

PrivAGS: Differentially Private Attributed Graph Synthesis

SHUZHAN YE, Zhejiang University, China

LU CHEN*, Zhejiang University, China

ZHIKUN ZHANG, Zhejiang University, China

YUNJUN GAO, Zhejiang University, China

YUXIANG WANG, Hangzhou Dianzi University, China

XIAOLIANG XU, Hangzhou Dianzi University, China

Attributed graphs are extensively utilized in marketing, friend recommendations, disease prediction, etc. In attributed graphs, nodes are associated with attributes to enrich the graph representation, while edges indicate relationships between nodes. However, ensuring data privacy when publishing attributed graphs is a significant challenge due to the sensitive nature of both attributes and relationships. Existing methods fail to preserve graph structures effectively and neglect correlations among node attributes, leading to diminished utility for published synthetic graphs. To address these issues, we propose PrivAGS, a framework for publishing attributed graphs with Rényi Differential Privacy (RDP) guarantees. PrivAGS reconstructs graph structures and attributes based on community structures to capture tightly connected features. We propose a bounded Gaussian threshold mechanism to preserve attribute correlations and utilize probabilistic graph models with optimized inference structures to infer distributions and release node attributes. Additionally, PrivAGS introduces a new structural model, MCEG, to capture clustering structures and enable efficient graph reconstruction. Extensive experiments on five real-world datasets show that PrivAGS generates privacy-preserving, high-utility synthetic data.

CCS Concepts: • **Security and privacy** → **Data anonymization and sanitization**.

Additional Key Words and Phrases: Rényi differential privacy, synthetic graph

ACM Reference Format:

SHUZHAN YE, LU CHEN, ZHIKUN ZHANG, YUNJUN GAO, YUXIANG WANG, and XIAOLIANG XU. 2025. PrivAGS: Differentially Private Attributed Graph Synthesis. *Proc. ACM Manag. Data* 3, 6 (SIGMOD), Article 351 (December 2025), 25 pages. <https://doi.org/10.1145/3769816>

1 Introduction

Large-scale, real-world attributed graphs have been extensively utilized across various domains, such as social networks [20], email networks [21], voting networks [29], etc. In attributed graphs, nodes are usually associated with attributes to enrich the graph information (e.g., user nodes in social networks may possess attributes including age, location, interests, and level of education), enabling more effective analysis. However, the associated attributes can be highly sensitive, such as religion, interests, or sexual orientation. Furthermore, attributed graphs encode complex relationships between individuals (e.g., friendships, acquaintances, sexual relationships), many of which are

Authors' Contact Information: SHUZHAN YE, yeshuzhan123@zju.edu.cn, Zhejiang University, China; LU CHEN, luchen@zju.edu.cn, Zhejiang University, China; ZHIKUN ZHANG, zhikun@zju.edu.cn, Zhejiang University, China; YUNJUN GAO, gaoyj@zju.edu.cn, Zhejiang University, China; YUXIANG WANG, lsswyx@hdu.edu.cn, Hangzhou Dianzi University, China; XIAOLIANG XU, xxl@hdu.edu.cn, Hangzhou Dianzi University, China.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

© 2025 Copyright held by the owner/author(s). Publication rights licensed to ACM.

ACM 2836-6573/2025/12-ART351

<https://doi.org/10.1145/3769816>

sensitive. Even the disclosure of less sensitive attributes can assist adversaries in identifying the true identities of nodes and revealing their friendship connections. A classic approach to protecting the privacy of graph analysis is anonymization [24, 51, 52], which conceals the personally identifiable information of the nodes. However, prior research has demonstrated that when combined with auxiliary information, anonymized graphs can be easily de-anonymized by attackers [2, 30].

To address the limitations of anonymization, *differential privacy* (DP) [11] was introduced and has become the gold standard for privacy protection. It has been widely adopted by companies and government agencies for data privacy-preserving analysis. For example, Uber uses Flex [18] to securely answer SQL queries, while LinkedIn’s Pinot [36] enables analysts to gain insights into member engagement.

In the context of differential privacy, there are two primary strategies for safeguarding the privacy of graph data: (1) designing tailored DP algorithms for specific statistical properties, such as degree distribution [14] or clustering coefficients [43], and (2) generating synthetic graphs to replace the original data. The generation offers two significant advantages: (1) it eliminates the need for extensive domain expertise to analyze the privacy requirements of each specific algorithm, and (2) it reduces the necessity of disclosing proprietary technical details when collaborating with data owners. Motivated by these considerations, we focus on the generation of synthetic attributed graphs within the framework of DP.

Existing Solutions. Previous research [6, 8, 19, 31, 40, 45, 46] on differentially private attributed graph synthesis has predominantly treated structural and attribute features as independent components, fundamentally overlooking the intrinsic homophily properties where structural connectivity and attribute similarity are deeply intertwined. This compartmentalized approach severely limits the synthetic graph’s ability to capture real-world data characteristics, diminishing practical utility. Furthermore, privacy-preserving high-dimensional attribute publishing presents a fundamental trade-off between dimensionality and utility. Univariate distributions fail to preserve essential correlations, while high-dimensional joint distributions introduce computational intractability and excessive noise, ultimately compromising synthetic graph fidelity. These limitations manifest as three critical challenges.

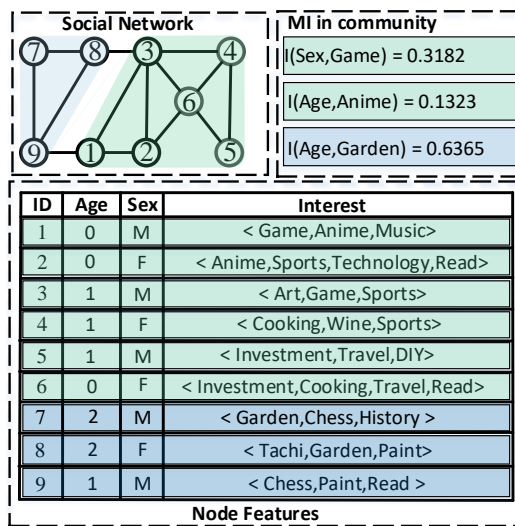


Fig. 1. Motivating example

Challenges I: How to effectively capture and synthesize tightly connected features in graphs? Real-world networks are inherently characterized by tightly connected features, such as community structures and clustering patterns, which represent import interactions among nodes and highlight the need for synthesis methods that preserve tightly connected features while scaling to global topology. In distinction to group in group privacy [25], communities are subgraph structures consisting of densely interconnected nodes where connection density within the community typically exceeds that between the community and external nodes without relying on shared attributes such as node attributes or common neighbors [32, 37], serving as natural units for capturing tightly connected features. As prior research indicates that treating dense subgraph as the generation granularity better preserves deep topological information [6, 45, 46]. This insight inspired our design to leverage communities as the generation granularity for enhanced fidelity and efficiency. For instance, in the Figure 1, the graph partitions into two distinct communities: $\{1, 2, 3, 4, 5, 6\}$ and $\{7, 8, 9\}$, demonstrating how these structures encapsulate local patterns that global approaches might overlook. Another related core difficulty is preserving clustering characteristics, which arise from triangular relationships formed by closed and open triplets. Existing methods often rely on privacy-preserving computations of closed triplets followed by post-generation adjustments, but this overlooks open triplets' contributions to clustering accuracy and incurs high computational costs due to their sparse distribution. Motivated by these limitations, we developed the MCMC-Based Cohesive Edge Generation (MCEG) model, drawing from the insight that triangle is formulated by open triplets, and model edge generation as a cohesion-aware Markov chain—with state transitions encoding open triplet probabilities via discretized bucketing. This ensures natural features reconstruction with theoretical guarantees of convergence while maintaining efficiency over explicit adjustments.

Challenge II: How to publish high-dimensional attribute data while preserving attribute correlations and logical patterns? Publishing high-dimensional attributes requires balancing privacy and utility with the preservation of intricate correlations, as attributes in real networks are interdependent and often amplified by structural elements like communities—e.g., reflecting logical patterns such as how title and workclass influence income in financial networks [47]. As depicted in the Motivating Example (Figure 1), the mutual information (MI) between attributes like Age and Garden increases from 0.52 (MI over all the graph) to 0.63 (MI within community $\{7, 8, 9\}$), underscoring communities' role in revealing stronger attribute correlations and guiding our decision to use them as the publishing granularity to capture amplified correlations while mitigating re-identification risks. The dual challenge involves privately identifying meaningful dependencies and inferring accurate marginals in high-dimensional spaces. Traditional approaches [8, 19] sample from univariate marginals, disrupting correlations and failing to maintain logical patterns [47, 50]. To overcome this, our solution integrates two innovations: i) the Bounded Gaussian Threshold Mechanism (BGTm) detects pairwise dependencies without splitting the privacy budget across all $\binom{d}{2}$ pairs—a flaw that amplifies noise in prior methods, ensuring privacy guarantee while providing explicit correlation standards; and ii) for marginal inference, based on noise analysis, we propose an inferential structure termed optimized inference structure (OIS), to support marginal distribution reasoning in high-dimensional spaces. We prove that finding this structure is NP-hard and solve it using integer programming relaxation and the difference of convex algorithm.

Challenge III: How to capture the intrinsic patterns between graph structure and attributes? Previous approaches primarily focus on conventional pattern modeling that considers structure and attributes separately. However, attributed graphs exhibit fundamental homophily properties where structural connectivity and attribute similarity are deeply intertwined—nodes with high homophily properties are more likely to be connected [48]. The key insight is that edge formation is not random but

follows patterns based on both structural connectivity and attribute similarity. This motivates our development of composite cohesiveness, a unified metric that captures the intrinsic patterns between graph structure and attributes.

Our contributions are summarized as follows.

- We propose PrivAGS, an efficient and effective attributed graph synthesis framework that generates synthetic attributed graphs satisfying RDP under edge-level while preserving key structural and attribute features.
- We propose the BGTm, which effectively captures privacy-preserving correlations in high-dimensional spaces. Furthermore, leveraging insights from noise analysis, we propose a novel inference structure that reduces noise interference in high-dimensional distributional inference, thereby enhancing the practical utility of the published attributes.
- We develop composite cohesiveness as a unified measure of edge affinity that captures both structural connectivity and attribute similarity, providing the theoretical foundation for MCEG's edge generation process that naturally preserves clustering and homophily patterns.
- We conduct extensive experiments on five real-world attributed graph datasets under multiple evaluation metrics to demonstrate the effectiveness and efficiency of PrivAGS.

2 Related Work

This section provides an overview of existing work on synthetic graph publishing under differential privacy.

2.1 Graph Structure Publishing

Many studies focus on publishing synthetic graph structures with privacy guarantees. Privacy concepts in graph structure generation primarily involve node adjacency [17, 49] and edge adjacency [6, 8, 19, 26, 31, 33, 34, 40, 45, 46].

2.1.1 Node Adjacency-based Publishing. Node adjacency defines neighboring datasets based on node differences, ensuring protection. PrivCom [49] adds noise to low-dimensional vectors, generating noisy Katz matrices. Jian et al. [17] achieve node-DP via edge deletion and random node insertion.

2.1.2 Edge Adjacency-based Publishing. Edge adjacency considers the difference in individual edges as the variation between datasets. This approach is divided into two methods: adjacency matrix-based and parameterized graph model-based.

Adjacency Matrix-based. Graph edges are represented as an $n \times n$ adjacency matrix, with perturbations ensuring privacy. TmF [31] adds Laplace noise to matrix entries, selecting the top m noisy entries as edges. DER [6] reorders nodes and perturbs regions using quadtree density to reconstruct the graph. PrivGraph [46] reduces noise impact by aggregating structures with community features.

Parameterized Graph Model-based. Graph structures are generated by extracting topological parameters like degree distribution, triangle count, and clustering features. The Chung-Lu model [9] generates edges by perturbing node degrees based on degree distributions. ERGMs [26] and Kronecker graphs [40] use structural estimation and recursive Kronecker products, respectively. However, ERGMs and Kronecker models fail to capture clustering features. Jorgensen et al. [19] propose the AGM-Tricycle model, which adjusts structure using triangle count, and CPGM [8] improves the capture of clustering features via community structures.

