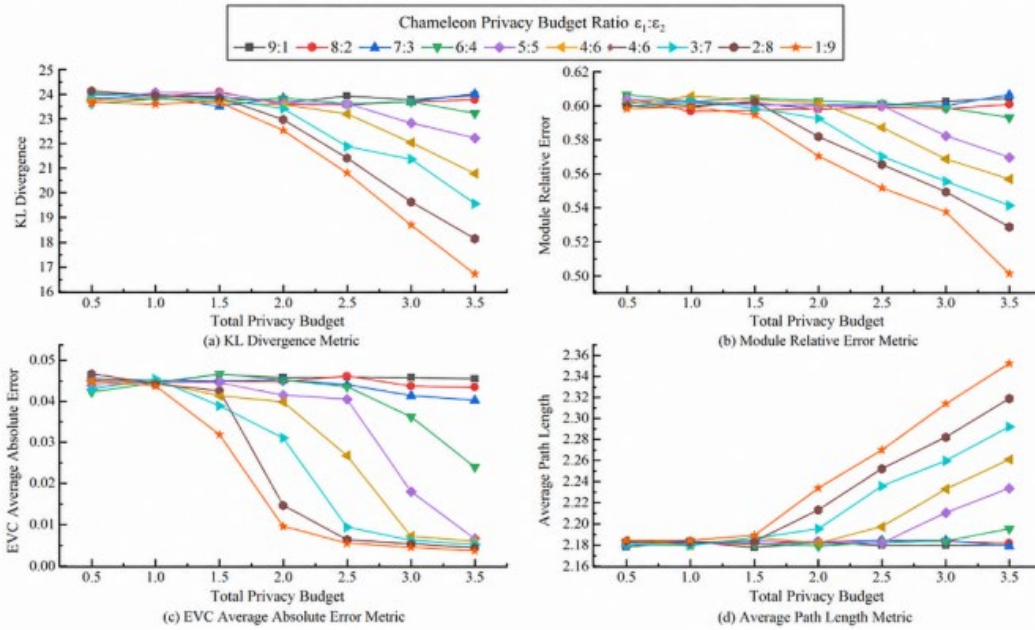


Figure 2. Chameleon synthetic graph utility under different privacy budget allocation ratios

In Figure 2, the curves for total budgets 0.5 and 1.0 are visually close across several allocation ratios. As the total budget increases, the curves separate more clearly, and allocating a larger share to the fine-grained neighbor-list mechanism tends to improve the displayed structural metrics. The degree signal remains useful as a calibration target. Based on this descriptive pattern, $\epsilon_1 : \epsilon_2 = 1 : 9$ is used

as the default setting in the reported experiments. The figure does not establish a statistically tested optimum, and cross-dataset allocation sensitivity remains unverified.

Figure 3 reports KL divergence of degree distributions across the four datasets. In the displayed curves, DDP-DGP generally has lower KL divergence than the comparison methods, particularly at smaller privacy budgets. This

pattern is consistent with using privatized degree information to calibrate false-edge suppression before

reconstruction. Because run-level uncertainty estimates are not reported, the comparison is interpreted descriptively.

