GRASP: Differentially Private Graph Reconstruction Defense with Structured Perturbation

Zhiyu Guo[†‡]
Institute of Computing Technology, CAS
University of Chinese Academy of Sciences, CAS
Beijing, China
guozhiyu22@mails.ucas.ac.cn

Yang Liu[*†‡]
Institute of Computing Technology, CAS
Beijing, China
liuyang2023@ict.ac.cn

Xiang Ao[*†‡]
Institute of Computing Technology, CAS
University of Chinese Academy of Sciences, CAS
Beijing, China
aoxiang@ict.ac.cn

Qing He[†‡]
Institute of Computing Technology, CAS
University of Chinese Academy of Sciences, CAS
Beijing, China
heqing@ict.ac.cn

Abstract

In this paper, we reveal that existing Differentially Private Graph Neural Networks (DP-GNNs) are not effective against Graph Reconstruction Attack (GRA). We further attribute the ineffectiveness of existing DP-GNNs against GRA to their unstructured perturbation mechanism, which only induces unidirectional shift in the embedding similarity distribution. Specifically, this perturbation mechanism tends to decrease the embedding similarity of all node pairs without significantly disrupting the relative ranking, thus allowing GRA to still reconstruct the original graph structure by leveraging the relative ranking of similarities. To address this, we propose a novel Differentially Private **Gra**ph Neural Network based on **S**tructured **P**erturbation (GRASP). Specifically, we observe that independent noise tends to decrease the embedding similarity, while identical noise tends to increase it. By integrating these two types of noise using a Bernoulli technique, we introduce a simple yet effective structured perturbation mechanism, which promotes bidirectional shift in the embedding similarity distribution, thereby effectively disrupting the relative ranking and defending against GRA. Extensive experiments on eight benchmark datasets demonstrate that GRASP effectively defends against GRA. Furthermore, GRASP achieves a superior privacy-utility trade-off compared to existing graph structure protection methods. The implementation of GRASP is available at https://github.com/ZhiyuZone/GRASP/.

CCS Concepts

• **Security and privacy → Social network security and privacy**;
• **Mathematics of computing → Graph algorithms**.

[*]Corresponding authors.
[†]Key Lab of Intelligent Information Processing, Institute of Computing Technology, Chinese Academy of Sciences (CAS), Beijing, China. Xiang Ao is also at Institute of Intelligent Computing Technology, CAS, Suzhou, China.
[‡]State Key Lab of AI Safety, Beijing, China.

KDD '25, August 3–7, 2025, Toronto, ON, Canada

ACM ISBN 979-8-4007-1454-2/2025/08
https://doi.org/10.1145/3711896.3736992

Keywords

Graph Neural Networks, Graph Reconstruction Attack, Differential Privacy, Privacy Protection

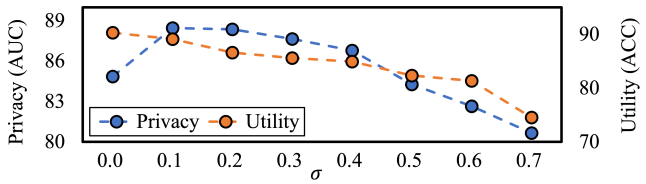
ACM Reference Format:
Zhiyu Guo, Yang Liu, Xiang Ao, and Qing He. 2025. GRASP: Differentially Private Graph Reconstruction Defense with Structured Perturbation. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2 (KDD '25), August 3–7, 2025, Toronto, ON, Canada.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3711896.3736992

1 Introduction

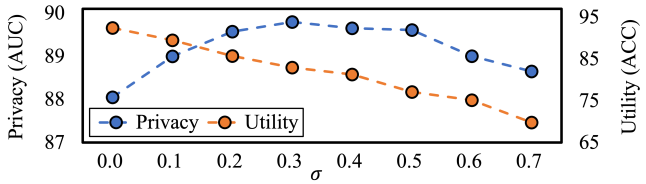
With the wide deployment of Graph Neural Networks (GNNs) [13, 14, 20–22, 34, 40], their security concerns have become increasingly prominent [4, 17–19, 28, 41]. Since GNNs take graph data as input, their output may inadvertently leak sensitive graph structural information. This security risk has been empirically validated through Graph Reconstruction Attacks (GRA) [8, 41–43, 45], which aim to recover the original graph structure from the output of GNNs. GRA can achieve this goal solely by measuring the embedding similarity of node pairs [5, 10, 37].

To protect sensitive graph structure, recent studies have introduced edge-level Differential Privacy (DP) [1, 6] into GNNs and proposed DP-GNNs [3, 30, 36, 38]. As representative DP-GNN methods, GAP [30], DPDGC [3], and PMP [38] adopt an aggregation perturbation mechanism, which adds Gaussian noise at each GNN layer to independently perturb the aggregated node embeddings. Compared to other graph structure protection methods, such as regularization-based techniques [35, 45], the advantage of DP-GNNs lies in their ability to provide protection at both training and inference stages, with formal privacy guarantees and low computational overhead [30]. Therefore, DP-GNNs are widely recognized as advanced privacy protection methods for graph structure.

However, our empirical results surprisingly reveal that current state-of-the-art DP-GNNs [3, 30] are not effective against GRA. Specifically, we integrate the popular aggregation perturbation mechanism into a basic Graph Convolutional Network (GCN) [14] by adding independent Gaussian noise to the aggregated node embeddings at each GCN layer. After training, we perform a similarity-based GRA [5, 10, 37] on the trained GCN, where the probability of each edge in the reconstructed graph structure is derived from



(a) Privacy-utility trade-off on the PubMed dataset.



(b) Privacy-utility trade-off on the Computer dataset.

Figure 1: Privacy-utility trade-off of the aggregation perturbation mechanism adopted by existing DP-GNNs under varying standard deviations σ of Gaussian noise. Privacy is evaluated by the performance of similarity-based GRA with the AUC metric, while utility is evaluated by the performance of node classification tasks with the ACC metric.

the embedding similarity of all node pairs, and the AUC metric is used to evaluate the binary classification performance of the attack. The results are shown in Figure 1. It can be observed that as the Gaussian noise increases, the model's performance in downstream tasks significantly decreases. However, the performance of GRA does not significantly decrease, and even unexpectedly increases in some cases. This empirical study demonstrates that the aggregation perturbation mechanism adopted by existing DP-GNNs fails to achieve an effective privacy-utility trade-off when evaluating the performance of graph structure protection against GRA. This ineffectiveness does not align with the original goal of differential privacy protection for graph structure.

Based on the above observations, we further attribute the ineffectiveness of existing DP-GNNs against GRA to their unstructured perturbation mechanism, which only induces unidirectional shift in the embedding similarity distribution. Specifically, the aggregation perturbation mechanism adopted by existing DP-GNNs injects independent additive Gaussian noise into each node embedding in GNNs. This unstructured perturbation mechanism uniformly injects noise and tends to decrease the embedding similarity of all node pairs without significantly disrupting the relative ranking. Therefore, at low perturbation intensity, similarity-based GRA [37] can still reconstruct the original graph structure by leveraging the relative ranking of embedding similarities. At high perturbation intensity, the embedding similarity of all node pairs is distributed at a low level, which effectively defends against GRA but severely distorts node embeddings, thereby harming downstream task performance. Overall, existing DP-GNNs fail to achieve an effective privacy-utility trade-off in defending against GRA due to the inherent flaw of their unstructured perturbation mechanism.