The above methods are ineffective in simultaneously capturing the graph's structural and attribute features. We aim to capture both of these aspects.

2.2 Graph Attribute Publishing

Node attributes resemble tabular data. Thus, we also consider tabular data generation with privacy guarantees. Previous studies [8, 16, 19, 49] assume attribute independence and use noisy univariate distributions, neglecting attribute correlations. Recent studies [7, 27, 35, 47, 50] utilize multivariate joint distributions for attribute release. For instance, PriView [35] approximates high-dimensional distributions using k -dimensional joint distributions for efficient sampling. PrivBayes [47] constructs k -dimensional dependency graphs based on mutual information and applies the chain rule for sampling. PGM [27] employs probabilistic graphical models for joint distribution inference, while PrivSyn [50] captures pairwise correlations with low global sensitivity and uses a gradual update method for release. However, these methods assume that attributes are wholly independent or introduce significant noise variance in capturing their correlations.

3 PRELIMINARIES

3.1 Differential Privacy

Differential Privacy (DP) [11] is a framework for scenarios involving a trusted data curator. This curator collects data from individual users, processes it in a way that satisfies the principles of DP, and then publishes the results. Intuitively, the concept of differential privacy ensures that the influence of any single element in the dataset on the output remains strictly limited.

DEFINITION 1. ϵ -Differential Privacy (varepsilon-DP). An algorithm \mathcal{A} satisfies ϵ -differential privacy, where $\epsilon > 0$, if and only if for any two neighboring datasets D and D' , we have

$$\forall T \subseteq \text{Range}(\mathcal{A}) : \Pr[(\mathcal{A}(D) \in T)] \leq e^\epsilon \cdot \Pr[\mathcal{A}(D') \in T]$$

3.2 Rényi Differential Privacy

Differential Privacy (DP) [11] ensures data privacy by limiting the impact of any single sample on the output through noise addition. However, its strict definition can overly reduce data utility, especially in complex tasks. To address it, Rényi Differential Privacy (RDP) [28] refines privacy loss measurement using Rényi divergence, offering greater flexibility and high precision. Formally, RDP is defined as follows.

DEFINITION 2 (RÉNYI DIFFERENTIAL PRIVACY). An algorithm \mathcal{A} satisfies (α, ϵ) -RDP, where $\alpha > 1$ and $\epsilon > 0$, if for any two neighboring datasets D and D' , the Rényi divergence of order α between the distributions of $\mathcal{A}(D)$ and $\mathcal{A}(D')$ is bounded by ϵ . Formally,

$$D_\alpha(\mathcal{A}(D) \parallel \mathcal{A}(D')) = \frac{1}{\alpha - 1} \log \mathbb{E}_{x \sim \mathcal{A}(D')} \left[\left(\frac{\Pr[\mathcal{A}(D) = x]}{\Pr[\mathcal{A}(D') = x]} \right)^\alpha \right] \leq \epsilon.$$

where $\text{Range}(\mathcal{A})$ denotes the set of all possible outputs of algorithm \mathcal{A} . We consider two data sets D and D' to be neighbors, denoted as $D \simeq D'$ when D' is derived from D by adding, removing, or modifying a single data tuple.

Gaussian Mechanism. Gaussian mechanism (GM) satisfies the RDP requirements by adding random Gaussian noise to the aggregated results. The magnitude of the noise depends on GS_f , i.e., global sensitivity,

$$GS_f = \max_{D \simeq D'} \|f(D) - f(D')\|_2,$$

where f denotes the aggregation function, and D or D' is the user data. When f outputs a scalar, the Gaussian mechanism \mathcal{A} is as:

$$\mathcal{A}_f(D) = f(D) + \mathcal{N}(0, \sigma^2),$$

Table 1. Summary of Notation

σ	Standard deviation of Gaussian distribution
α	Order of Rényi differential privacy
ε	Privacy budget
n	Node number of the graph
m	Edge number of the graph
d	Attribute dimension of the graph
D	Node normalized sampling probability
W	Sum edge weight of graph
$W_{i,j}$	Weight of the edge (i, j)
ϕ	Potential function
A_G	Attribute set of all dimensions
$A_G(\cdot)_i$	The i -th dimension attribute
$A_G(v)_i$	The i -th dimension attribute value of node v
$P(A_G)$	Probability distribution of node attributes
Π	Composite cohesiveness probability distribution
n_b	Bucket number of distribution
$N(u)$	Neighbor node set of node u
$f(u, v)$	Composite cohesiveness between nodes u and v
$\Pi(f(u, v))$	Probability of $f(u, v)$

where $\mathcal{N}(0, \sigma^2)$ stands for a random variable sampled from the Gaussian distribution with the scale parameter $\sigma^2 = \frac{(GS_f)^2 \alpha}{2\varepsilon}$. When f outputs a vector, \mathcal{A} adds independent samples of $\mathcal{N}(0, \sigma^2)$ to each element of the vector.

Although the Gaussian mechanism is widely adopted for its flexibility, its application to unbounded data requires additional consideration of global sensitivity, rendering it unsuitable for scenarios with constrained result ranges. By bounding the domain and output of the Gaussian mechanism, users can achieve enhanced privacy guarantees and more controlled result ranges [10, 23].

Bounded Gaussian Mechanism. Let the bounded query result s be in $[c_{kl}, c_{km}]_{k=\{1, \dots, r\}}$, where r denotes the number of elements in the query result and $[c_{kl}, c_{km}]_{k=\{1, \dots, r\}}$ be the bound of the k -th result. The bounded gaussian mechanism of order 2 generates the noisy result $s^* \in [c_{kl}, c_{km}]$ with ε -DP by drawing from the following probability density function:

$$f(s^* | c_{kl} \leq s^* \leq c_{km}, k = \{1, \dots, r\}) = \prod_{k=1}^r \frac{2 \exp\left\{-\frac{|s_k^* - s_k|}{\sigma}\right\}^2}{2\sigma \Gamma\left(\frac{1}{2}\right) A(s_k, \sigma, 2)}$$

where $\sigma \geq (2\varepsilon^{-1} (\sum_{k=1}^r \binom{2}{j} |c_{kl} - c_{km}|^{2-j} \Delta_{1,k}^j + \Delta_2^2))^{\frac{1}{2}}$, $A(s_k, \sigma, 2) = \Pr(c_{kl} \leq s_k^* \leq c_{km}; s_k, \sigma, 2) = (\Gamma(\frac{1}{2}))^{-1} (\gamma[\frac{1}{2}, (c_{kl} - s_k)/\sigma] + \gamma[\frac{1}{2}, (s_k - c_{kl})/\sigma])$ (γ is the lower incomplete gamma function), $\Delta_{1,k}$ is the global sensitivity L_1 of s_k , and Δ_2 is the global sensitivity L_2 of s .

Composition Properties of RDP. The following RDP composition properties are commonly used to build complex differentially private algorithms from simpler subroutines.

- **Sequential Composition.** Combining k subroutines that satisfy (α, ε_i) -RDP ($i \in \{1, \dots, k\}$) results in a mechanism satisfying (α, ε) -RDP, where $\varepsilon = \sum_{i=1}^k \varepsilon_i$.
- **Parallel Composition.** Given k algorithms operating on disjoint subsets, each satisfying (α, ε_i) -RDP ($i \in \{1, \dots, k\}$), the result satisfies RDP for $(\alpha, \max_{i=1}^k \varepsilon_i)$.

- **Post-processing.** Given an (α, ϵ) -RDP algorithm \mathcal{A} , releasing $g(\mathcal{A}(D))$ for any function g still satisfies (α, ϵ) -RDP. In other words, post-processing the output of an RDP algorithm does not incur additional privacy loss.

PROPOSITION 1. *If \mathcal{A} satisfies (α, ϵ) -RDP, then \mathcal{A} is $(\epsilon + \frac{\log(1/\delta)}{\alpha-1}, \delta)$ -DP for any $\delta > 0$ [28].*

Proposition 1 shows that for a fixed privacy budget ϵ and δ , (α, ϵ) -RDP provides a tighter bound on the cumulative privacy loss under composition, making it more suitable for algorithms consist of a large number of tasks.

3.3 Differential Privacy for Attributed Graph

The definition of neighboring datasets underpins differential privacy, determining both the target and strength of privacy protection. Differential privacy was originally defined in the context of tabular data, and thus, neighboring datasets differ in the presence of an individual (i.e., row) [11, 12]. To apply differential privacy into the graph analysis, it is essential to establish a precise notion of what it means for two attributed graphs to be neighbors [3].

DEFINITION 3 (ATTRIBUTED GRAPH). *An attributed graph is defined as $G = (V_G, E_G, A_G)$, where $V_G(E_G)$ is the set of nodes (edge) and A_G is the set of attributes associated with nodes. For each $v \in V_G$, it has a set of attributes $A_G = \{a_1, a_2, \dots, a_d\}$, where d is the number of attributes, $A_G(v)_i$ indicate the i -th attribute a_i of node v .*

For categorical attributes, we employ numerical representations, while numerical attributes are discretized. Specifically, we define the domain for age using distinct values (e.g., 0, 1, 2), where 0 corresponds to the age range [12, 25], 1 to [26, 50], and 2 to ages above 50. To address the limitations of using distinct values for attributes such as sex and interest, we introduce binary representations: “female” (F) and “male” (M) for sex, and seventeen binary attributes for interest (e.g., $\langle \text{sports, Investment, Travel} \rangle$, $\langle \text{Sports, Investment, Beauty care} \rangle$, etc.), each with a domain of $\{0, 1\}$. Consequently, the total number of attributes is $d = 19$. Here, $A_G = \{age, sex, \langle Game, Anime, Music \rangle, \dots, \langle History, Taichi, Read \rangle\}$, and $A_G(1) = [0, 0, 1, 1, \dots, 0]$.

We extend the concept of neighboring datasets from the traditional edge adjacency [3] to the more comprehensive **Dimension Attribute and Edge (DAE) Neighboring**, along with its associated definitions of differential privacy.

DEFINITION 4 (DIMENSION ATTRIBUTE AND EDGE NEIGHBORING). *Given an attributed graph $G = (V_G, E_G, A_G)$, an attributed graph $G' = (V_{G'}, E_{G'}, A_{G'})$ is dimension attribute and edge neighbor of G if G and G' are distinguished solely by one edge and a single node attribute. We use \sim to denote neighboring graphs and ∇ to represent set differences, with the mathematical definition:*

$$G \sim G' \Leftrightarrow |E_G \nabla E_{G'}| = 1 \cap |A_G \nabla A_{G'}| = 1 \quad (1)$$

DEFINITION 5 ((α, ϵ) -DAE RDP). *An algorithm \mathcal{A} satisfies the (α, ϵ) -DAE RDP, where $\alpha > 1$ and $\epsilon > 0$, if and only if for any two neighboring DAE graphs G and G' ,*

$$D_\alpha(\mathcal{A}(G) \parallel \mathcal{A}(G')) \leq \epsilon$$

where $D_\alpha(\cdot \parallel \cdot)$ denotes the Rényi divergence of order α between the output distributions of the algorithm \mathcal{A} on graphs G and G' . The parameter α controls the trade-off between privacy loss and utility, and ϵ represents the privacy budget.

4 Overview of PrivAGS

We proceed to introduce the overview of our solution PrivAGS. Figure 2 illustrates the framework of PrivAGS, which consists of community detection (Appendix A [1]), attribute synthesis (Sec. 5), and graph reconstruction (Sec. 6).