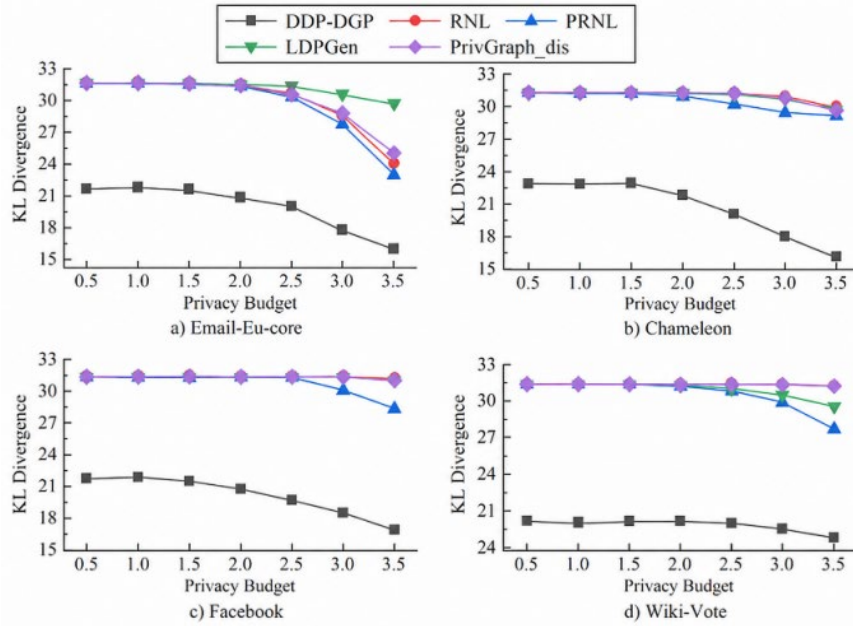


Figure 3. Experimental results of KL divergence of degree sequences under different privacy budgets

Figure 4 reports modularity relative error. Across the displayed curves, DDP-DGP shows lower error than the comparison labels over much of the tested budget range. A plausible mechanism is that community partitioning on the privatized graph provides local structural guidance before intra- and inter-community reconstruction. Because no

inferential statistics are reported, the comparison is descriptive rather than a claim of statistical significance. Moreover, the centralized-DP provenance of PrivGraph means its structural reference should not be interpreted as an equal-privacy LDP comparison.

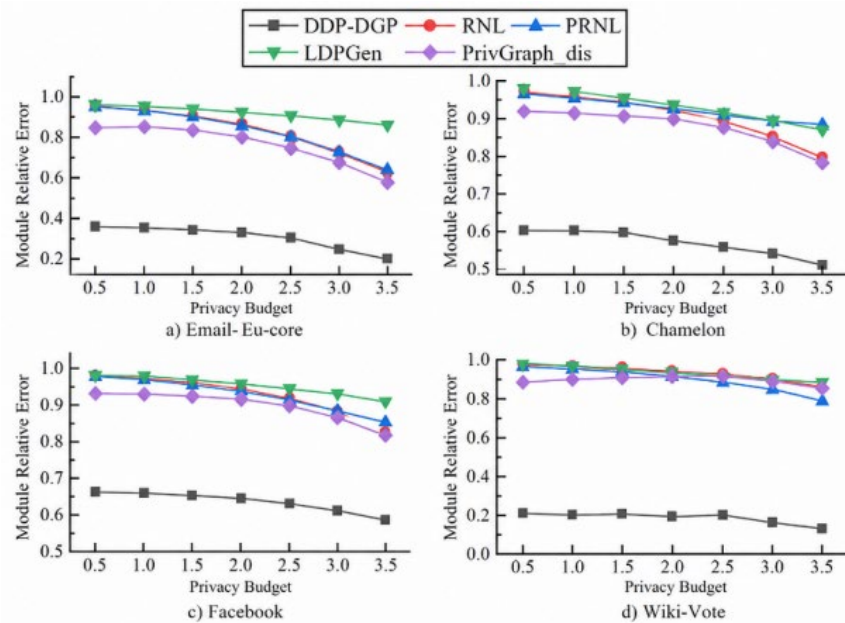


Figure 4. Experimental results of relative error of modularity under different privacy budgets

Figure 5 compares mean absolute error of eigenvector centrality. The displayed curves show a clear advantage for DDP-DGP on Chameleon and Facebook and a generally small error on Wiki-vote, while the gap on email-Eu-core is smaller. These patterns are consistent with degree

calibration and local structural reconstruction preserving influential-node structure. However, the non-monotonic fluctuations in some curves and the absence of run-level uncertainty estimates require a descriptive rather than inferential interpretation.