To address the limitations of existing DP-GNN methods, we propose a novel Differentially Private **Gra**ph Neural Network with **S**tructured **P**erturbation (GRASP). Specifically, we reveal that independent noise tends to decrease the embedding similarity, while identical noise tends to increase it. To fully perturb the embedding similarity distribution by leveraging both types of noise, we propose a structured perturbation implemented by a simple yet effective Bernoulli technique, where each node embedding is randomly perturbed by either independent or identical noise with a predefined probability. This structured perturbation mechanism integrates the advantages of both types of noise, thereby promoting a bidirectional shift in the embedding similarity distribution. Compared to existing DP-GNN methods, our proposed method can more effectively disrupt the relative ranking of embedding similarities, thereby defending against GRA and achieving a better privacy-utility trade-off. It is noteworthy that our proposed GRASP inherits the advantages of existing DP-GNN methods, providing protection during both training and inference with formal privacy guarantees. Moreover, GRASP is a model-agnostic approach that can be flexibly combined with any GNN architecture. Extensive experiments conducted on eight benchmark datasets demonstrate that GRASP effectively defends against GRA. Furthermore, compared with existing graph structure protection methods, GRASP achieves an outstanding privacy-utility trade-off.

In summary, our contributions can be listed as follows:

- We attribute the ineffectiveness of existing DP-GNNs against GRA to their unstructured perturbation, which only induces a unidirectional shift in the embedding similarity.
- We propose a novel DP-GNN based on structured perturbation, which promotes a bidirectional shift in the embedding similarity and effectively defends against GRA.
- Extensive experiments are conducted to validate that our proposed method achieves a superior privacy-utility trade-off in defending against GRA.

2 Preliminaries

2.1 Graph Neural Networks

A graph is denoted as $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$, where the node set $\mathcal{V} = \{v_1, \ldots, v_N\}$ comprises N nodes and the edge set $\mathcal{E} = \{e_1, \ldots, e_E\}$ comprises E edges. Each node has a D-dimensional feature vector, denoted as $\mathbf{X} \in \mathbb{R}^{N \times D}$. The graph structure of $\mathcal{G}$ can be represented as an adjacency matrix $\mathbf{A} \in \{0,1\}^{N \times N}$, where $\mathbf{A}_{ij} = 1$ if $(i, j) \in \mathcal{E}$. Additionally, we denote all node labels as one-hot vectors $\mathbf{Y} \in \{0,1\}^{N \times C}$, where C is the number of classes.

The learning process of GNNs is typically formulated as a message-passing mechanism that iteratively aggregates information from a node's neighbors. For node v_i, the set of its neighbors is denoted as $\mathcal{N}(v_i) = \{v_j | \mathbf{A}_{ij} = 1\}$. In the l-th layer, the node embeddings $\mathbf{H}^{(l)} \in \mathbb{R}^{N \times d}$ are computed by updating the node embeddings from the previous layer $\mathbf{H}^{(l-1)}$ according to the formulation:

$$\mathbf{H}_i^{(l)} = \text{UPDATE}\left(\text{AGG}\left(\{\mathbf{H}_j^{(l-1)} | v_j \in \overline{\mathcal{N}}(v_i)\}\right)\right). \tag{1}$$

Here, $\overline{\mathcal{N}}(v_i) = \mathcal{N}(v_i) \cup \{v_i\}$ denotes node v_i's extended neighborhood. AGG$(\cdot)$ represents the aggregation function, which combines the information from neighboring nodes. The UPDATE$(\cdot)$ function transforms the aggregated information into new node embeddings.

2.2 Graph Reconstruction Attacks

Graph Reconstruction Attacks (GRA) [41, 43, 45] aim to recover the underlying structure of a graph, typically based on prior knowledge and a trained GNN. These attacks can be considered as edge-level membership inference attacks (MIA) [28, 41].

Definition 1 (Graph Reconstruction Attack). *Given a set of prior knowledge $\mathcal{K}$ and a trained GNN f_{θ^*}, Graph Reconstruction Attack seeks to recover the original adjacency matrix $\hat{\mathbf{A}}^*$ of the corresponding input graph $\mathcal{G}$. Specifically, the attack aims to maximize the likelihood of the graph's adjacency matrix as follows:*

$$\hat{\mathbf{A}}^* = \arg\max_{\hat{\mathbf{A}}} P(\hat{\mathbf{A}}|f_{\theta^*}, \mathcal{K}), \tag{2}$$

where $P(\cdot)$ denotes the attack method that generates $\hat{\mathbf{A}}$, and $\mathcal{K}$ represents a subset of the available knowledge, including node features $\mathbf{X}$, labels $\mathbf{Y}$, node embeddings $\mathbf{H}$, or predicted labels $\hat{\mathbf{Y}}$.

One type of GRA relies solely on the similarity between node embeddings [5, 10, 37]. Node pairs with high embedding similarity are more likely to be connected with an edge. Given node embeddings $\mathbf{H}_i$ and $\mathbf{H}_j$, the attack predicts the likelihood of an edge between v_i and v_j based on their similarity:

$$\hat{\mathbf{A}}^*_{ij} = \sigma(\mathrm{Sim}(\mathbf{H}_i, \mathbf{H}_j)), \tag{3}$$

where $\sigma(\cdot)$ is the activation function, and $\mathrm{Sim}(\cdot,\cdot)$ is a similarity function that typically uses cosine or correlation similarity, both of which have been proven to be effective in prior research [5, 10, 37]. This similarity-based GRA is a simple yet effective attack strategy without requiring other prior knowledge. **In this paper, we mainly focus on defending against this type of attack.**

2.3 Differentially Private GNNs

Differential privacy (DP) [1, 6] protects data privacy by ensuring that the output of a computation does not change significantly when any single data point is added or removed from the dataset. To safeguard the privacy of graph data, DP has been integrated into GNNs, leading to the development of DP-GNNs [3, 30, 36, 38]. In particular, the concept of edge-level DP [30] has been introduced specifically to protect the structure of the graph.

Definition 2 (Edge-Level Differential Privacy). *Let $\mathcal{G}$ and $\mathcal{G}'$ be two graphs that are edge-level adjacent, meaning that $\mathcal{G}$ and $\mathcal{G}'$ differ by at most one edge. An algorithm $\mathcal{A}$ is said to satisfy (ϵ, δ)-edge-level differential privacy if, for every pair of $\mathcal{G}$ and $\mathcal{G}'$, and for any set of possible outputs S, the following condition holds:*

$$\Pr[\mathcal{A}(\mathcal{G}) \in S] \leq e^{\epsilon} \Pr[\mathcal{A}(\mathcal{G}') \in S] + \delta. \tag{4}$$

Here, the parameter $\epsilon > 0$ is the privacy budget, with smaller values of ϵ providing stronger privacy guarantees. The parameter $\delta > 0$ represents the probability that the privacy guarantee may fail.

To achieve edge-level DP in GNNs, recent studies have introduced an aggregation perturbation mechanism [3, 30, 38]. The core of this mechanism is to add independent Gaussian noise to each node embedding during the aggregation process of GNNs. Specifically, for each node v_i in the l-th layer, the aggregation perturbation process is formulated as follows:

$$\mathbf{H}_i^{(l)} = \mathrm{UPDATE}\left(\mathrm{AGG}\left(\{\mathbf{H}_j^{(l-1)} | v_j \in \overline{\mathcal{N}}(v_i)\}\right) + \mathbf{n}_i^{(l)}\right). \tag{5}$$

Here, $\mathrm{AGG}(\cdot)$ aggregates information from neighboring nodes and then adds Gaussian noise $\mathbf{n}_i \sim \mathcal{N}(0, \sigma^2 \mathbf{I}_d)$ with zero mean and covariance $\sigma^2 \mathbf{I}_d$, where $\mathbf{I}_d$ is the d-dimensional identity matrix.