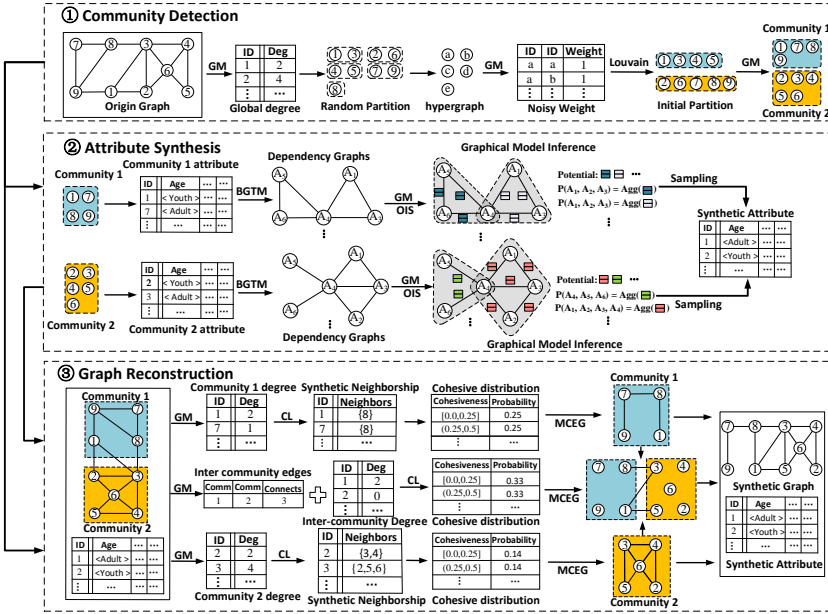


Fig. 2. Framework of PrivAGS

Stage 1: Community Detection. We design a community detection algorithm to partition nodes efficiently. First, we determine the number of supernodes based on graph density and global noisy degree mappings. Nodes are then randomly assigned to supernodes, and noisy connections between supernode pairs form weighted superedges and a weighted graph. Using this weighted graph, PrivAGS performs community partitioning with the Louvain algorithm [4]. Finally, PrivAGS refines the partition using noisy edge connections between nodes and communities. Due to the limited space, further details are provided in [1] Appendix A.

Stage 2: Attribute Synthesis. We use the community partitions obtained from Stage 1 as the basis for attribute synthesis. For each community, we extract the corresponding attribute sets and construct an attribute dependency graph using the Bounded Gaussian Threshold Mechanism (BGTm). After that, we employ the optimized inference structure (OIS) with minimal noise power as the inference structure of the probabilistic graphical model to infer attribute distribution and release node attributes.

Stage 3: Graph Reconstruction. In graph reconstruction, community partitions and synthetic attributes are used with the Gaussian mechanism to extract inter-community node degree mappings within each community. Synthetic neighbor relationships are generated through these mappings and attributes from Stage 2 to calculate the community's cohesiveness distribution. This distribution, combined with the degree mappings, allows MCEG to reconstruct the internal community structure. For the intra-community structure, PrivAGS computes intra-community node degree mappings using global degree mappings from Stage 1 and inter-community mappings. It then generates the required MCEG parameters and applies them to create edges between communities.

Privacy Budget Analysis. Our approach includes three steps: community partitioning, attribute synthesis, and graph reconstruction. The community partitioning uses ϵ_1 . Attribute synthesis involves dependency graph construction (ϵ_2) and attribute generation (ϵ_3), while optimal inference structure requires no budget. Graph reconstruction consumes ϵ_4 . Thus, the total privacy budget is $\epsilon = \epsilon_1 + \epsilon_2 + \epsilon_3 + \epsilon_4$.

THEOREM 1. *Our solution satisfies (α, ϵ) -DAE RDP, where $\epsilon = \epsilon_1 + \epsilon_2 + \epsilon_3 + \epsilon_4$.*

Theorem 1 guarantees that PrivAGS generates synthetic graphs meeting the (α, ϵ) -DAE RDP privacy requirements. A full analysis and proof can be found in Appendix D [1].

Time complexity analysis. The algorithm's time complexity emerges from three core phases. Community detection combines node ranking ($O(n \log n)$), hypergraph processing ($O(m + n\sqrt{m})$), and pairwise noise injection across C_a communities ($O(C_a^2)$). Attribute synthesis involves mutual information computation ($O(d^2 n)$), dependency graph optimization ($O(C_a T m_a^3)$), and attribute generation ($O(C_a d^2)$). Graph reconstruction integrates intra-community structures ($O(n + m)$) and inter-community edges ($O(C_a^2 + m_{\text{inter}})$). The aggregated time complexity is:

$$O\left(n \log n + n\sqrt{m} + m + C_a^2 + d^2 n + C_a T m_a^3 + C_a d^2 + m_{\text{inter}}\right),$$

where n , m_a , d , and C_a denote the total node count, average dependency graph edges, attribute dimensionality, and community number, respectively. Practical efficiency is maintained through $m_a, d \ll n$ and $C_a \ll n$. The theoretical $O(n^2)$ worst case (when $C_a = n$) remains negligible in real-world scenarios.

5 Attribute Synthesis

5.1 Dependency Graph Construction

Modeling dependencies among attributes is essential for generating high-utility synthetic data under differential privacy. While saturated log-linear models have been employed to capture full joint distributions through high-order interactions [7], such high-order correlations tend to diminish in strength and interpretability as dimensionality increases [15]. Thus, we focus on modeling pairwise dependencies. Existing methods partition the privacy budget across $\binom{d}{2}$ attribute pairs and add noise to each correlation, but suffer from two limitations: the lack of principled correlation thresholds, and severe noise accumulation as dimensionality grows.

To overcome these challenges, we propose the Bounded Gaussian Threshold Mechanism (BGTm), a differentially private mechanism designed to detect significant pairwise dependencies in high-dimensional settings without partitioning the privacy budget. Instead of estimating correlation values directly and comparing them in the presence of independent noise, BGTm reformulates the problem as a binary threshold decision task. It first adopts mutual information (MI) to capture nonlinear dependencies [41], and then, drawing on insights from Gaussian copula theory [5], it derives a statistically grounded MI threshold by mapping from Kendall's τ , a more interpretable measure of rank correlation. Both the MI values and the derived threshold are perturbed using bounded Gaussian noise. This enables BGTm to perform a series of noisy threshold comparisons, thereby outputting a binary vector that represents the existence or absence of dependency between each attribute pair.

PrivAGS employs mutual information (MI) to capture nonlinear dependencies robustly [41]. Through copula functions [39], MI converts to Kendall's tau using joint probability density as the product of copula density and marginal PDFs [5], providing correlation standards for MI. Under the Gaussian copula, the mutual information $I(A_G(\cdot)_k, A_G(\cdot)_l)$ and Kendall's tau $\tau(A_G(\cdot)_k, A_G(\cdot)_l)$ between variables $A_G(\cdot)_k$ and $A_G(\cdot)_l$ are computed as follows:

$$I(A_G(\cdot)_k, A_G(\cdot)_l) = -\frac{1}{2} \log(1 - \theta^2) \quad (2)$$

$$\tau(A_G(\cdot)_k, A_G(\cdot)_l) = \frac{2}{\pi} \sin^{-1}(\theta) \quad (3)$$

where θ is the correlation parameter of copula function.

According to Equations 2 and 3, we derive the MI threshold η for a given τ , providing statistically meaningful criteria for dependency identification:

$$\eta = -\frac{1}{2} \log \left(1 - \sin^2 \left(\frac{|\tau|\pi}{2} \right) \right). \quad (4)$$

The key insight is that when learning binary threshold comparisons rather than exact query values, the privacy budget need not be partitioned across queries. This approach is viable when a single tuple difference affects all threshold comparisons uniformly. We now formally prove BGTM's privacy guarantee.

THEOREM 2. *Bounded Gaussian threshold mechanism satisfies the (α, ϵ_2) -DAE RDP.*

PROOF. Let $w = \binom{d}{2}$ denote the number of node pairs, the BGTM outputs a vector $v = [v_1, v_2, \dots, v_w]$, where $v_i = 1$ if the i -th pair of attribute vectors is correlated, and $v_i = 0$ otherwise. Let $P(v = b)$ and $P'(v = b)$ represent the probabilities of vector $b \in \{0, 1\}^w$ in adjacent datasets D and D' , respectively. We prove that for all such datasets and vectors b :

$$D_\alpha(P||P') = \frac{1}{\alpha - 1} \log \mathbb{E}_{b \sim P'} \left[\left(\frac{P(v = b)}{P'(v = b)} \right)^\alpha \right] \leq \epsilon_2$$

Let $v^{<i}$ denote the first $i - 1$ elements in v . The likelihood ratio decomposes as:

$$\frac{P(v = b)}{P'(v = b)} = \frac{\prod_{i=1}^w P(v_i = b_i | v^{<i})}{\prod_{i=1}^w P'(v_i = b_i | v^{<i})} = \prod_{i:b_i=1} \frac{P(v_i = 1 | v^{<i})}{P'(v_i = 1 | v^{<i})} \cdot \prod_{i:b_i=0} \frac{P(v_i = 0 | v^{<i})}{P'(v_i = 0 | v^{<i})} \quad (5)$$

For mutual information bounded in $[0, \log N]$, the bounded Gaussian mechanism yields PDFs:

$$f_{\text{bound}}^{I_i^*, D}(I_i^* | I_i, \sigma_i^2) = \begin{cases} \frac{2 \exp\left(-\frac{(I_i^* - I_i)^2}{2\sigma_i^2}\right)}{2\sigma_i \sqrt{2\pi A}(I_i, \sigma_i)}, & \text{if } I_i^* \in [0, \log N] \\ 0, & \text{otherwise} \end{cases}$$

When $v^{<i}$ is fixed, $v_i = 0$ if and only if $I_i^* \leq \eta^*$, where η^* is the noisy threshold. Let $H_i(\eta^*) = P(v_i = 0 | v^{<i})$. Then:

$$H_i(\eta^*) = \int_0^{\eta^*} f_{\text{bound}}^{I_i^*, D}(I_i^* | I_i, \sigma_i^2) dI_i^*$$

For neighboring dataset D' with mutual information I_i' , we have $H_i'(\eta^*) = \int_0^{\eta^*} f_{\text{bound}}^{I_i^*, D'}(I_i^* | I_i', \sigma_i^2) dI_i^*$.

The bounded Gaussian mechanism satisfies: when $\sigma_i \geq \sqrt{\frac{4(\log N \cdot \Delta I + \Delta I)}{\epsilon_b}}$, we have $\frac{f_{\text{bound}}^{I_i^*, D}}{f_{\text{bound}}^{I_i^*, D'}} \leq \exp(\epsilon_b/2)$ (see more details in [23] Appendix C), implying $H_i(\eta^*) \leq \exp(\epsilon_b/2) H_i'(\eta^*)$. Since η^* follows bounded Gaussian distribution with scale σ_η :

$$\begin{aligned} \prod_{i:b_i=0}^w P(v_i = 0 | v^{<i}) &= \int_0^{\eta^*} p[\eta^* = x] \prod_{i:b_i=0} H_i(x) dx \\ &\leq \exp\left(\frac{\epsilon_b}{2}\right) \int_0^{\eta^*} p[\eta^* = x] \prod_{i:b_i=0} H_i'(x) dx \\ &= \exp\left(\frac{\epsilon_b}{2}\right) \prod_{i:b_i=0}^w P'(v_i = 0 | v^{<i}). \end{aligned} \quad (6)$$

Similarly, $\prod_{i:b_i=1} P(v_i = 1|v^{<i}) \leq \exp(\varepsilon_b/2) \prod_{i:b_i=1} P'(v_i = 1|v^{<i})$. Therefore: $\frac{P(v=b)}{P'(v=b)} \leq \exp(\varepsilon_b)$. The Rényi divergence becomes:

$$\begin{aligned} D_\alpha(P||P') &= \frac{1}{\alpha-1} \log \mathbb{E}_{b \sim P'} \left[\left(\frac{P(b)}{P'(b)} \right)^\alpha \right] \\ &\leq \frac{1}{\alpha-1} \log(\exp(\alpha\varepsilon_b)) = \frac{\alpha\varepsilon_b}{\alpha-1} \end{aligned} \quad (7)$$

Setting $\frac{\alpha\varepsilon_b}{\alpha-1} = \varepsilon_2$ gives $\varepsilon_b = \frac{(\alpha-1)\varepsilon_2}{\alpha}$ and $\sigma_i^2 = \sigma_\eta^2 = \frac{4\alpha(\log N \cdot \Delta I + \Delta I)}{(\alpha-1)\varepsilon_2}$, ensuring the dependency graph satisfies (α, ε_2) -RDP. \square

Algorithm 1 constructs dependency graphs using the BGTM. It calculates the MI threshold η via Eq. 4 (line 1), identifies the largest attribute domain N in A_G (line 2), and sets $\sigma = \frac{4\alpha(\log N \cdot \Delta I + \Delta I)}{(\alpha-1)\varepsilon_2}$ based on Theorem 2 for (α, ε_2) -DAE RDP. For each community C_a , it initializes M_a with nodes $V_{M_a} = \{A_i : 1 \leq i \leq d\}$ and empty edges E_{M_a} (lines 5–7), applies the bounded generalized Gaussian mechanism for noisy threshold η^* , computes noisy mutual information I^* for each pair (A_k, A_l) , and adds edges where $I^* \geq \eta^*$ (lines 8–11). It returns $M = \cup_{C_a \in \mathcal{C}_\mathcal{P}} \{M_a\}$ (line 12).