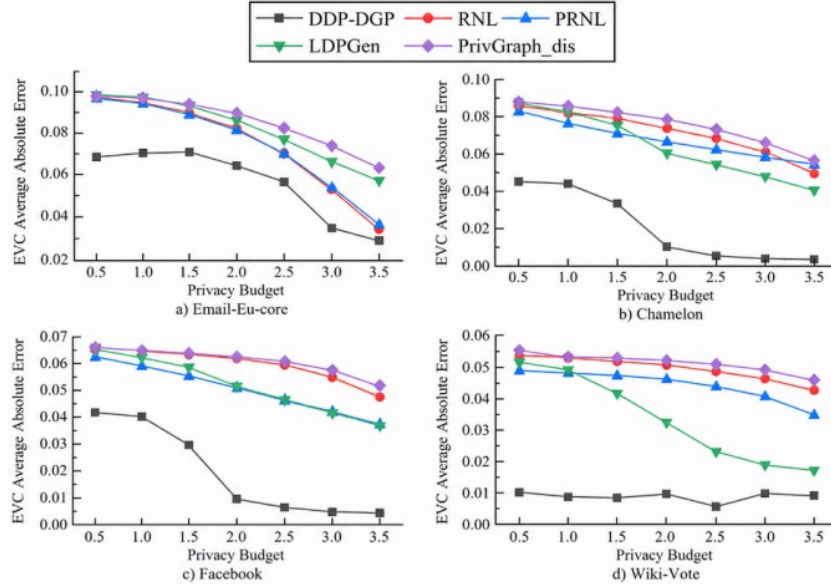


Figure 5. Change of mean absolute error of eigenvector centrality under different privacy budgets

Figure 6 compares average path length with the original graph. DDP-DGP remains comparatively stable across the displayed privacy budgets and is visually closer to the original value in several low-budget settings. The degree-calibrated sampling step provides a mechanism-level

explanation because it limits the large number of false positives produced by bitwise randomized response in sparse graphs. No inferential significance is claimed because run-level uncertainty estimates and LCC sizes are not reported.

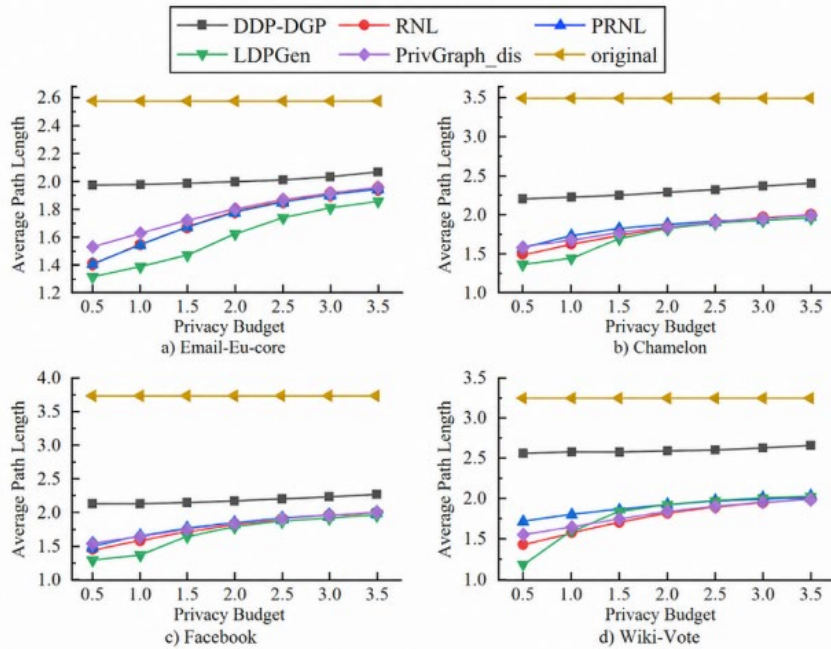


Figure 6. Change of average path length under different privacy budgets

Figure 7 provides a partial ablation focused on community-aware reconstruction: the complete DDP-DGP is compared with a graph based on the sampled privatized adjacency matrix and with a Chung-Lu-only reconstruction without community detection. The complete pipeline has lower modularity error in the displayed bars, indicating that community information contributes to the reported

modularity preservation. This experiment is not an exhaustive component-wise ablation: degree-only perturbation, neighbor-list-only perturbation, dual-granularity perturbation without degree-calibrated sampling, and reconstruction without the Chung-Lu component are not evaluated in the present study.