3 Structured Perturbation

In this section, we characterize the independent noise used in existing DP-GNNs as unstructured perturbation, meaning that the noise is patternless and lacks any systematic structure. Our analysis reveals that such unstructured perturbation possesses inherent limitations in defending against GRA. In contrast, our proposed structured perturbation, which incorporates specifically designed intrinsic patterns, can effectively defend against GRA.

3.1 Limitations of Existing DP-GNNs

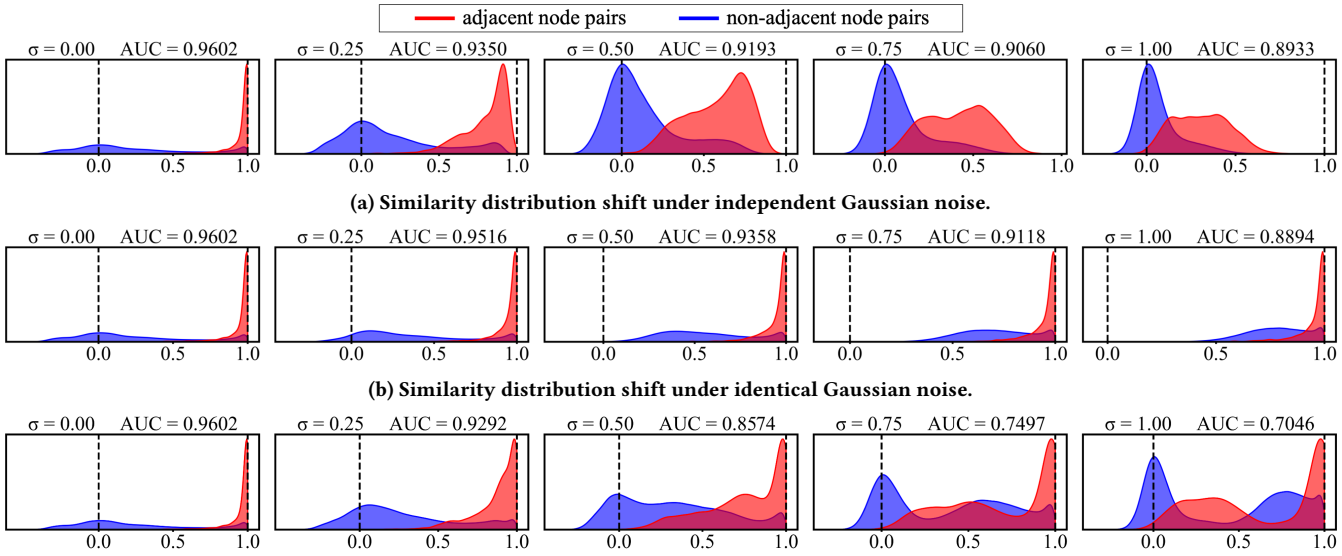
Our experiments shown in Figure 1 indicate that existing DP-GNNs cannot defend against GRA. As a simple yet effective attack strategy, similarity-based GRA reconstructs the original graph structure solely by measuring the embedding similarity of node pairs. Therefore, to further explore the reasons behind the ineffectiveness of existing DP-GNNs against GRA, we conducted experiments to examine the specific impact of independent noise on the distribution of embedding similarity.

Specifically, we trained a basic Graph Convolutional Network (GCN) [14] on the Cora dataset to obtain the trained model f_{θ^*}, which generates node embeddings $\mathbf{H} = f_{\theta^*}(\mathbf{X}, \mathbf{A})$. Then, we added independent Gaussian noise to the embeddings, where the noise for each node embedding is represented by $\mathbf{n_i} \sim \mathcal{N}(0, \sigma^2 \mathbf{I}_d)$, with $\mathbf{I}_d$ being the identity matrix of size d. The noisy embeddings are denoted as $\tilde{\mathbf{H}}_i = \mathbf{H}_i + \mathbf{n}_i$. Afterward, we calculated the embedding similarity of all node pairs using cosine similarity:

$$\mathrm{Sim}(\tilde{\mathbf{H}}_i, \tilde{\mathbf{H}}_j) = \frac{\tilde{\mathbf{H}}_i \cdot \tilde{\mathbf{H}}_j}{\|\tilde{\mathbf{H}}_i\| \|\tilde{\mathbf{H}}_j\|} = \frac{(\mathbf{H}_i + \mathbf{n}_i) \cdot (\mathbf{H}_j + \mathbf{n}_j)}{\|\mathbf{H}_i + \mathbf{n}_i\| \|\mathbf{H}_j + \mathbf{n}_j\|}. \tag{6}$$

We investigated the impact of independent Gaussian noise on node embedding similarity distributions using kernel density estimation (KDE), as shown in Figure 2a. Our analysis yields three key findings:

- **Baseline Performance ($\sigma = 0$):** In the noiseless case, adjacent node pairs exhibit much higher embedding similarity than non-adjacent pairs. This distinction enables GRA to effectively reconstruct the original graph structure based on similarity ranking. The performance of GRA (edge-level binary classification) is measured with the AUC metric.
- **Noise-Induced Similarity Shift:** As the noise intensity (standard deviation σ) increases, we observe a consistent leftward shift in the similarity distributions of all node pairs, converging toward zero. This aligns with the theoretical expectation that the cosine similarity between independent Gaussian noise vectors approaches zero.
- **Differentiated Noise Impact:** Importantly, the similarity distributions of non-adjacent node pairs remain stable under noise, as their initial similarity is already near zero. In contrast, adjacent pairs are more sensitive: their similarity distribution shifts toward that of non-adjacent pairs as σ increases. This asymmetry suggests that independent Gaussian noise primarily disrupts high-similarity pairs while leaving low-similarity pairs relatively unaffected.



(a) Similarity distribution shift under independent Gaussian noise.

(b) Similarity distribution shift under identical Gaussian noise.

(c) Similarity distribution shift under the integration of independent and identical Gaussian noise.

Figure 2: Embedding similarity distribution for adjacent (red curve) and non-adjacent (blue curve) node pairs under three types of Gaussian noise mechanisms with varying standard deviations. The performance of GRA is evaluated with the AUC metric. As unstructured perturbations, independent or identical Gaussian noise can only induce unidirectional shifts in embedding similarity. In contrast, our proposed structured perturbation integrates these two types of noise using a Bernoulli technique, promoting bidirectional shifts in embedding similarity and effectively defending against GRA.

These findings highlight the limitations of the unstructured perturbation mechanism adopted by existing DP-GNNs in defending against GRA, which only induces unidirectional shift in the embedding similarity distribution. At low noise levels, the similarity of adjacent node pairs remains distinguishable from that of non-adjacent node pairs, making similarity-based GRA still effective. While high noise intensities eventually collapse all similarities to zero, this comes at the cost of severe embedding distortion.