Algorithm 1: Dependency graph construction

Input: Community partition $\mathcal{C}_\mathcal{P}$, graph attribute set $A_G = (A_1, A_2, \dots, A_d)$, privacy budget ε_2 , Kendall's tau threshold $\tau = 0.8$, the order of Rényi divergence α

Output: Dependency graph sequence M

```

1  $\eta = -\frac{1}{2} \log(1 - \sin^2(\frac{|\tau|\pi}{2}))$ 
2 Identify the largest attribute domain  $N$  within  $A_G$ 
3 set  $\sigma = \frac{4\alpha(\log N \cdot \Delta I + \Delta I)}{(\alpha-1)\varepsilon_2}$ 
4 foreach  $C_a$  in  $\mathcal{C}_\mathcal{P}$  do
5    $M_a = (V_{M_a}, E_{M_a} = \emptyset)$  with  $V_{M_a} = \{A_1, \dots, A_d\}$ 
6    $A_a \leftarrow A_G(C_a)$ 
7    $\eta^* = \text{BGTM}(\eta, \sigma, [0, \log N])$ 
8   foreach each attribute node pair  $(A_k, A_l)$  do
9      $I = \text{MI}(A_a(\cdot)_k, A_a(\cdot)_l)$ ,  $I^* = \text{BGTM}(I, \sigma, [0, \log N])$ 
10    if  $I^* \geq \eta^*$  then
11      Add edge  $(A_k, A_l)$  to  $E_{M_a}$ 
12 return  $M = \cup_{C_a \in \mathcal{C}_\mathcal{P}} \{M_a\}$ 

```

Table 2 presents the global sensitivity (GS) of major correlation metrics in existing tabular data publishing methods [47, 50] and their total correlation noise variance (TCNV) under (α, ε) -DAE RDP, where d denotes attribute dimensionality. The total noise variance of BGTM is $\frac{d(d-1) \cdot 4\alpha(\log N \cdot \Delta_{MI} + \Delta_{MI})}{2(\alpha-1)\varepsilon}$. Setting $\varepsilon = 1$, $\alpha = 2$, $\log N = 1$, and $n = 300$ (typical for binary attributes in attributed graph datasets with community size around 300), we compute theoretical noise variance in Table 3. As dimensionality increases, BGTM achieves progressively lower total noise variance.

5.2 Optimized Inference Structure

Dependency graphs form the core structure of graphical models, providing the foundation for attribute marginal reasoning via graphical models [27]. Standard differential privacy mechanisms for graphical models face a fundamental challenge: how do we partition dependency graphs to minimize noise accumulation while preserving essential correlations? We propose the optimized inference structure (OIS) transforms the inference structure selection from a purely structural

Table 2. Analysis of privacy-preserving correlation metrics

Metric	MI	SUC	InDif
GS	$\frac{2 \log(\frac{n+1}{2}) + (n+1) \log(\frac{n+1}{n-1})}{n}$	$\frac{2 + \frac{1}{\ln 2} + 2 \log n}{n}$	$\frac{4}{n}$
TCNV	$\frac{\alpha d^2 (d-1)^2 GS_{MI}^2}{2\epsilon}$	$\frac{\alpha d^2 (d-1)^2 GS_{SUC}^2}{2\epsilon}$	$\frac{16\alpha d^2 (d-1)^2}{2\epsilon n^2}$

Table 3. Theoretical noise variance of different metrics

	d=2	d=5	d=10	d=15	d=20
MI	0.006	0.6698	13.5651	73.8574	241.8282
SUC	0.008	0.8413	17.0369	92.7569	303.7210
InDif	0.001	0.0711	1.4400	7.8400	25.6711
BGTM	0.031	0.3126	1.4068	3.2826	5.9400

problem into an optimization problem with clear objectives. We analyze how noise propagates through exponential potential functions, and formulate subgraph partitioning as minimizing total noise power across all inference operations. The key insight is that different partitioning strategies dramatically affect cumulative noise, allowing us to find configurations that preserve correlations while achieving superior privacy-utility trade-offs. We assume the dependency graph is connected for simplicity; otherwise, each connected component is processed separately.

Potential Function. The potential function $\phi_{ij}(A_G(\cdot)_i, A_G(\cdot)_j; \theta_{ij}) = \exp(P(A_G(\cdot)_i, A_G(\cdot)_j) + \theta_{ij})$ quantifies dependencies between adjacent variables as an unnormalized 2-way marginal distribution, where θ_{ij} is the adjustment parameter.

Graphical Model. A graphical model $G_M = (V_M, E_M, \theta_M)$ consists of nodes $v_i \in V_M$ corresponding to attribute variables $A_G(\cdot)_i$, edges E_M representing correlations, and parameters θ_M ensuring the potential $\exp(P(A_G(\cdot)_i, A_G(\cdot)_j) + \theta_{ij})$ aligns with $P(A_G(\cdot)_i, A_G(\cdot)_j)$ after normalization. The joint probability marginal $P(A_G)$ is approximated through potential aggregation:

$$P(A_G) = \frac{1}{Z} \prod_{(i,j) \in E_M} \phi_{ij}(A_G(\cdot)_i, A_G(\cdot)_j; \theta_{ij}),$$

where Z is the partition function [27, 47].

To achieve differential privacy, we introduce Gaussian noise to 2-way marginal probabilities. The noisy 2-way marginal probability is:

$$\tilde{P}(A_G(\cdot)_i, A_G(\cdot)_j) = P(A_G(\cdot)_i, A_G(\cdot)_j) + \epsilon_{ij}, \quad (8)$$

where $\epsilon_{ij} \sim \mathcal{N}(0, |A_G(\cdot)_i| \cdot |A_G(\cdot)_j| \cdot \sigma^2)$, yielding the modified potential function:

$$\exp(P(A_G(\cdot)_i, A_G(\cdot)_j) + \epsilon_{ij} + \tilde{\theta}_{ij}). \quad (9)$$

Since $\tilde{\theta}_{ij}$ and $P(A_G(\cdot)_i, A_G(\cdot)_j)$ do not independently introduce noise disturbances, and ϵ_{ij} is the primary noise source, we focus on $\exp(\epsilon_{ij})$ to analyze noise impact. As noise operates within the exponential term, propagation transforms from additive to multiplicative space, dramatically amplifying noise characteristics. We measure this using total noise power:

$$E[e^{2x}] = e^{2|A_G(\cdot)_i| \cdot |A_G(\cdot)_j| \sigma^2}. \quad (10)$$

Since direct inference is computationally intractable [7], we decompose G_M into subgraphs. Let T_{v_i, v_j} denote the domain size of $P(A_G(\cdot)_i, A_G(\cdot)_j)$. In probabilistic graphical models, probability marginals are derived by aggregating all edge potential functions within the model. This design effectively captures comprehensive correlations and incorporates cross-marginal influences on target marginal, while simultaneously propagating noise power from other edge potentials into the

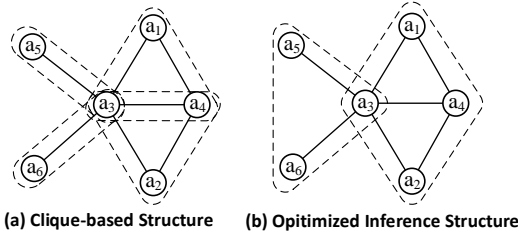


Fig. 3. A dependency graph example

aggregated results. When underlying edge potentials differ during this aggregation process, the resulting target marginal correspondingly vary. Consequently, for each subgraph CL_k , to publish the attributes represented by the nodes within the subgraph, we need to compute $|CL_k|$ univariate or conditional marginals, each with noise power $e^{2\sigma^2 \sum_{(v_i, v_j) \in CL_k} T_{v_i v_j}}$. The total noise power for subgraph CL_k is therefore given by $|CL_k| \cdot e^{2\sigma^2 \sum_{(v_i, v_j) \in CL_k} T_{v_i v_j}}$.

The total noise power of G_M is:

$$\text{TotalPow}(G_M) = \sum_{k=1}^c |CL_k| \cdot e^{2\sigma^2 \sum_{(v_i, v_j) \in CL_k} T_{v_i v_j}}, \quad (11)$$

where $\sigma^2 = \frac{2\alpha|E_M|}{n^2\epsilon_3}$, n is the size of G_M , and $|E_M|$ is the number of noisy marginals.

Current methods typically use maximal cliques for subgraph partitioning, but this approach has fundamental limitations rooted in the exponential nature of noise propagation. Maximal clique structures cause exponential numerical growth in dense graphs due to redundant edges, dramatically elevating total noise power. In sparse graphs, one-edge-one-clique correspondence leads to unreasonable partitioning and repeated inference, introducing additional noise. Therefore, we aim to find subgraph partitions with minimal total noise power, termed **Optimized Inference Structure (OIS)**:

DEFINITION 6. Optimized Inference Structure (OIS) problem. Given a dependency graph G_{M_a} , we want to derive an optimized inference structure configuration $C\mathcal{L} = \{CL_1, \dots, CL_c\}$ such that: (1) The total noise power $\text{TotalPow}(G_M)$ is minimum for all possible values of c ($1 \leq c \leq |E_M|$) (2) $G_M = CL_1 \cup CL_2 \cup \dots \cup CL_c$, (3) $\forall CL_i \neq \emptyset$ and CL_i does not share any common edge with CL_j if $i \neq j$.

We utilize Example 1 illustrated in Figure 3 to demonstrate the differences between maximum clique and OIS approaches.

Example 1. Figure 3 illustrates the subgraph division under maximum clique constraints and the optimized inference structure for attribute set $A = a_1, \dots, a_6$, with domain sizes 2, 2, 5, 5, 2, 2 and $\sigma^2 = 0.005$. Figure (a) shows that the maximum clique division yields four cliques: $A_1A_2A_3$, $A_2A_3A_4$, A_3A_5 , A_3A_6 , with corresponding noise powers $3e^{0.45}$, $3e^{0.45}$, $2e^{0.1}$, and $2e^{0.1}$. In contrast, the optimal inference structure consists of two subgraphs: $A_3A_5A_6$ and $A_1A_2A_3A_4$, incurring noise powers of $3e^{0.2}$ and $4e^{0.65}$, respectively. The optimal structure achieves lower noise ($3e^{0.2} < 4e^{0.1}$ and $4e^{0.65} < 6e^{0.45}$) and better correlation preservation.

While OIS significantly enhances accuracy of estimated joint marginals, the optimization problem is computationally challenging.

THEOREM 3. The OIS problem is NP-hard.