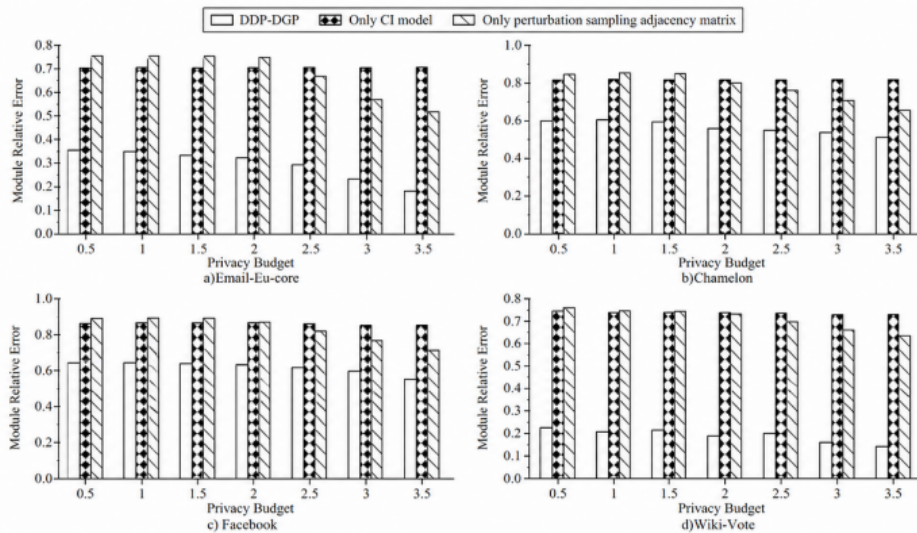


Figure 7. Evaluation of impact of Community Detection in DDP-DGP

Overall, Figures 2-7 suggest that DDP-DGP can preserve several structural properties under the tested privacy settings. The experimental evidence is nevertheless limited to the reported mean curves: run-level uncertainty estimates, an exhaustive component-wise ablation suite, and empirical scaling measurements are not reported. The conclusions are therefore restricted to descriptive structural utility under the evaluated conditions.

These descriptive results motivate DDP-DGP for privacy-preserving graph sharing in decentralized social, recommendation, and edge-collaboration systems. The user-side mechanisms reduce direct disclosure of local relationships, while publisher-side reconstruction operates only on privatized outputs. Application-specific performance should nevertheless be validated on the actual downstream task and under the threat model of the deployment.

4.4. Additional Discussion on Deployment, Downstream Utility, and Extensibility

This subsection further discusses privacy-budget selection, downstream task utility, graph sparsity and scale, and extensibility to more complex distributed graph types. These discussions complement the preceding structural utility experiments and clarify the applicability boundary of the proposed method in distributed graph publishing.

Privacy-budget selection and personalization. The experiments use a global $\epsilon_1:\epsilon_2$ ratio rather than a personalized budget algorithm. Personalized budgets could in principle reflect user sensitivity, degree, role, or application risk, but a heterogeneous allocation changes both privacy accounting and reconstruction bias. The present manuscript therefore treats personalization as future work and does not claim that a personalized allocation strategy has been implemented or validated.

Influence of graph sparsity and scale. The four datasets span different sizes and densities, and the displayed structural metrics suggest that degree calibration is particularly relevant when randomized response creates many false positives in sparse graphs. Nevertheless, asymptotic analysis alone does not establish deployment scalability. Section 3.5 provides sparse and dense storage bounds, while empirical runtime, peak-memory, transmitted-byte, and reconstruction-time scaling is not evaluated in the present study and remains future implementation work.

Downstream-task utility. Degree distribution, modularity, eigenvector centrality, and average path length are useful structural diagnostics, but preservation of these metrics does not by itself demonstrate node-classification, link-prediction, recommendation, community-recovery, or influence-ranking performance. Direct downstream experiments require task-specific labels or splits, model configurations, and repeated-run evaluation. Such experiments are not included in the present study;

consequently, the empirical claim is limited to structural utility, and direct task-level validation is left for future work.

Extension to directed, weighted, dynamic, and attributed graphs. The current method is designed for undirected and unweighted distributed graphs. For directed graphs, the coarse-grained mechanism can be extended from a single degree to in-degree and out-degree perturbation, while the fine-grained neighbor list can be divided into incoming and outgoing adjacency lists. For weighted graphs, edge weights can be discretized and protected through local differential privacy mechanisms for numerical or categorical values, and the Chung-Lu reconstruction probability can be further adjusted by perturbed weight statistics. For dynamic temporal graphs, DDP-DGP can be integrated with dynamic or snapshot-update frameworks such as PPDU and PGU-DNDP [18, 19], with privacy accounting extended across repeated releases rather than assuming an identical update rule across the two methods. For attributed graphs, the structural privacy mechanism can be integrated with attribute perturbation methods such as AsgLDP [21] to jointly protect edge privacy and node attribute privacy.