3.2 Beyond Unstructured Perturbation

To better perturb low-similarity node pairs, we investigate an identical noise mechanism. Instead of applying independent noise $\mathbf{n}_i$, we inject identical Gaussian noise $\bar{\mathbf{n}} \sim \mathcal{N}(0, \sigma^2 \mathbf{I}_d)$ to all node embeddings, yielding perturbed embeddings $\tilde{\mathbf{H}}_i = \mathbf{H}_i + \bar{\mathbf{n}}$. The cosine similarity between perturbed node embeddings is computed as:

$$\mathrm{Sim}(\tilde{\mathbf{H}}_i, \tilde{\mathbf{H}}_j) = \frac{\tilde{\mathbf{H}}_i \cdot \tilde{\mathbf{H}}_j}{\|\tilde{\mathbf{H}}_i\| \|\tilde{\mathbf{H}}_j\|} = \frac{(\mathbf{H}_i + \bar{\mathbf{n}}) \cdot (\mathbf{H}_j + \bar{\mathbf{n}})}{\|\mathbf{H}_i + \bar{\mathbf{n}}\| \|\mathbf{H}_j + \bar{\mathbf{n}}\|}. \tag{7}$$

Similarly, we visualize the embedding similarity distribution using KDE in Figure 2b. Identical noise shows an inverse effect compared to independent noise, systematically increasing the embedding similarity of all node pairs. Despite introducing only a single noise vector (which seems less disruptive to sensitive information), identical noise still achieves comparable confusion effects as independent noise on embedding similarity between adjacent and non-adjacent node pairs, reflected by the AUC metric measuring GRA performance. This counterintuitive effectiveness of identical noise challenges conventional assumption about the independence of noise in DP frameworks for graph structure protection.

Based on the above analysis, we reveal that independent noise primarily reduces the similarities of high-similarity pairs while having minimal impact on low-similarity pairs. Conversely, identical noise increases pairwise similarities but shows limited impact on already high-similarity pairs. To promote bidirectional shift in the embedding similarity distribution—reducing the similarities of adjacent node pairs and increasing those of non-adjacent pairs—we propose a structured perturbation mechanism that uses the Bernoulli technique to combine both types of noise. For each node embedding $\mathbf{H}_i$, we sample a Bernoulli weight $\lambda_i \sim \mathrm{Bernoulli}(p)$ to control noise selection between identical noise $\bar{\mathbf{n}}$ and independent noise $\mathbf{n}_i$:

$$\tilde{\mathbf{H}}_i = \mathbf{H}_i + \lambda_i \bar{\mathbf{n}} + (1 - \lambda_i)\mathbf{n}_i, \tag{8}$$

We compute pairwise cosine similarities with $p = 0.7$, and visualize distributions via KDE in Figure 2c. Key findings reveal:

- This structured perturbation induces a bidirectional shift in embedding similarity—non-adjacent pairs exhibit similarity increases while adjacent pairs show decreases.
- At equivalent noise magnitudes, this approach achieves superior similarity ranking disruption compared to independent or identical noise mechanisms, effectively obscuring the discriminative boundary between the embedding similarity of adjacent and non-adjacent node pairs.
- Empirical results measured by the AUC metric demonstrate enhanced defense against GRA without injecting high-intensity Gaussian noise, indicating an improved privacy-utility trade-off in defending against GRA.

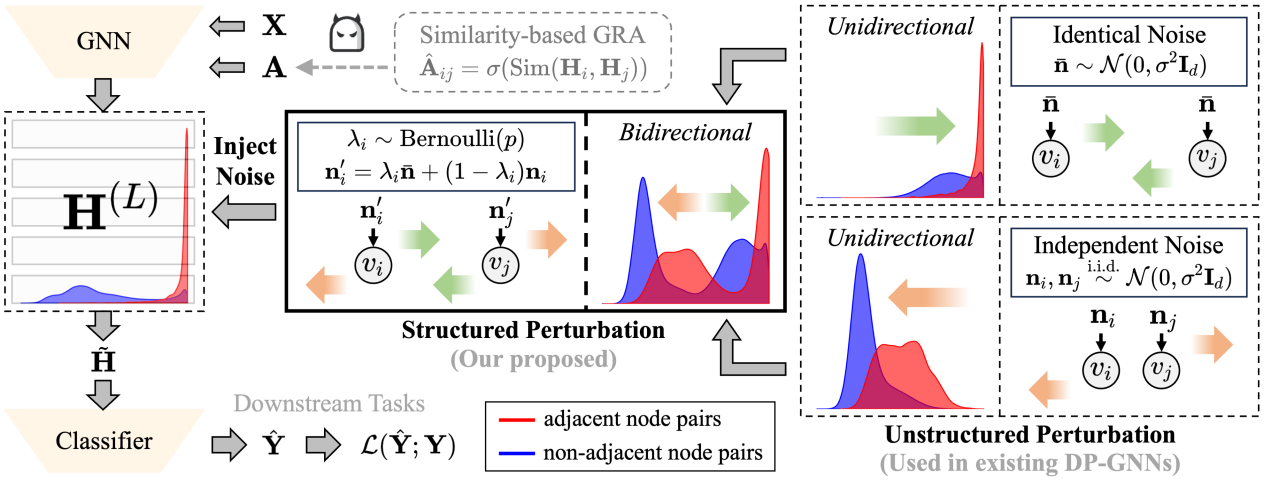


Figure 3: Illustration of our proposed GRASP model. The independent noise $\mathbf{n}_i$ tends to decrease the embedding similarity (push v_i and v_j apart), while identical noise $\bar{\mathbf{n}}$ tends to increase it (pull v_i and v_j closer). Each node embedding $\mathbf{H}_i$ is randomly perturbed by either independent or identical noise with a Bernoulli probability p. This structured perturbation mechanism promotes a bidirectional shift in the embedding similarity distribution, which significantly disrupts the relative ranking and enhances the defense against GRA.

4 The Proposed Defense Method

Overview. Based on the exploratory analysis, the ineffectiveness of existing DP-GNNs against GRA is attributed to their unstructured perturbation mechanisms, which only induce unidirectional shifts in the embedding similarity. To address the limitations of existing DP-GNN methods, we propose a novel Differentially Private GNN with Structured Perturbation (GRASP), which promotes bidirectional shifts in the embedding similarity and effectively defends against GRA, as illustrated in Figure 3. The proposed structured perturbation leverages both identical noises and independent noises and promotes a bidirectional shift in the embedding similarity distribution, which significantly disrupts the relative ranking and enhances the defense effectiveness against GRA.

Model Details. Formally, the graph $\mathcal{G}$ with node features $\mathbf{X}$ and graph structure $\mathbf{A}$ is inputted into a GNN model f_θ (e.g., GCN [14], GAT [34], or GraphSAGE [9]), which iteratively aggregates information from a node's neighbors:

$$\mathbf{H}_i^{(l)} = \mathrm{UPDATE}\Big(\mathrm{AGG}\big(\{\mathbf{H}_j^{(l-1)} \mid v_j \in \overline{\mathcal{N}}(v_i)\}\big)\Big), \tag{9}$$

where $\mathbf{H}^{(0)} = \mathbf{X}$. After L rounds of iterative aggregation, we obtain the final node embeddings $\mathbf{H}^{(L)}$. To defend against GRA, we aim to add Gaussian noise to $\mathbf{H}^{(L)}$. Unlike existing DP-GNNs that add independent Gaussian noise to each node embedding, we introduce a simple yet effective Bernoulli technique, where each node embedding $\mathbf{H}_i^{(L)}$ is randomly perturbed by either independent or identical noise with a predefined probability p. Specifically, we sample one identical Gaussian noise vector $\bar{\mathbf{n}}$ and independent Gaussian noise vectors $\mathbf{n}_i$ from the same Gaussian distribution:

$$\bar{\mathbf{n}} \sim \mathcal{N}(0, \sigma^2 \mathbf{I}_d), \quad \mathbf{n}_i \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, \sigma^2 \mathbf{I}_d), \tag{10}$$

where $\sigma > 0$ is the standard deviation, and $\mathbf{I}_d$ is the d-dimensional identity matrix. All noise vectors are pairwise independent, i.e., $\bar{\mathbf{n}} \perp\!\!\!\perp \mathbf{n}_i$ and $\mathbf{n}_i \perp\!\!\!\perp \mathbf{n}_j$ for $i \neq j$. Then, we introduce a Bernoulli technique by generating a weight λ_i for each node, which integrates the identical Gaussian noise vector $\bar{\mathbf{n}}$ and the independent Gaussian noise vector $\mathbf{n}_i$ with probability p:

$$\lambda_i \sim \mathrm{Bernoulli}(p), \quad \mathbf{n}'_i = \lambda_i \bar{\mathbf{n}} + (1-\lambda_i)\mathbf{n}_i. \tag{11}$$