PROOF. We reduce the NP-hard 3-Partition problem [7] to the β -OIS problem (the decision version of OIS). By mapping nodes to 3-Partition elements with corresponding weights, and setting edge

Algorithm 2: Optimized inference structure**Input:** Dependency graph sequence M **Output:** subgraph configuration sequence $\mathcal{C}\mathcal{L}$

```

1 foreach  $C_a$  in  $\mathcal{C}_p$  do
2   for  $d = 1$  to  $|E_{M_a}|$  do
3      $\mathcal{C}\mathcal{L}_{a,d} \leftarrow \text{OISfinding}(M_{a,d})$ 
4     Calculate the total noise power  $TotalPow(G_{M_a})$  based on the division  $\mathcal{C}\mathcal{L}_{a,d}$ 
5      $\mathcal{C}\mathcal{L}_{a,d} = \text{argmin}(TotalPow(G_{M_a}))$ 
6 return  $\mathcal{C}\mathcal{L} = \cup_{C_a \in \mathcal{C}_p} \{\mathcal{C}\mathcal{L}_{a,d}\}$ 

```

weights as the average of connected nodes, β -OIS becomes equivalent to 3-Partition. A full proof is given in [1] Appendix E. \square

Next, we present an approximation algorithm based on the relaxation of mixed-integer programming and the difference of convex algorithm for the OIS problem, which is called **OISfinding**.

We model the domain size of each 2-way marginal as edge weights in a graphical model, transforming the OIS problem into finding minimum edge-weight subgraphs. Consider a graph with n nodes and m edges partitioned into d subgraphs, where c_i denotes the domain size for edge i . The assignment is defined by a binary matrix $Z = [z_{i,k}]_{m \times d}$ with $z_{i,k} = 1$ indicating that edge i belongs to subgraph k , and an incidence matrix $E = [e_{i,p}]_{m \times n}$ where for each edge $i = (u_i, v_i)$, $e_{i,u_i} = e_{i,v_i} = 1$ and all other entries are zero. The node-edge connection count matrix $W = E^T Z$ computes $w_{p,k} = \sum_i e_{i,p} z_{i,k}$. To eliminate duplicate node counting from edge intersections, we apply the element-wise operation $\hat{W} = \min(W, 1)$, yielding deduplicated node counts $F_k(Z) = \sum_p \hat{w}_{p,k}$. With noise energy $G_k(Z) = e^{[\sigma^2(\sum_i z_{i,k} c_i)]}$, the OIS formulation is:

$$\min_Z \sum_{k=1}^d G_k(Z) F_k(Z) \quad \text{s.t.} \quad z_{i,k} \in \{0, 1\}, \quad \sum_k z_{i,k} = 1 \quad (12)$$

Approximating $G_k(Z) \approx 1 + \sigma^2 c^T z_{\cdot,k}$ for small σ and relaxing to $z_{i,k} \in [0, 1]$, we obtain the continuous optimization problem:

$$\min_Z \sum_{k=1}^d [1 + \sigma^2 c^T z_{\cdot,k}] F_k(Z) \quad \text{s.t.} \quad z_{i,k} \in [0, 1], \quad \sum_k z_{i,k} = 1$$

The non-convex nature of $F_k(Z)$ is addressed through difference-of-convex (DC) decomposition using the identity $\min(a, 1) = a - \max(a - 1, 0)$. This allows us to express:

$$F_k(Z) = \sum_p \min((E^T Z)_{p,k}, 1) = A_k(Z) - B_k(Z)$$

where $A_k(Z) = \sum_p (E^T Z)_{p,k}$ is linear in Z , and $B_k(Z) = \sum_p \max((E^T Z)_{p,k} - 1, 0)$ is convex. Substituting this decomposition, the objective becomes:

$$\sum_k [1 + \sigma^2 c^T z_{\cdot,k}] (A_k - B_k) = \underbrace{\sum_k [1 + \sigma^2 c^T z_{\cdot,k}] A_k}_{H_1(Z)} - \underbrace{\sum_k [1 + \sigma^2 c^T z_{\cdot,k}] B_k}_{H_2(Z)}$$

While $H_2(Z)$ is convex, $H_1(Z)$ contains non-convex cross-terms. To convexify $H_1(Z)$, we apply the algebraic identity $uv = \frac{1}{4}(u+v)^2 - \frac{1}{4}(u-v)^2$ to the product $(\sigma^2 c^\top z_{\cdot,k})A_k(Z)$, resulting in:

$$H_1(Z) = \sum_k A_k(Z) + \frac{\sigma^2}{4} \sum_k (c^\top z_{\cdot,k} + A_k(Z))^2 - \frac{\sigma^2}{4} \sum_k (c^\top z_{\cdot,k} - A_k(Z))^2$$

This transformation converts the problematic cross-terms into quadratic forms. We then define:

$$H'_1(Z) = \sum_k A_k(Z) + \frac{\sigma^2}{4} \sum_k (c^\top z_{\cdot,k} + A_k(Z))^2$$

$$H'_2(Z) = H_2(Z) + \frac{\sigma^2}{4} \sum_k (c^\top z_{\cdot,k} - A_k(Z))^2$$

yielding the final DC program:

$$\min_Z [H'_1(Z) - H'_2(Z)] \quad \text{s.t.} \quad z_{i,k} \in [0, 1], \quad \sum_k z_{i,k} = 1 \quad (13)$$

where both H'_1 and H'_2 are convex functions of Z .

This DC structure enables efficient optimization via the Difference of Convex Algorithm (DCA). Subgradients for $H'_2(Z)$ are computable: for $B_k(Z) = \sum_{p=1}^n \max(\beta_{p,k}(Z) - 1, 0)$ where $\beta_{p,k}(Z) = (E^\top Z)_{p,k}$, the subgradient $\partial B_k(Z)/\partial z_{i,k}$ involves the active set $\{p : \beta_{p,k}(Z) \geq 1\}$. The term $\frac{\sigma^2}{4} \sum_{k=1}^d (c^\top z_{\cdot,k} - A_k(Z))^2$ is differentiable with gradient $\frac{\sigma^2}{2} (c^\top z_{\cdot,k} - A_k(Z)) (c_i - \sum_{p=1}^n e_{i,p})$ for each $z_{i,k}$. At each iteration, a subgradient matrix $\nabla H'_2(Z^{(t)})$ can be computed from the current solution $Z^{(t)}$. The optimization yields a continuous probability matrix $Z^* \in [0, 1]^{m \times d}$, where each entry $z_{i,k}^*$ represents the probability that edge i is assigned to cluster k .

Algorithm 2 finds the minimum noise power subgraph configuration for each community. For community C_a , we determine subgraph configuration $CL_{a,d}$ for different initial subgraph numbers d using approximation method **OISfinding**, then calculate total noise power $TotalPow(G_{(M_a)_d})$ based on configuration $CL_{a,d}$ (lines 2–4). We select the configuration with minimum noise variance as the inferred structure for community C_a (line 5) and return $CL = \cup_{\{C_a\}} CL_a$ (line 6).

5.3 Noised Attribute Generation

Directly sampling from the joint distribution is computationally infeasible. Thus, we introduce an efficient local sampling method. Each community is allocated a privacy budget ϵ_3 , based on the marginal distributions of the configuration CL_a . We apply the Gaussian mechanism to add noise to the marginals. The graphical model training framework [27] is used for training. When distributions are inferred, conditional sampling is performed iteratively, using previously sampled attributes to generate more useful ones until all attributes are sampled. Please refer to Appendix B [1] for more details.

6 Graph Reconstruction

6.1 Parameterized graph model

A fundamental challenge in privacy-preserving graph synthesis lies in capturing the complex interplay between structural patterns and node attributes that characterize real-world networks. Existing approaches often struggle to preserve clustering characteristics and triangle formations—critical structural properties that define meaningful graph topology. To address this challenge, PrivAGS introduces the MCMC-Based Cohesive Edge Generation (MCEG) model, which leverages the insight that meaningful graph structures emerge naturally from probabilistic processes that respect both node similarity and local clustering dynamics.

The core design philosophy behind MCEG rests on three key observations: (1) edge formation in real networks is fundamentally driven by node cohesiveness—nodes with similar attributes or structural positions are more likely to connect; (2) triangle formation patterns can be naturally preserved through Markov chain processes that model cohesion state transitions; and (3) discrete probability distributions over cohesiveness levels enable efficient sampling while maintaining structural fidelity. Rather than relying on post-hoc triangle adjustment mechanisms used in prior work, MCEG models edge generation as a cohesion-aware Markov process. This approach naturally reconstructs clustering patterns by generating edges through transitions between cohesiveness states.

Cohesiveness. Given two nodes $u, v \in V_G$, the likelihood of an edge between them increases with their cohesiveness [13, 48]. We use the Jaccard coefficient to quantify structural cohesiveness based on common neighbors: $S(u, v) = \frac{|N(u) \cap N(v)|}{|N(u) \cup N(v)|}$. For attribute-based cohesiveness, we discretize attributes and apply the normalized Manhattan distance: $A(u, v) = 1 - \frac{\sum_{i=1}^d |Z(A_G(u)_i) - Z(A_G(v)_i)|}{d}$, where $Z(\cdot)$ normalizes $A(\cdot)$ to $[0, 1]$, removing dimensional influence. We quantify this cohesiveness through two complementary measures that capture different aspects of edge affinity. So that the composite cohesiveness is as $f(u, v) = A(u, v) + S(u, v)$, where higher values denote a stronger connection between nodes.

To bridge continuous cohesiveness values with discrete probability modeling, MCEG employs a bucketing mechanism that transforms the cohesiveness spectrum into manageable probability distributions. The edge generation probability $\Pi(f(u, v)|n_b, E_G, A_G)$ discretizes continuous cohesiveness into n_b buckets, preserving ordinal relationships while enabling efficient probability learning. This bucket-level calibration ensures that synthetic graphs maintain consistency with the original graph's neighborhood patterns and attribute characteristics.

The key innovation in MCEG lies in modeling edge generation as a Markov chain of cohesion state transitions, naturally preserving clustering characteristics without explicit triangle adjustment. Traditional approaches enforce triangles through post-hoc adjustment, choosing a node k from the neighbors of v and directly adding the edge (u, k) if it does not already exist, the open triad illustrated in Figure 4 (b) is transformed into the triangle depicted in Figure 4 (a). This operation increases the number of triangles in the graph and reduces the discrepancy in clustering characteristics between the synthetic graph and the original graph.

The post-hoc adjustment of triangles naturally induces transitions between cohesiveness states of constituent edges, creating Markov transition patterns that mirror triangle formation without requiring explicit node identification. For a triangle with edges (u, v) , (v, k) , and (k, u) , the probability decomposes as: $Pr[E_{uv} = 1, E_{vk} = 1, E_{ku} = 1] = Pr[E_{uv} = 1] \times Pr[E_{vk} = 1|E_{uv} = 1] \times Pr[E_{ku} = 1|E_{vk} = 1]$.

The conditional term $Pr[E_{vk} = 1|E_{uv} = 1]$ represents the open triplet probability, encodable through bucketing distributions $\Pi(f(u, v))$ and $\Pi(f(v, k))$. This establishes Markov transitions between cohesion states, enabling natural triangle reconstruction. To ensure convergence to the target distribution, MCEG satisfies the detailed balance condition:

$$\Pi(f(u, v))P((u, v) \rightarrow (u', v')) = \Pi(f(u', v'))P((u', v') \rightarrow (u, v)),$$

where $P((u, v) \rightarrow (u', v')) = D((u', v')|(u, v))\alpha((u, v) \rightarrow (u', v'))$ and $P((u', v') \rightarrow (u, v)) = D((u, v)|(u', v'))\alpha((u', v') \rightarrow (u, v))$. Since edges are sampled independently with $D((u, v)|(u', v')) = D(u)D(v)$ and guided by the composite cohesiveness principle (stronger cohesion implies higher connection likelihood), we derive the acceptance probability:

$$\alpha((u, v) \rightarrow (u', v')) = \begin{cases} 1, & \text{if } f(u', v') \geq f(u, v), \\ \frac{\Pi(f(u', v'))D((u, v))}{\Pi(f(u, v))D((u', v'))}, & \text{if } f(v, k) < f(u, v). \end{cases} \quad (14)$$

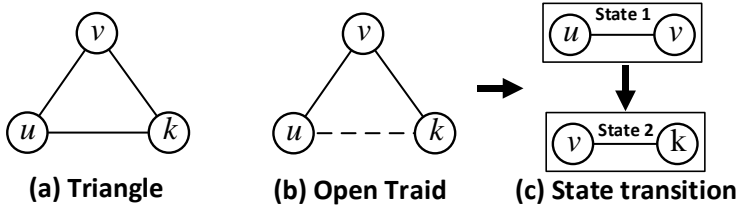


Fig. 4. Triangle formation process

Algorithm 3: MCMC-Based Cohesive Edge Generating

Input: Degree distribution D , Cohesiveness distribution Π , The number of edges m , Neighbor relationship N_G , Node attribute A_G

Output: Synthetic Graph $G' = (V_{G'}, E_{G'}, A_{G'})$

```

1 Initialize  $E_{G'} = \emptyset$ , and randomly sample nodes  $u, v$  from  $D$ ;
2 while  $|E_{G'}| < m$  do
3   Randomly sample node pair  $u'$  and  $v'$  from  $D$ ;
4   Compute acceptance probability  $\alpha((u, v) \rightarrow (u', v'))$  by  $N_G$  and  $A_G$ , draw a random
   number  $r$  from  $[0, 1]$ ;
5   if  $r \leq \alpha$  then
6     update  $u = u', v = v'$ , add edge  $(u', v')$  to  $E_{G'}$ ;
7 return  $E_{G'}$ ;

```

THEOREM 4. *The stationary distribution of our Markov chain converges to the target distribution Π under the acceptance probability in Eq. 14.*

Theorem 4 ensures that our sampling process converges correctly to the target distribution, providing theoretical guarantees for the algorithm's correctness. A complete proof can be found in [1] Appendix F.