Limitations of edge-level privacy. This study focuses on edge-level local differential privacy, which protects whether a sensitive connection between two nodes exists. Edge privacy is suitable for many distributed graph-sharing scenarios because users often regard their direct relationships, interactions, or communication links as sensitive. However, edge-level privacy is weaker than node-level privacy. In highly sensitive applications, disclosing the existence of a node together with all its associated edges may reveal stronger personal or organizational information. Node-level privacy usually requires stronger perturbation and may cause larger utility loss. Therefore, extending DDP-DGP from edge-level privacy to joint node-edge-attribute privacy is an important direction for future research.

5. Conclusions

This paper presents DDP-DGP as an integrative dual-granularity framework for privacy-preserving publication of undirected, unweighted decentralized graphs. The privacy model explicitly distinguishes edge adjacency from arbitrary whole-neighbor-vector changes, while the fine-grained allocation $\varepsilon_b = \varepsilon_2 / (N-1)$ ensures that sequential composition yields an ε_2 bound for the complete neighbor list. The degree mechanism uses full output support and boundary-corrected windows; the sampling rate, clipping rules, endpoint reconciliation, inter-community count s_{ab} , and clipped Chung-Lu reconstruction are specified mathematically. Publisher operations use only privatized outputs and therefore constitute post-processing. The complexity analysis gives sparse $O(N + |E| + |E_G|)$ and dense $O(N^2)$ storage characterizations. The four-dataset experiments provide descriptive evidence of structural

utility, while run-level uncertainty estimates, exhaustive component-wise ablation, empirical scaling, and direct downstream-task tests are outside the present evaluation. Future work will extend the framework to node-level, attribute, directed, weighted, and dynamic privacy settings and will evaluate task-level graph-learning utility directly.

References

- [1] Frimpong SA, Han M, Effah EK, Adjei JK, Hanson I, Brown P. A deep decentralized privacy-preservation framework for online social networks. *Blockchain Res Appl*. 2024;5(4):100233.
- [2] European Parliament and Council of the European Union. Regulation (EU) 2016/679 of 27 April 2016 (General Data Protection Regulation). *Off J Eur Union*. 2016;L119:1-88.
- [3] Duchi JC, Jordan MI, Wainwright MJ. Local privacy and statistical minimax rates. In: *Proceedings of the 2013 IEEE 54th Annual Symposium on Foundations of Computer Science*; 26-29 October 2013; Piscataway, NJ: IEEE; 2013. p. 429-438.
- [4] Warner SL. Randomized response: a survey technique for eliminating evasive answer bias. *J Am Stat Assoc*. 1965;60(309):63-69.
- [5] Hay M, Li C, Miklau G, et al. Accurate estimation of the degree distribution of private networks. In: *Proceedings of the 2009 Ninth IEEE International Conference on Data Mining*; 6-9 December 2009; Piscataway, NJ: IEEE; 2009. p. 169-178.
- [6] Hawkins C, Chen B, Yazdani K, et al. Node and edge differential privacy for graph Laplacian spectra: mechanisms and scaling laws. *IEEE Trans Netw Sci Eng*. 2024;11(2):1690-1701.
- [7] Gao Y, Chen XB, Wang L, Ma KG. Distributed graph publishing method based on dual-granularity privacy protection. *J Comput Appl*. 2026;46(7):2164-2173.
- [8] Sun X, Sun L. Privacy preserving large-scale rating data publishing. *EAI Endorsed Trans Scalable Inf Syst*. 2013;1(1):e3.
- [9] Rehman SU, Nawaz A, Ali T, Amin N. g-Sum: a graph summarization approach for a single large social network. *EAI Endorsed Trans Scalable Inf Syst*. 2021;8(32):e2.
- [10] Iftikhar M, Wang Q, Lin Y. dK-microaggregation: anonymizing graphs with differential privacy guarantees. In: *Proceedings of the 2020 Pacific-Asia Conference on Knowledge Discovery and Data Mining*; 11-14 May 2020; Cham: Springer International Publishing; 2020. p. 191-203.
- [11] Gao T, Li F. Protecting social network with differential privacy under novel graph model. *IEEE Access*. 2020;8:185276-185289.
- [12] Huang H, Zhang D, Xiao F, Wang K, Gu J, Wang R. Privacy-preserving approach PBCN in social network with differential privacy. *IEEE Trans Netw Serv Manag*. 2020;17(2):931-945.
- [13] Kang HY, Ji YR, Zhang SX. Enhanced privacy preserving for social networks relational data based on personalized differential privacy. *Chin J Electron*. 2022;31(4):741-751.
- [14] Gao R, Chen XB, Gu Z, et al. Social network graph generation method satisfying personalized differential privacy. *J Taiyuan Univ Technol*. 2024;55(1):163-171.
- [15] Zhang M, Zhou J, Zhang G, Cui L, Gao T, Yu S. APDP: attribute-based personalized differential privacy data publishing scheme for social networks. *IEEE Trans Netw Sci Eng*. 2022;10(2):922-933.