Afterwards, we add the noise vector $\mathbf{n}'_i$ fused by the Bernoulli technique to each node embedding:

$$\tilde{\mathbf{H}}_i = \mathrm{Norm}(\mathbf{H}_i^{(L)} + \mathbf{n}'_i), \tag{12}$$

where $\mathrm{Norm}(\cdot)$ is LayerNorm [39] or L2-Norm. This structured perturbation mechanism integrates the advantages of both types of noise, thereby promoting bidirectional shift in the embedding similarity distribution and effectively defending against GRA.

This perturbation mechanism is applied during both the training and inference phases, thereby enabling full-phase privacy protection. For the node classification task, the final perturbed node embeddings $\tilde{\mathbf{H}}_i$ are fed into a classifier $f_{\text{classifier}}$, typically a one-layer MLP, which produces the predicted label $\hat{\mathbf{Y}}_i$ for node v_i:

$$\hat{\mathbf{Y}}_i = f_{\text{classifier}}(\tilde{\mathbf{H}}_i). \tag{13}$$

To optimize the model, we typically adopt a cross-entropy loss:

$$\mathcal{L}(\theta) = -\frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{C} \mathbf{Y}_{i,c} \log(\hat{\mathbf{Y}}_{i,c}), \tag{14}$$

where $\mathbf{Y}_{i,c}$ is the true label, and $\hat{\mathbf{Y}}_{i,c}$ is the predicted probability for node v_i in class c. The parameters θ of the model are learned through backpropagation during training.

Discussion. (1) Privacy Guarantee and Computational Efficiency. Our proposed GRASP method inherits the advantages of existing DP-GNN methods by providing privacy protection during both training and inference, which satisfies edge-level DP guarantees [30], ensuring that the model does not leak sensitive structural

Table 1: Comparison of privacy and utility for different perturbation mechanisms under varying Gaussian noise standard deviations. Privacy in defending against GRA is evaluated with the AUC metric, while utility in downstream node classification tasks is assessed with the ACC metric. Lower AUC and higher ACC indicate a better privacy-utility trade-off.

σ	Model	Cora AUC↓ ACC↑	CiteSeer AUC↓ ACC↑	PubMed AUC↓ ACC↑	Computer AUC↓ ACC↑	Photo AUC↓ ACC↑	CS AUC↓ ACC↑	Physics AUC↓ ACC↑	WikiCS AUC↓ ACC↑
0.0	GCN	95.81 / 87.75	98.08 / 77.48	84.86 / 90.29	87.99 / 92.44	87.91 / 95.95	93.58 / 94.55	89.61 / 97.20	92.95 / 79.88
	GAT	92.98 / 88.12	97.31 / 76.05	90.62 / 88.32	89.86 / 94.09	90.30 / 96.41	94.17 / 96.06	90.12 / 97.09	91.77 / 80.75
0.5	GCN+USP	91.77 / 87.09	88.39 / 73.65	83.34 / 86.83	86.19 / 91.09	86.43 / 95.88	85.82 / 92.42	84.38 / 94.96	89.58 / 79.65
	GCN+GRASP	88.85 / 87.66	70.25 / 76.43	64.91 / 88.22	71.15 / 90.64	84.13 / 95.75	70.13 / 93.87	70.43 / 95.75	70.21 / 79.72
	GAT+USP	88.13 / 87.85	92.38 / 75.53	84.80 / 88.20	88.03 / 93.80	88.04 / 96.31	93.00 / 95.95	75.95 / 95.13	88.89 / 80.44
	GAT+GRASP	74.47 / 87.75	78.67 / 75.53	68.66 / 88.17	88.05 / 93.68	87.58 / 96.37	90.35 / 95.97	66.34 / 95.92	69.79 / 80.67
1.0	GCN+USP	89.98 / 86.83	82.27 / 73.20	78.23 / 84.95	86.20 / 90.42	85.75 / 95.71	81.64 / 89.54	82.85 / 92.52	80.56 / 78.40
	GCN+GRASP	74.83 / 87.48	66.22 / 75.75	62.94 / 86.57	70.03 / 90.13	69.38 / 95.69	69.33 / 92.65	68.01 / 95.07	67.37 / 79.59
	GAT+USP	86.52 / 87.29	90.53 / 74.32	80.74 / 87.85	85.82 / 93.40	86.76 / 96.22	90.65 / 95.61	73.89 / 92.90	82.58 / 79.96
	GAT+GRASP	72.42 / 87.64	73.07 / 75.15	66.15 / 87.64	82.99 / 93.56	80.93 / 96.34	71.22 / 95.79	65.77 / 95.08	67.70 / 80.61
1.5	GCN+USP	88.70 / 86.65	77.62 / 71.77	75.82 / 83.81	87.08 / 90.28	86.30 / 95.63	79.75 / 86.71	81.83 / 91.51	77.25 / 77.35
	GCN+GRASP	70.26 / 87.38	64.88 / 74.70	62.15 / 85.79	69.28 / 89.15	68.83 / 95.48	67.97 / 91.89	67.27 / 94.59	64.97 / 79.33
	GAT+USP	86.18 / 86.28	88.29 / 74.17	78.24 / 87.55	85.34 / 93.35	84.37 / 96.14	85.76 / 94.67	71.61 / 90.74	73.79 / 79.73
	GAT+GRASP	70.64 / 87.38	71.12 / 74.92	65.17 / 87.28	78.77 / 93.49	69.24 / 96.26	70.29 / 95.20	64.69 / 94.39	64.85 / 80.15
2.0	GCN+USP	88.55 / 86.56	75.37 / 68.09	72.88 / 82.16	88.00 / 89.68	84.83 / 95.58	77.50 / 83.87	78.40 / 80.38	76.49 / 75.95
	GCN+GRASP	69.21 / 87.29	64.23 / 74.43	61.53 / 85.73	68.32 / 88.82	68.58 / 95.35	67.61 / 91.62	67.04 / 93.30	63.81 / 78.48
	GAT+USP	85.02 / 86.10	85.20 / 73.77	74.63 / 87.34	83.53 / 92.71	82.62 / 96.03	85.09 / 94.53	69.59 / 89.33	70.66 / 78.89
	GAT+GRASP	69.97 / 87.02	70.22 / 74.40	64.10 / 87.07	75.73 / 93.21	68.54 / 96.18	69.65 / 95.03	64.00 / 93.70	63.25 / 79.63

information about the graph. Moreover, the proposed mechanism has low computational overhead, as the noise injection process is straightforward and can be directly applied to the node embeddings. Details of the theoretical analysis for formal privacy guarantees are provided in Appendix A.

(2) Integration with varying GNN Architectures. GRASP distinguishes itself through its model-agnostic design. Unlike existing DP-GNN methods (e.g., GAP [30], DPDGC [3]), which require specialized architectures for privacy preservation, GRASP employs a structured perturbation mechanism that serves as a plug-and-play module. This mechanism can be seamlessly integrated into any GNN architecture without modifying their internal structures.