Algorithm 3 presents the pseudo-code for the MCMC-Based Cohesive Edge Generating (MCEG) method. The algorithm uses a Markov Chain Monte Carlo (MCMC) approach, where candidate edges are proposed based on the node degree mappings D and accepted with probability $\alpha((u', v') \rightarrow (u, v))$. It iteratively samples and updates node connections until the synthetic graph reaches the desired edge count (lines 2-7).

Convergence Analysis. For practical implementation, MCEG requires the underlying Markov chain to be irreducible and aperiodic. We ensure these properties through minimal adjustment—adding 1 to each node's degree count and cohesiveness bucket count, guaranteeing convergence with negligible sampling impact.

LEMMA 1. *The MCEG algorithm is irreducible.*

PROOF. Since the cohesiveness distribution Π contains no zero values, the acceptance probability between any two edge states exceeds zero, ensuring reachability within finite steps. \square

LEMMA 2. *The MCEG algorithm is aperiodic.*

PROOF. With positive values in the degree mapping and independent sampling processes, node selection is unrestricted. Consequently, every edge maintains positive sampling probability, ensuring aperiodicity. \square

Lemmas 1 and 2 establish that MCEG is both irreducible and aperiodic, guaranteeing convergence to the target distribution and confirming the algorithm's theoretical soundness.

Table 4. Statistics of datasets

Datasets	#Nodes	#Edges	# d_{max}	# d_{avg}	#Domain
Email	1,005	16,063	345	31.86	2^{10}
Facebook	4,039	88,234	141	35.72	2^{10}
Pages	22,470	170,823	709	15.20	2^{20}
Pokec	506,767	3,050,451	3,672	12.03	2^{20}
IMDB	1,198,175	12,269,644	4,136	20.48	2^{13}

Table 5. Parameters table

Parameter	Range
the privacy budget ϵ	1,2,3,4,5
the order of Rényi divergence α	2,3,4,5,6
the bucket of cohesiveness distribution n_b	50,250,500,750,1000

6.2 Edge Reconstruction

The practical deployment of MCEG requires extracting model parameters at the community level while preserving privacy. We achieve this by generating seed graphs through noisy node degree mappings with privacy budget ϵ_4 , enabling structure synthesis while maintaining differential privacy guarantees. Please refer to [1] Appendix C for detailed implementation.

7 EXPERIMENTAL EVALUATION

We first provide a detailed experimental setup. Then, we demonstrate the effectiveness of PrivAGS by comparing it with state-of-the-art methods, and we evaluate the efficiency of PrivAGS on datasets of varying sizes. Next, we conduct a parameter sensitivity analysis and noise propagation analysis of PrivAGS. Due to space limitations, we present the ablation studies and privacy budget allocation experiments in Appendices G and H [1], respectively.

7.1 Experimental Setup

Datasets. We use five real-world datasets: (1) Email [21], an email network where departments serve as node attributes and recipient departments as additional attributes; (2) Facebook [22], a social network with anonymized node attributes (top 10 dimensions as in [8]); (3) Pages [38], a webpage network where edges indicate mutual likes and page categories/purposes are as node attributes; (4) Pokec [42], a social network with node attributes including age, interest, gender, and languages spoken; (5) IMDB [44], a collaborative network of film industry professionals, with node attributes like professions (actors, directors, etc.). Table 4 summarizes the statistics of datasets, where $\#d_{max}$ denotes the maximum edge degree, $\#d_{avg}$ the average edge degree, and $\#Domain$ the number of possible attribute values.

Baselines. To the best of our knowledge, only two methods, AGM-DP [19] and CPGM-DP [8], protect both graph structure and node attribute privacy. AGM-DP includes two generative models, AGM-FCL and AGM-Tri, with AGM-Tri refining the synthetic graph via triangle adjustment as a post-processing step. We adapt these methods to the framework of Rényi differential privacy for a fair comparison. The primary method in our paper is *PrivAGS^a*.

Experimental Settings. In our experiments, we study the impact of parameters, as summarized in Table 5. Our method is implemented in python 3.90 and the experiments are conducted on a computer with Intel Xeon 2.1GHz CPU and 128 GB main memory.

Metrics. We evaluate the quality of the synthetic graph across three aspects, with metrics denoted by a tilde (\sim) above the symbols. **Clustering Character.** We measure clustering feature preservation using the relative error of the global clustering coefficient, $RE(C, \tilde{C}) = |C - \tilde{C}|$, where C and \tilde{C} are the global clustering coefficients of the original and synthetic graphs, respectively. To capture the

Table 6. Overall Utility Performance. The Best Values are Shown in Bold.

Dataset	Model	$RE(C, \tilde{C}) \downarrow$	$RE_{Tri} \downarrow$	$RE(E, \tilde{E}) \downarrow$	$KS(D, \tilde{D}) \downarrow$	$HL(D, \tilde{D}) \downarrow$	$NMI(C_{\mathcal{P}}, \tilde{C}_{\mathcal{P}}) \uparrow$	$L_1(P, \tilde{P}) \downarrow$
Email	PrivAGS ^a	0.0739	0.0525	0.0203	0.0611	0.0565	0.4685	0.0852
	AGM-Tri	0.1699	0.2051	0.0058	0.1134	0.0912	0.3736	0.1248
	AGM-FCL	0.5179	0.4603	0.0051	0.0782	0.0808	0.3522	0.1176
	CPGM-DP	0.1703	0.2058	0.0267	0.0928	0.0764	0.2286	0.2984
Facebook	PrivAGS ^a	0.1025	0.0924	0.0228	0.0959	0.0748	0.8527	0.1264
	AGM-Tri	0.2802	0.3913	0.0193	0.2189	0.1387	0.8118	0.2256
	AGM-FCL	0.6043	0.6361	0.0156	0.1536	0.0938	0.7178	0.1784
	CPGM-DP	0.2371	0.3472	0.0496	0.2144	0.1308	0.6996	0.3528
Pages	PrivAGS ^a	0.6606	0.6722	0.0216	0.0632	0.0517	0.8295	0.0706
	AGM-Tri	0.7107	2.4606	0.0171	0.4022	0.2931	0.6856	0.0904
	AGM-FCL	0.9584	0.9708	0.0155	0.1558	0.1296	0.6678	0.0812
	CPGM-DP	0.8105	1.5996	0.8632	0.2891	0.2616	0.4602	0.3328
Pocec	PrivAGS ^a	0.0823	0.0869	0.0288	0.1299	0.1477	0.1624	0.2379
	AGM-Tri	0.0918	0.1003	0.0172	0.1908	0.2082	0.0772	0.3112
	AGM-FCL	0.6093	0.6530	0.0144	0.1508	0.1714	0.0634	0.2888
	CPGM-DP	-	-	-	-	-	-	-
IMDB	PrivAGS ^a	0.0764	0.0782	0.0145	0.0534	0.1358	0.5021	0.2106
	AGM-Tri	0.7465	3.0423	0.0137	0.3558	0.1820	0.3974	0.3708
	AGM-FCL	0.9518	0.9847	0.0129	0.1674	0.1615	0.3213	0.3024
	CPGM-DP	-	-	-	-	-	-	-

interaction between closed and total triplets, we report the combined clustering error as $RE_{Tri} = |RE_{n_{\Delta}} - RE_{n_{\wedge}}|$, where n_{Δ} and n_{\wedge} represent closed and total triplets. **Topology Structure.** To quantify the difference in degree distributions, we use the Kolmogorov-Smirnov (KS) statistic [19], defined as $KS(D, \tilde{D}) = \max_d |F_D(d) - F_{\tilde{D}}(d)|$, where F_D and $F_{\tilde{D}}$ are the cumulative degree distributions of the original and synthetic graphs. Additionally, we report the Hellinger distance $HL(D, \tilde{D}) = \sqrt{\frac{\sum_d (2\sqrt{D(d)} - \sqrt{\tilde{D}(d)})^2}{2}}$, which is more sensitive to tail differences in the degree distributions, and the relative error of edge count $RE(E, \tilde{E})$, reflecting the overall structure. **Node Attribute.** Node attribute validity is assessed alongside structural features using two metrics: (1) *Edge Affinity*, which quantifies attribute-structure homogeneity. We compute the L_1 distance between the edge-based attribute combination distributions of the original and synthetic graphs, extending this to bivariate combinations. The edge affinity difference is given by: $L_1(P, \tilde{P}) = \frac{1}{|A_C|} \sum_{(a_i, a_j) \in A_C} \frac{1}{|E|} |P(a_i, a_j | E) - \tilde{P}(a_i, a_j | E)|$, where A_C is the set of attribute combinations, E is the set of edges, and P and \tilde{P} are the attribute combination distributions for the original and synthetic graphs. (2) *LouvainAA NMI*, which evaluates community structure preservation using Normalized Mutual Information (NMI) [46]. The NMI between two partitions $C_{\mathcal{P}}$ and $\tilde{C}_{\mathcal{P}}$ is denoted as $NMI(C_{\mathcal{P}}, \tilde{C}_{\mathcal{P}})$.

7.2 Overall Result

We evaluate the overall utility of PrivAGS compared with baselines. Table 6 presents the performance comparison between PrivAGS^a and baseline methods across all evaluation metrics, where all results are averaged values computed over all combinations of privacy budgets and Rényi divergence parameters. It should be noted that PrivAGS^a employs a fixed hyperparameter of $n_b = 500$. The arrow behind the indicator represents the ideal trend for that indicator. The results of CPGM-DP on the Pocec and IMDB dataset are omitted due to the $O(n^2)$ time complexity of the CPGM model, where n is the number of nodes.

Results in terms of clustering character. PrivAGS achieves the best clustering performance through its innovative composite cohesiveness mechanism that unifies structural connectivity

and attribute similarity into $f(u, v) = A(u, v) + S(u, v)$. Unlike other methods that treat features independently, PrivAGS incorporates clustering features through this composite cohesion measure, with MCEG's cohesion-aware MCMC naturally preserving clustering characteristics through Markov state transitions rather than explicit triangle adjustments. Leveraging community structure as the fundamental granularity for synthesis, combined with BGTm's correlation preservation within communities, PrivAGS exhibits the smallest $RE(C, \tilde{C})$ and RE_{Tri} values across most datasets. While PrivAGS leverages advanced composite cohesion and graph reconstruction techniques to address the high information loss and limited ability of prior methods in capturing tightly connected features, it still struggle to capture subtle, low-homogeneity topological patterns within the network. Specifically, due to the limitations imposed by privacy budget constraints and the discrete bucket mechanism, coarse-grained community detection and composite cohesion inevitably disrupt delicate fine-grained structural relationships and low-homogeneity topological features. This problem highlight the limitations of current statistical paradigms in modeling complex topological features, underscoring the need for more advanced methods that can comprehensively capture intricate structural patterns in privacy-preserving graph synthesis.