- [16] Li Y, Tao X, Zhang X, Wang M, Wang S. Break the data barriers while keeping privacy: a graph differential privacy method. *IEEE Internet Things J.* 2023;10(5):3840-3850.
- [17] Yuan Q, Zhang Z, Du L, Chen M, Cheng P, Sun M. PrivGraph: differentially private graph data publication by exploiting community information. In: *Proceedings of the 32nd USENIX Security Symposium*; 9-11 August 2023; Berkeley, CA: USENIX Association; 2023. p. 3241-3258.
- [18] Hou L, Ni W, Zhang S, Fu N, Zhang D. PPDU: dynamic graph publication with local differential privacy. *Knowl Inf Syst.* 2023;65(7):2965-2989.
- [19] Gao R, Chen XB, Zhang ZC. Dynamic social network privacy publishing method for partial graph updating. *J Comput Appl.* 2024;44(12):3831-3838.
- [20] Qin Z, Yu T, Yang Y, Khalil I, Xiao X, Ren K. Generating synthetic decentralized social graphs with local differential privacy. In: *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*; 30 October-3 November 2017; New York, NY: ACM; 2017. p. 425-438.
- [21] 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.
- [22] Lin WY, Li BC, Wang C. Towards private learning on decentralized graphs with local differential privacy. *IEEE Trans Inf Forensics Secur.* 2022;17:2936-2946.
- [23] Fu N, Ni W, Hou L, Zhang D, Zhang R. Community detection in decentralized social networks with local differential privacy. *Inf Sci.* 2024;661:120164.
- [24] Guo T, Peng S, Zhang Z, Yang M, Lam KY. Improving the accuracy of locally differentially private community detection by order-consistent data perturbation. In: *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*; 14-18 July 2024; New York, NY: ACM; 2024. p. 1743-1752.
- [25] Li Z, Li RH, Liao M, Jin F, Wang G. Privacy-preserving graph embedding based on local differential privacy. In: *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*; 21-25 October 2024; New York, NY: ACM; 2024. p. 1316-1325.
- [26] Xu W, Qiao P, Liu S, Zheng Z, Cao Y, Li Z. Continuous publication of weighted graphs with local differential privacy. *Proc VLDB Endow.* 2025;18(11):4214-4226.
- [27] Imola J, Murakami T, Chaudhuri K. Communication-efficient triangle counting under local differential privacy. In: *Proceedings of the 31st USENIX Security Symposium*; 10-12 August 2022; Berkeley, CA: USENIX Association; 2022. p. 537-554.
- [28] Dwork C, Roth A. The algorithmic foundations of differential privacy. *Found Trends Theor Comput Sci.* 2014;9(3-4):211-407.
- [29] Yang M, Guo T, Zhu T, Tjuawinata I, Zhao J, Lam KY. Local differential privacy and its applications: a comprehensive survey. *Comput Stand Interfaces.* 2024;89:103827.
- [30] Christofides TC. A generalized randomized response technique. *Metrika.* 2003;57(2):195-200.
- [31] Blondel VD, Guillaume JL, Lambiotte R, Lefebvre E. Fast unfolding of communities in large networks. *J Stat Mech Theory Exp.* 2008;2008(10):P10008.
- [32] Aiello W, Chung F, Lu L. A random graph model for massive graphs. In: *Proceedings of the Thirty-Second Annual ACM Symposium on Theory of Computing*; 21-23 May 2000; New York, NY: ACM; 2000. p. 171-180.
- [33] Chung F, Lu L. Connected components in random graphs with given expected degree sequences. *Ann Comb.* 2002;6(2):125-145.