5 Experiments

In this section, we conduct a comprehensive evaluation of the proposed GRASP model on benchmark datasets, with the aim of answering the following research questions:

- **RQ1:** Does GRASP outperform state-of-the-art methods in privacy-utility trade-off when defending against GRA?
- **RQ2:** Can GRASP be integrated into existing DP-GNNs to enhance their privacy-utility trade-off against GRA?
- **RQ3:** What is the sensitivity of GRASP with respect to the Bernoulli probability p of integrating two types of noise?

5.1 Experimental Setup

5.1.1 Datasets. We select eight classic graph benchmark datasets for node classification. Cora, CiteSeer, and PubMed are three widely used citation networks [31]. Computer and Photo [32] are co-purchase networks where nodes represent goods and edges indicate that the connected goods are frequently bought together. CS and Physics [32] are co-authorship networks where nodes denote authors and edges represent that the authors have co-authored at least one paper. We adhere to the widely accepted practice of training/validation/test splits of 60%/20%/20% [23]. Furthermore, we utilize the WikiCS dataset and use the official splits provided in [26]. Details of the datasets are provided in Appendix B.1.

5.1.2 Baselines. GAP [30] and DPDGC [3] are state-of-the-art DP-GNN methods that design specialized model architectures for graph privacy protection. Both methods incorporate unstructured perturbation into the node embeddings, and have been proven to satisfy edge-level DP guarantees. MC-GPB [45] is an advanced regularization-based method, which reduces the mutual information between node embeddings and the graph structure via an explicit loss function. Moreover, we integrate unstructured perturbation (USP) adopted by existing DP-GNNs and our proposed GRASP into the basic GCN [14] and GAT [34], yielding the models GCN+USP and GAT+USP, as well as GCN+GRASP and GAT+GRASP.

5.1.3 Settings. Our proposed GRASP method is model-agnostic. Recent work [23] fine-tuned the hyperparameters of GCN and GAT on classic graph benchmark datasets, and we directly use their model settings as the backbone. For the baselines GAP, DPDGC, and MC-GPB, we use their publicly available code and the hyperparameter search ranges proposed in the respective papers. Details of the settings are provided in Appendix B.2.

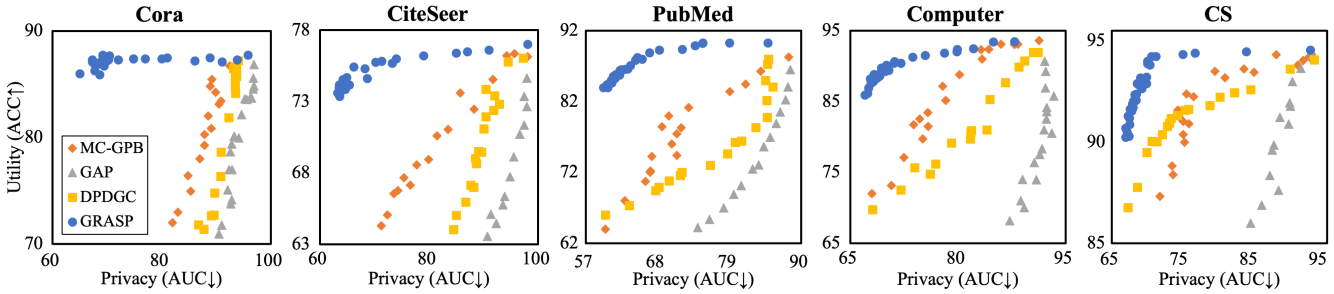
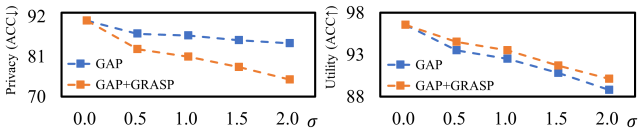
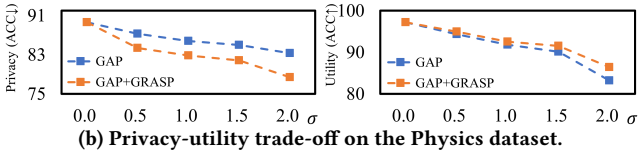


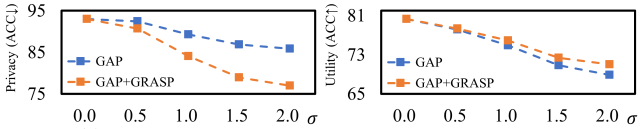
Figure 4: Comparison of privacy-utility trade-off with baseline methods. The points in the figures represent experimental results under different hyperparameter settings (e.g., noise intensity for DP-GNNs, regularization weight for MC-GPB).



(a) Privacy-utility trade-off on the Photo dataset.



(b) Privacy-utility trade-off on the Physics dataset.



(c) Privacy-utility trade-off on the WikiCS dataset.

Figure 5: Comparison of privacy and utility performance by integrating GRASP into GAP. Existing DP-GNN methods integrating our proposed GRASP achieve a better privacy-utility trade-off in defending against GRA.

5.1.4 Metrics. Privacy protection methods often trade off privacy and utility in downstream tasks. GRA can be viewed as an edge-level binary classification problem, and we use the AUC metric to evaluate the effectiveness of protecting graph structure in defending against GRA. The ACC metric is used to assess the model's utility in node classification tasks. Details are provided in Appendix B.3.

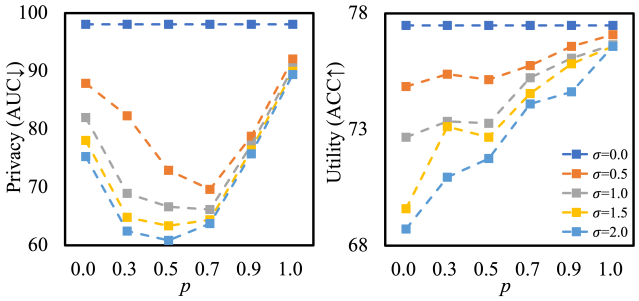
5.1.5 Privacy-Utility Trade-off Comparison (RQ1). To assess the effectiveness of the proposed GRASP model, we first validate the limitations of unstructured perturbations (USP) used in state-of-the-art DP-GNNs, such as GAP and DPDGC, which add independent noise to node embeddings. For comparison, we integrate USP into the basic GCN and GAT architectures, resulting in GCN+USP and GAT+USP models. Additionally, we incorporate GRASP into these architectures to create GCN+GRASP and GAT+GRASP models. We train and evaluate these models with varying Gaussian noise standard deviations, measuring privacy protection through GRA performance and utility through downstream node classification tasks. The results are shown in Table 1.

As the noise standard deviation increases, both methods show a trade-off between privacy and utility, with the performance of GRA and downstream tasks decreasing. Our method, however, significantly reduces GRA performance at low noise levels (e.g., $\sigma = 0.5, 1.0$), effectively preserving graph structure privacy, whereas unstructured perturbations are less effective at such noise levels. Furthermore, although utility in downstream tasks decreases with increasing noise, GRASP consistently outperforms USP-based methods. We attribute this to the structured perturbation mechanism of GRASP, which uses more identical noise, reducing randomness and lessening the impact on downstream tasks. Overall, GRASP's structured perturbation mechanism enhances GRA defense and achieves a better privacy-utility trade-off.