Results in terms of topology structure. Regarding topological structure, PrivAGS shows weaker edge count performance due to distributing its privacy budget across multiple components, causing greater disturbance to global degree distribution. However, it excels in key metrics like the KS statistic and Hellinger distance. PrivAGS enhances topological reconstruction by using communities to extract degree distribution information and recalculating intra- and inter-community distributions. It constructs a seed graph from these distributions to prevent noise injection in composite cohesion distribution. Conversely, AGM-FCL and AGM-Tri reduce global sensitivity from $2n - 2$ to k by truncating node edge counts, yet still introduce substantial noise. Other methods calculate edge probabilities using edge-based attributes while ignoring topological structures, limiting topological expressiveness. AGM-Tri and CPGM-DP's crude triangle adjustments during post-processing impair topological representation, yielding inferior performance versus AGM-FCL.

Results in terms of node attribute. Compared to the baseline method that releases attributes based only on univariate marginal distributions, PrivAGS achieves superior performance (i.e., $NMI(C_{\mathcal{P}}, \tilde{C}_{\mathcal{P}})$ and $L_1(P, \tilde{P})$) by utilizing multidimensional distributions and a local sampling approach to attribute release. This enhanced performance can be attributed to PrivAGS's innovative approach, which introduces a bounded Gaussian threshold mechanism to more effectively capture intricate dependencies among attributes and construct comprehensive dependency graphs. Moreover, the framework leverages an optimized inference structure as the foundational architecture for distribution inference, which significantly mitigates the impact of noise on the distribution.

7.3 Efficiency Evaluation

Figure 5 presents the average response time results for all methods across all privacy budget and Rényi differential privacy divergence combinations, where PrivAGS^a is configured with hyperparameter $n_b = 500$. Response times are presented as multiples of PrivAGS's time on each dataset. PrivAGS is slightly less efficient than AGM-FCL on the Email dataset due to its heuristic-based edge generation model (MCEG), which prioritizes high composite cohesion edges. These edges concentrate in small regions, causing redundant sampling and reduced efficiency. For other datasets, PrivAGS achieves at least 1.25X efficiency improvement. On large-scale datasets, PrivAGS maintains good synthesis efficiency through density-based compression that enables effective supernode merging, reducing time complexity from computing inter-community edge weight noise. Additionally, MCEG allows efficient edge generation even with low acceptance probabilities on large datasets.

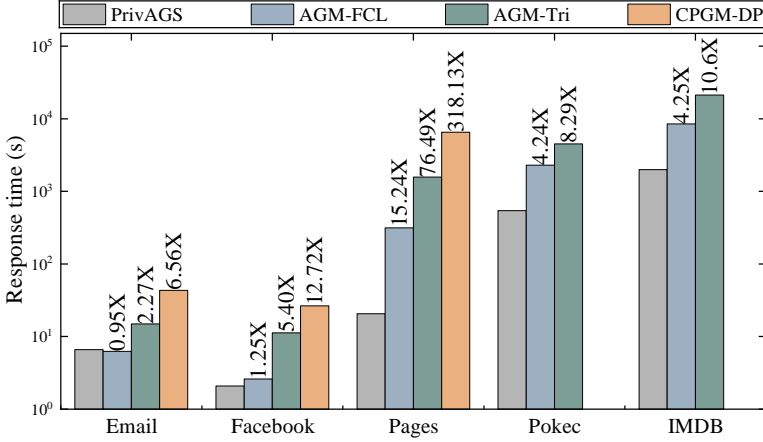


Fig. 5. Efficiency result of different methods

7.4 Parameter Sensitivity Analysis

The parameters employed in Figures 6 and 7 are highlighted in bold in Table 5 (set as the default values). When investigating the sensitivity of any individual parameter, all other parameters are held constant to ensure controlled experimental conditions.

Impact of privacy parameter α . Figures 6(a)-6(f) show the results under different privacy parameters α . As α increases, privacy protection strengthens, injecting more noise, which causes a deterioration in the performance of all methods in terms of three representative metrics.

Impact of privacy budget ϵ . As shown in Figures 6(g)-6(h), with an increased privacy budget, only PrivAGS exhibits a fluctuating downward trend in clustering features. Other methods either fail to control triplet formation due to triangle adjustment or cannot capture clustering features effectively. In contrast, PrivAGS uses composite cohesion to better capture clustering features, improving as the privacy budget increases. For topological structure and node attributes, most methods show improved performance with higher privacy budgets.

Impact of composite cohesiveness distribution bucket number n_b . Figures 7(a)-7(f) show the results by varying the number of bins n_b on Email dataset. As the number of bins n_b increases, errors in the three main metrics decrease, allowing for more precise graph features in synthesis. However, it also dilutes bin probabilities, reducing sampling efficiency. The increase in response time with n_b , as shown in the figure, confirms this trend.

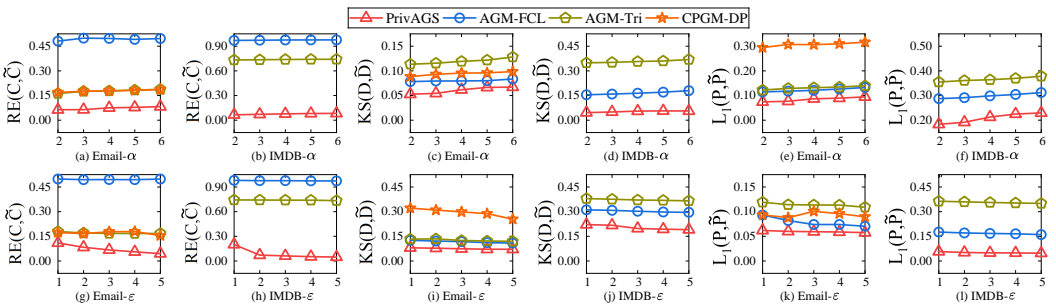


Fig. 6. Parameter sensitivity

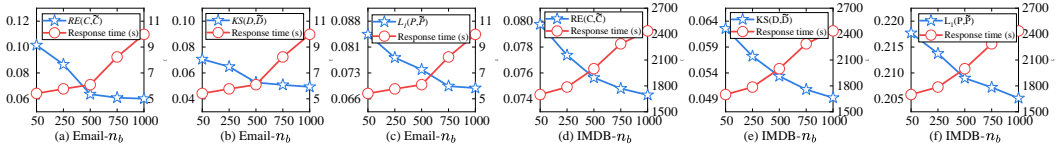


Fig. 7. Parameter sensitivity (Email and IMDB): effectiveness (left Y axis) and efficiency (right Y axis)

Table 7. Analysis of Noise Propagation. The best result is shown in **bold**.

Dataset	Model	$RE(C, \tilde{C}) \downarrow$	$RE_{Tri} \downarrow$	$RE(E, \tilde{E}) \downarrow$	$KS(D, \tilde{D}) \downarrow$	$HL(D, \tilde{D}) \downarrow$	$NMI(C_p, \tilde{C}_p) \uparrow$	$L_1(P, \tilde{P}) \downarrow$
Email	PrivAGS ^a	0.0739	0.0525	0.0203	0.0611	0.0565	0.4685	0.0852
	PrivAGS ⁴	0.0427	0.0413	0	0.0514	0.0419	0.4889	0.0802
	PrivAGS ³	0.0294	0.0337	0	0.0438	0.0376	0.5016	0.0648
	PrivAGS ²	0.0258	0.0244	0	0.0412	0.0367	0.5679	0.0503
	PrivAGS ¹	0.0131	0.0169	0	0.0405	0.0358	0.7150	0.0142
Facebook	PrivAGS ^a	0.1025	0.0924	0.0228	0.0959	0.0748	0.8527	0.1264
	PrivAGS ⁴	0.0993	0.0904	0	0.0890	0.0682	0.8586	0.1172
	PrivAGS ³	0.0984	0.0893	0	0.0877	0.0676	0.8817	0.0989
	PrivAGS ²	0.0886	0.0842	0	0.0847	0.0675	0.8970	0.0899
	PrivAGS ¹	0.0761	0.0803	0	0.0725	0.0614	0.9140	0.0772
Pages	PrivAGS ^a	0.6606	0.6722	0.0216	0.0632	0.0517	0.8295	0.0706
	PrivAGS ⁴	0.5946	0.6197	0	0.0632	0.0447	0.8504	0.0694
	PrivAGS ³	0.5714	0.5837	0	0.0528	0.0416	0.8594	0.0677
	PrivAGS ²	0.5547	0.5669	0	0.0525	0.0410	0.8665	0.0666
	PrivAGS ¹	0.4454	0.4521	0	0.0518	0.0395	0.8708	0.0511
Pokey	PrivAGS ^a	0.0823	0.0869	0.0288	0.1299	0.1477	0.1624	0.2379
	PrivAGS ⁴	0.0801	0.0834	0	0.1201	0.1382	0.1622	0.2305
	PrivAGS ³	0.0776	0.0795	0	0.1025	0.1297	0.1701	0.2261
	PrivAGS ²	0.0762	0.0789	0	0.0989	0.1264	0.1736	0.2236
	PrivAGS ¹	0.0695	0.0713	0	0.0874	0.1156	0.1824	0.2169
IMDB	PrivAGS ^a	0.0764	0.0782	0.0145	0.0534	0.1358	0.5021	0.2106
	PrivAGS ⁴	0.0705	0.0735	0	0.0505	0.1267	0.5254	0.2059
	PrivAGS ³	0.0692	0.0712	0	0.0499	0.1124	0.5326	0.1904
	PrivAGS ²	0.0676	0.701	0	0.0493	0.1101	0.5417	0.1886
	PrivAGS ¹	0.0612	0.0654	0	0.0476	0.1029	0.5598	0.1729

7.5 Noise Propagation Analysis

We analyze noise propagation through the algorithmic pipeline by comparing the original method PrivAGS^a with its variants PrivAGS¹, PrivAGS², PrivAGS³, and PrivAGS⁴. We employ a data substitution strategy wherein noisy inputs at each stage are sequentially replaced with ground truth data following the pipeline order: community detection (Stage 1) → attribute association (Stage 2) → attribute synthesis (Stage 3) → graph reconstruction (Stage 4). The superscript notation indicates the starting stage from which data substitution occurs; for instance, PrivAGS² denotes that the synthetic graph data utilized in Stages 2, 3, and 4 are replaced with the ground truth data.

Table 7 presents the average results for the respective methods across all privacy budget and Rényi divergence combinations when n_b is fixed at 500. As observed, there exists a consistent degradation in terms of all metrics as the proportion of the ground truth data decreases. Specifically, PrivAGS⁴ exhibits improvements over PrivAGS^a in node attributes, with moderate enhancements in clustering characteristics and topological structures. This stems from the fact that PrivAGS⁴ primarily replaces real data pertaining to node degrees and community edge quantities. Under the PrivAGS³, we observe substantial improvements across all metrics compared to PrivAGS⁴,

as PrivAGS³ directly substitutes noisy attribute distributions with true attribute distributions—high-quality attributes provide superior guidance for graph generation. PrivAGS² demonstrates significant improvements in node attributes relative to PrivAGS³, while showing little enhancements in other aspects, since PrivAGS² incorporates authentic attribute correlation data as input, thereby elevating attribute generation quality and producing higher-fidelity synthetic graphs. PrivAGS¹ further advances upon PrivAGS² by incorporating genuine community partitions as input. Since communities serve as the fundamental granularity for both attribute and structure generation in our framework, superior community partitioning can effectively guide the synthesis of both structural and attribute components, creating cascading quality improvements throughout the pipeline.

Furthermore, the aforementioned table reveals that PrivAGS¹ achieves the most substantial improvement over PrivAGS², followed by PrivAGS³'s enhancement over PrivAGS⁴, while PrivAGS⁴'s improvement over PrivAGS^a remains minimal. In conclusion, the Community Division and Attributes Synthesis stages propagate the most significant noise among the four stages, whereas the Graph Reconstruction stage introduces the least noise propagation.

8 Conclusion

We propose PrivAGS, a novel attributed graph synthesis framework based on Rényi differential privacy. The framework employs the bounded Gaussian threshold mechanism and an optimal inference structure with minimal noise power to capture attribute correlations and release node attributes in high-dimensional spaces, mitigating the utility degradation caused by noise in high-dimensional spaces. Furthermore, PrivAGS leverages composite cohesiveness to capture edge homophily and tightly connected features in attributed graphs, upon which it utilizes a cohesion-aware MCMC edge generation method that naturally preserves clustering characteristics and generates graphs efficiently through Markovian state transitions. Experimental results demonstrate that PrivAGS can efficiently synthesize attributed graphs with high utility.

In future, we plan to consider group privacy or leverage deep generative models to capture more comprehensive and complex graph structures, thereby enabling more effective privacy-preserving attributed graph synthesis.

9 Acknowledgments

This work was supported in part by the NSFC under Grants No. (62472377, 62402431, 62025206, U23A20296). Lu Chen is the corresponding author of the work.

References

- [1] Anonymous. 2025. PrivAGS: Differentially Private Attributed Graph Synthesis. (2025). <https://anonymous.4open.science/r/PrivAGS-E663/PrivAGS.pdf>
- [2] Lars Backstrom, Cynthia Dwork, and Jon Kleinberg. 2007. Wherefore art thou r3579x? anonymized social networks, hidden patterns, and structural steganography. In *WWW*. 181–190.
- [3] Jeremiah Blocki, Avrim Blum, Anupam Datta, and Or Sheffet. 2013. Differentially private data analysis of social networks via restricted sensitivity. In *Proceedings of the 4th Conference on Innovations in Theoretical Computer Science* (Berkeley, California, USA) (*ITCS '13*). Association for Computing Machinery, New York, NY, USA, 87–96. doi:10.1145/2422436.2422449
- [4] Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. 2008. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment* 2008, 10 (2008), P10008.
- [5] Mohammad Bolbolian. 2020. Relationship Between Kendall's tau Correlation and Mutual Information. *Revista Colombiana de Estadística* 43 (06 2020), 3–20.
- [6] Rui Chen, Benjamin C. M. Fung, Philip S. Yu, and Bipin C. Desai. 2014. Correlated network data publication via differential privacy. *VLDB J.* 23, 4 (2014), 653–676.
- [7] Rui Chen, Qian Xiao, Yu Zhang, and Jianliang Xu. 2015. Differentially Private High-Dimensional Data Publication via Sampling-Based Inference. In *SIGKDD*. 129–138.

- [8] Xihui Chen, Sjouke Mauw, and Yuniior Ramírez-Cruz. 2020. Publishing Community-Preserving Attributed Social Graphs with a Differential Privacy Guarantee. *Proc. Priv. Enhancing Technol.* 2020, 4 (2020), 131–152.
- [9] Fan Chung and Linyuan Lu. 2002. The average distances in random graphs with given expected degrees. *Proceedings of the National Academy of Sciences* 99, 25 (2002), 15879–15882.
- [10] William Croft, Jörg-Rüdiger Sack, and Wei Shi. 2022. Differential Privacy via a Truncated and Normalized Laplace Mechanism. *Journal of Computer Science and Technology* 37, 2 (2022), 369–388. doi:10.1007/s11390-020-0193-z
- [11] Cynthia Dwork. 2006. Differential privacy. In *ICALP*. Springer-Verlag, Berlin, Heidelberg, 1–12.
- [12] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. 2006. Calibrating noise to sensitivity in private data analysis. In *TCC*. 265–284.
- [13] Aditya Grover and Jure Leskovec. 2016. node2vec: Scalable Feature Learning for Networks. In *SIGKDD*. 855–864.
- [14] Michael Hay, Chao Li, Jerome Miklau, and David D. Jensen. 2009. Accurate Estimation of the Degree Distribution of Private Networks. In *ICDM*. 169–178.
- [15] Ihab F. Ilyas, Volker Markl, Peter J. Haas, Paul Brown, and Ashraf Aboulnaga. 2004. CORDS: Automatic Discovery of Correlations and Soft Functional Dependencies. In *SIGMOD*. 647–658.
- [16] Tianxi Ji, Changqing Luo, Yifan Guo, Qianlong Wang, Lixing Yu, and Pan Li. 2020. Community Detection in Online Social Networks: A Differentially Private and Parsimonious Approach. *IEEE Transactions on Computational Social Systems* 7, 1 (2020), 151–163.
- [17] Xun Jian, Yue Wang, and Lei Chen. 2023. Publishing Graphs Under Node Differential Privacy. *IEEE Transactions on Knowledge and Data Engineering* 35, 4 (2023), 4164–4177.
- [18] Noah Johnson, Joseph P. Near, and Dawn Song. 2018. Towards practical differential privacy for SQL queries. 11, 5 (2018), 526–539.
- [19] Zach Jorgensen, Ting Yu, and Graham Cormode. 2016. Publishing Attributed Social Graphs with Formal Privacy Guarantees. In *SIGMOD*. 107–122.
- [20] Jure Leskovec, Daniel Huttenlocher, and Jon Kleinberg. 2010. Signed networks in social media. In *SIGCHI*. 1361–1370.
- [21] Jure Leskovec, Jon Kleinberg, and Christos Faloutsos. 2007. Graph evolution: Densification and shrinking diameters. *ACM Trans. Knowl. Discov. Data* 1, 1 (March 2007), 2–es. doi:10.1145/1217299.1217301
- [22] Jure Leskovec and Julian McAuley. 2012. Learning to Discover Social Circles in Ego Networks. In *Advances in Neural Information Processing Systems*, Vol. 25.
- [23] Fang Liu. 2019. Generalized Gaussian Mechanism for Differential Privacy. *IEEE Transactions on Knowledge and Data Engineering* 31, 4 (2019), 747–756.
- [24] Kun Liu and Evimaria Terzi. 2008. Towards identity anonymization on graphs. In *SIGMOD*. 93–106.
- [25] Michele Loi and Markus Christen. 2020. Two concepts of group privacy. *Philosophy & Technology* 33, 2 (2020), 207–224.
- [26] Wentian Lu and Jerome Miklau. 2014. Exponential random graph estimation under differential privacy. In *SIGKDD*. 921–930.
- [27] Ryan Mckenna, Daniel Sheldon, and Jerome Miklau. 2019. Graphical-model based estimation and inference for differential privacy. In *PMLR*, Vol. 97. 4435–4444.
- [28] Ilya Mironov. 2017. Rényi Differential Privacy. In *CSF*. 263–275.
- [29] Peter J Mucha and Mason A Porter. 2010. Communities in multislice voting networks. *Chaos: An Interdisciplinary Journal of Nonlinear Science* 20, 4 (2010).
- [30] Arvind Narayanan and Vitaly Shmatikov. 2009. De-anonymizing social networks. In *2009 30th IEEE symposium on security and privacy*. 173–187.
- [31] Hiep H. Nguyen, Abdessamad Imine, and Michaël Rusinowitch. 2015. Differentially Private Publication of Social Graphs at Linear Cost. In *ASONAM*. 596–599.
- [32] Hiep H. Nguyen, Abdessamad Imine, and Michaël Rusinowitch. 2016. Detecting Communities under Differential Privacy. In *WPES*. 83–93.
- [33] Joseph J. Pfeiffer, Timothy La Fond, Sebastian Moreno, and Jennifer Neville. 2012. Fast Generation of Large Scale Social Networks While Incorporating Transitive Closures. In *2012 International Conference on Privacy, Security, Risk and Trust and 2012 International Conference on Social Computing*. 154–165. doi:10.1109/SocialCom-PASSAT.2012.130
- [34] Joseph J. Pfeiffer, Sebastian Moreno, Timothy La Fond, Jennifer Neville, and Brian Gallagher. 2014. Attributed graph models: modeling network structure with correlated attributes. In *WWW*. 831–842.
- [35] Wahbeh Qardaji, Weining Yang, and Ninghui Li. 2014. PriView: practical differentially private release of marginal contingency tables. In *SIGMOD*. 1435–1446.
- [36] Ryan Rogers, Subbu Subramaniam, Sean Peng, David Durfee, Seunghyun Lee, Santosh Kumar Kancha, Shraddha Sahay, and Parvez Ahammad. 2020. LinkedIn’s Audience Engagements API: A Privacy Preserving Data Analytics System at Scale. *CoRR abs/2002.05839* (2020). arXiv:2002.05839 <https://arxiv.org/abs/2002.05839>
- [37] Martin Rosvall and Carl T. Bergstrom. 2008. Maps of random walks on complex networks reveal community structure. *Proceedings of the National Academy of Sciences* 105, 4 (2008), 1118–1123.

- [38] Benedek Rozemberczki, Carl Allen, and Rik Sarkar. 2021. Multi-scale attributed node embedding. *Journal of Complex Networks* 9, 2 (2021), cnab014.
- [39] Thorsten Schmidt. 2007. Coping with copulas. *Copulas - From Theory to Application in Finance* (01 2007).
- [40] C. Seshadhri, Ali Pinar, and Tamara G. Kolda. 2011. An In-depth Study of Stochastic Kronecker Graphs. In *ICDM*. 587–596.
- [41] Lin Song, Peter Langfelder, and Steve Horvath. 2012. Comparison of co-expression measures: Mutual information, correlation, and model based indices. *BMC bioinformatics* 13 (12 2012), 328.
- [42] Lubos Takac and Michal Zabovsky. 2012. Data analysis in public social networks. In *International scientific conference and international workshop present day trends of innovations*, Vol. 1.
- [43] Yue Wang, Xintao Wu, Jun Zhu, and Yang Xiang. 2013. On learning cluster coefficient of private networks. *Social network analysis and mining* 3 (2013), 925–938.
- [44] Yuxiang Wang, Shuzhan Ye, Xiaoliang Xu, Yuxia Geng, Zhenghe Zhao, Xiangyu Ke, and Tianxing Wu. 2024. Scalable Community Search with Accuracy Guarantee on Attributed Graphs. In *2024 IEEE 40th International Conference on Data Engineering (ICDE)*. 2737–2750. doi:10.1109/ICDE60146.2024.00214
- [45] Qian Xiao, Rui Chen, and Kian-Lee Tan. 2014. Differentially private network data release via structural inference. In *SIGKDD*. 911–920.
- [46] Quan Yuan, Zhikun Zhang, Linkang Du, Min Chen, Peng Cheng, and Mingyang Sun. 2023. PrivGraph: differentially private graph data publication by exploiting community information. In *USENIX Security*. Article 182, 18 pages.
- [47] Jun Zhang, Graham Cormode, Cecilia M. Procopiuc, Divesh Srivastava, and Xiaokui Xiao. 2017. PrivBayes: Private Data Release via Bayesian Networks. *ACM Trans. Database Syst.* 42, 4, Article 25 (2017), 41 pages.
- [48] Muhan Zhang and Yixin Chen. 2018. Link prediction based on graph neural networks. In *NIPS*. 5171–5181.
- [49] Sen Zhang, Weiwei Ni, and Nan Fu. 2020. Community Preserved Social Graph Publishing with Node Differential Privacy. In *ICDM*. 1400–1405.
- [50] Zhikun Zhang, Tianhao Wang, Ninghui Li, Jean Honorio, Michael Backes, Shibo He, Jiming Chen, and Yang Zhang. 2021. PrivSyn: Differentially Private Data Synthesis. In *USENIX Security*.
- [51] Bin Zhou and Jian Pei. 2008. Preserving privacy in social networks against neighborhood attacks. In *ICDE*. 506–515.
- [52] Lei Zou, Lei Chen, and M Tamer Özsu. 2009. K-automorphism: A general framework for privacy preserving network publication. *PVLDB* 2, 1 (2009), 946–957.

Received April 2025; revised July 2025; accepted August 2025