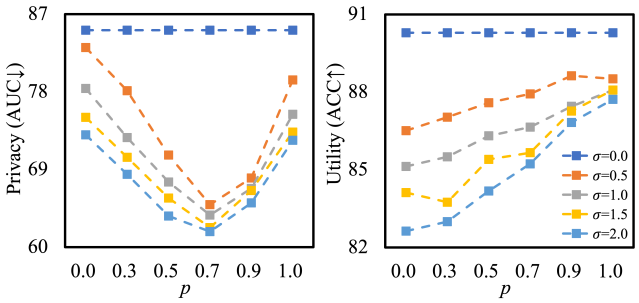
To fairly compare the privacy-utility trade-offs with baseline methods, we evaluated model performance under varying hyperparameters (e.g., noise intensity for DP-GNNs and regularization weight for MC-GPB), as shown in Figure 4. Our experiments show that DP-GNN methods such as GAP and DPDGC, which use unstructured perturbations, fail to adequately disrupt the relative ranking of embedding similarities, leaving them vulnerable to GRA. Although MC-GPB incorporates a regularization term to mitigate GRA, it degrades node embedding quality. In contrast, GRASP employs a structured perturbation mechanism, inducing bidirectional shifts in embedding similarity. Consequently, GRASP more effectively defends against GRA while preserving downstream utility, achieving a superior privacy-utility trade-off.

5.1.6 Integration with Existing DP-GNNs (RQ2). To further validate the effectiveness of GRASP's structured perturbation in defending against GRA, we integrate it into GAP, a DP-GNN method relying on unstructured perturbation, and compare privacy-utility trade-offs. As shown in Figure 5, the GAP model integrating GRASP demonstrates enhanced resistance against GRA while simultaneously improving utility on downstream tasks. This improvement originates from structured perturbation achieving sufficient embedding similarity disruption while reducing unnecessary noise randomness compared to unstructured perturbation.

5.1.7 Sensitivity Analysis (RQ3). Our proposed GRASP introduces a structured perturbation mechanism, integrating identical and independent Gaussian noise with Bernoulli probability p. To investigate the effect of p on graph structure protection and downstream task utility, we conduct experiments on the CiteSeer and PubMed datasets. We vary the value of p to evaluate its impact, where $p = 0.0$ corresponds to the use of purely independent noise (as used in existing DP-GNNs), and $p = 1.0$ represents the use of entirely identical



(a) Privacy and utility performance on the CiteSeer dataset.



(b) Privacy and utility performance on the PubMed dataset.

Figure 6: Performance of GRASP with varying hyperparameter p. As p increases, the effectiveness of GRA initially decreases and then increases, while the utility in node classification tasks tends to improve.

noise. These experiments are conducted under different noise standard deviations σ to assess the trade-offs in privacy protection and utility. The results are shown in Figure 6.

From the privacy curves, we observe that as p increases, the GRA performance initially decreases and then increases. This suggests that neither pure identical noise nor independent noise provides the optimal privacy protection for graph structure, as they only induce unidirectional shifts in embedding similarities. By integrating both types of noise, our method achieves better privacy protection against GRA. Regarding the utility curves, we notice that as p increases, the utility in downstream tasks tends to improve. This can be attributed to the increasing proportion of identical noise, which introduces less randomness compared to independent noise, thus reducing the disruption to downstream tasks.

6 Related Work

6.1 Graph Reconstruction Attacks

Graph reconstruction attacks (GRAs), also known as link stealing attacks [10, 15], aim to infer graph structural information from GNNs. Existing research has explored various attack scenarios with different levels of adversary capabilities and prior knowledge. When attackers can only access node embeddings, similarity-based GRAs have shown significant effectiveness. [5] empirically demonstrated that pairwise node similarities in GNN embeddings can reveal structural patterns. [2] further provided both theoretical and empirical evidence for reconstructing graphs from DeepWalk [29] embeddings. Recently, [37] established a theoretical framework for similarity-based GRAs. For adversaries with black-box access to GNN models, [10] systematically studied attack strategies under different prior knowledge conditions, including node features, partial graph structures, and shadow datasets. In white-box attack scenarios where model internals are accessible, [43] formulated GRA as an optimization problem that maximizes node classification accuracy using gradient-based methods. [45] introduced an information-theoretic approach to exploit intermediate representations in GNNs. Recent studies also investigate adversaries capable of modifying graph data. [36] proposed LinkTeller, which perturbs node features to expose structural vulnerabilities through influence analysis. [25] demonstrated that injecting malicious nodes into graphs can enable edge inference. Despite the diversity of attack methods, attacks requiring stronger assumptions achieve only limited performance improvements compared to embedding-based GRAs [37]. Therefore, similarity-based GRAs deserve prioritized attention due to their practical relevance—they require minimal prior knowledge and pose immediate threats in real-world applications.

6.2 Graph Reconstruction Defenses

GRAs can be seen as edge-level membership inference attacks [11, 12, 16, 33]. Differential privacy (DP) [7, 24, 44] offers a principled defense framework by formally limiting structural information leakage. Recent works [3, 30, 36] integrate edge-level DP into GNNs to protect sensitive graph topologies. While standard DP training [1] ensures provable privacy during model training, it fails to protect against inference-phase attacks [3]. To address this, existing defenses primarily inject noise into GNN computations. Early approaches [36] applied edge perturbation before GNN training, but such aggressive preprocessing severely degrades utility. Modern DP-GNN frameworks [3, 30, 37, 38] instead perturb node embeddings during aggregation layers—this preserves utility while ensuring privacy guarantees across both training and inference at minimal computational cost. Alternative methods based on information bottleneck principles [35, 45] attempt to minimize privacy leakage through regularization or adversarial training. However, these approaches lack formal privacy guarantees and introduce significant computational overhead [37]. DP-GNNs currently represent the state-of-the-art defense paradigm due to three key advantages: end-to-end protection across all learning phases, mathematically bounded privacy risks, and practical computational efficiency [30]. Nevertheless, they neither evaluate defense effectiveness against graph reconstruction attacks nor optimize noise injection mechanisms for edge-level privacy preservation.

7 Conclusion

In this paper, we demonstrate that existing DP-GNNs remain vulnerable to GRA, as their unstructured perturbations inadequately disrupt the relative ranking of embedding similarities. To address this, we propose GRASP, a novel DP-GNN that introduces structured perturbations by integrating both independent and identical noise, inducing bidirectional shifts in similarity distribution to invalidate GRA. Experiments confirm GRASP's superior defense against GRA and improved privacy-utility trade-offs. Future work will investigate the theoretical foundations of structured perturbations for preserving graph structural privacy under DP frameworks.

Acknowledgments

The research work is supported by National Key R&D Plan No. 2022YFC3303302, the National Natural Science Foundation of China under Grant No. U2436209, 62476263, 62406307. Xiang Ao is also supported by the Beijing Nova Program No. 20230484430, the Innovation Funding of ICT, CAS under Grant No. E461060. Yang Liu is also supported by the Postdoctoral Fellowship Program of CPSF under Grant No. GZB20240761.

References

[1] Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. 2016. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*. 308–318.

[2] Sudhanshu Chanpuriya, Cameron Musco, Konstantinos Sotiropoulos, and Charalampos Tsourakakis. 2021. Deepwalking backwards: From embeddings back to graphs. In *International Conference on Machine Learning*. PMLR, 1473–1483.

[3] Eli Chien, Wei-Ning Chen, Chao Pan, Pan Li, Ayfer Ozgur, and Olgica Milenkovic. 2023. Differentially private decoupled graph convolutions for multigranular topology protection. In *Advances in Neural Information Processing Systems*, Vol. 36.

[4] Yuanhao Ding, Yang Liu, Yugang Ji, Weigao Wen, Qing He, and Xiang Ao. 2025. SPEAR: A structure-preserving manipulation method for graph backdoor attacks. In *Proceedings of the ACM Web Conference 2025*. 1237–1247.

[5] Vasisht Duddu, Antoine Boutet, and Virat Shejwalkar. 2020. Quantifying privacy leakage in graph embedding. In *MobiQuitous 2020-17th EAI International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services*. 76–85.

[6] Cynthia Dwork. 2006. Differential privacy. In *International Colloquium on Automata, Languages, and Programming*. Springer, 1–12.

[7] Cynthia Dwork. 2008. Differential privacy: A survey of results. In *International Conference on Theory and Applications of Models of Computation*. Springer, 1–19.

[8] Faqian Guan, Tianqing Zhu, Wanlei Zhou, and Kim-Kwang Raymond Choo. 2024. Graph neural networks: A survey on the links between privacy and security. *Artificial Intelligence Review* 57, 2 (2024), 40.

[9] Will Hamilton, Zhitao Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. In *Advances in Neural Information Processing Systems*, Vol. 30.

[10] Xinlei He, Jinyuan Jia, Michael Backes, Neil Zhenqiang Gong, and Yang Zhang. 2021. Stealing links from graph neural networks. In *30th USENIX security symposium*. 2669–2686.

[11] Xinlei He, Rui Wen, Yixin Wu, Michael Backes, Yun Shen, and Yang Zhang. 2021. Node-level membership inference attacks against graph neural networks. *arXiv preprint arXiv:2102.05429* (2021).

[12] Hongsheng Hu, Zoran Salcic, Lichao Sun, Gillian Dobbie, Philip S Yu, and Xuyun Zhang. 2022. Membership inference attacks on machine learning: A survey. *Comput. Surveys* 54, 11s (2022), 1–37.

[13] Mengda Huang, Yang Liu, Xiang Ao, Kuan Li, Jianfeng Chi, Jinghua Feng, Hao Yang, and Qing He. 2022. AUC-oriented graph neural network for fraud detection. In *Proceedings of the ACM Web Conference 2022*. 1311–1321.

[14] Thomas N Kipf and Max Welling. 2017. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations*.

[15] Aashish Kolluri, Teodora Baluta, Bryan Hooi, and Prateek Saxena. 2022. LPGNet: Link private graph networks for node classification. In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*. 1813–1827.

[16] Jiacheng Li, Ninghui Li, and Bruno Ribeiro. 2021. Membership inference attacks and defenses in classification models. In *Proceedings of the Eleventh ACM Conference on Data and Application Security and Privacy*. 5–16.

[17] Kuan Li, YiWen Chen, Yang Liu, Jin Wang, Qing He, Minhao Cheng, and Xiang Ao. 2024. Boosting the adversarial robustness of graph neural networks: An OOD perspective. In *International Conference on Learning Representations*.

[18] Kuan Li, Yang Liu, Xiang Ao, Jianfeng Chi, Jinghua Feng, Hao Yang, and Qing He. 2022. Reliable representations make a stronger defender: Unsupervised structure refinement for robust GNN. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 925–935.

[19] Kuan Li, Yang Liu, Xiang Ao, and Qing He. 2023. Revisiting graph adversarial attack and defense from a data distribution perspective. In *International Conference on Learning Representations*.

[20] Yang Liu, Xiang Ao, Fuli Feng, and Qing He. 2022. UD-GNN: Uncertainty-aware debiased training on semi-homophilous graphs. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 1131–1140.

[21] Yang Liu, Xiang Ao, Fuli Feng, Yunshan Ma, Kuan Li, Tat-Seng Chua, and Qing He. 2023. FLOOD: A flexible invariant learning framework for out-of-distribution generalization on graphs. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 1548–1558.

[22] Yang Liu, Xiang Ao, Zidi Qin, Jianfeng Chi, Jinghua Feng, Hao Yang, and Qing He. 2021. Pick and choose: A GNN-based imbalanced learning approach for fraud detection. In *Proceedings of the ACM Web Conference 2021*. 3168–3177.

[23] Yuankai Luo, Lei Shi, and Xiao-Ming Wu. 2024. Classic GNNs are strong baselines: Reassessing gnns for node classification. In *The Thirty-eight Conference on Neural Information Processing Systems*.

[24] Frank McSherry and Kunal Talwar. 2007. Mechanism design via differential privacy. In *48th IEEE Annual Symposium on Foundations of Computer Science*. IEEE, 94–103.

[25] Lingshuo Meng, Yijie Bai, Yanjiao Chen, Yutong Hu, Wenyuan Xu, and Haiqin Weng. 2023. Devil in disguise: Breaching graph neural networks privacy through infiltration. In *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security*. 1153–1167.

[26] Péter Mernyei and Cătălina Cangea. 2020. Wiki-cs: A wikipedia-based benchmark for graph neural networks. *arXiv preprint arXiv:2007.02901* (2020).

[27] Ilya Mironov. 2017. Rényi differential privacy. In *2017 IEEE 30th Computer Security Foundations Symposium*. IEEE, 263–275.

[28] Iyiola E Olatunji, Wolfgang Nejdl, and Megha Khosla. 2021. Membership inference attack on graph neural networks. In *2021 Third IEEE International Conference on Trust, Privacy and Security in Intelligent Systems and Applications*. IEEE, 11–20.

[29] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. 2014. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 701–710.

[30] Sina Sajadmanesh, Ali Shahin Shamsabadi, Aurélien Bellet, and Daniel Gatica-Perez. 2023. GAP: Differentially private graph neural networks with aggregation perturbation. In *32nd USENIX Security Symposium*. 3223–3240.

[31] Prithviraj Sen, Galileo Namata, Mustafa Bilgic, Lise Getoor, Brian Galligher, and Tina Eliassi-Rad. 2008. Collective classification in network data. *AI Magazine* 29, 3 (2008), 93–93.

[32] Oleksandr Shchur, Maximilian Mumme, Aleksandar Bojchevski, and Stephan Günnemann. 2018. Pitfalls of graph neural network evaluation. *arXiv preprint arXiv:1811.05868* (2018).

[33] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. 2017. Membership inference attacks against machine learning models. In *2017 IEEE Symposium on Security and Privacy*. IEEE, 3–18.

[34] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. 2017. Graph attention networks. *arXiv preprint arXiv:1710.10903* (2017).

[35] Binghui Wang, Jiayi Guo, Ang Li, Yiran Chen, and Hai Li. 2021. Privacy-preserving representation learning on graphs: A mutual information perspective. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 1667–1676.

[36] Fan Wu, Yunhui Long, Ce Zhang, and Bo Li. 2022. LinkTeller: Recovering private edges from graph neural networks via influence analysis. In *2022 IEEE Symposium on Security and Privacy*. IEEE, 2005–2024.

[37] Ruofan Wu, Guanhua Fang, Mingyang Zhang, Qiying Pan, Tengfei Liu, and Weiqiang Wang. 2024. On provable privacy vulnerabilities of graph representations. In *Advances in Neural Information Processing Systems*, Vol. 37. 90891–90933.

[38] Ruofan Wu, Mingyang Zhang, Lingjuan Lyu, Xiaolong Xu, Xiuquan Hao, Xinyi Fu, Tengfei Liu, Tianyi Zhang, and Weiqiang Wang. 2023. Privacy-preserving design of graph neural networks with applications to vertical federated learning. In *NeurIPS 2023 Workshop: New Frontiers in Graph Learning*.

[39] Jingjing Xu, Xu Sun, Zhiyuan Zhang, Guangxiang Zhao, and Junyang Lin. 2019. Understanding and improving layer normalization. In *Advances in Neural Information Processing Systems*, Vol. 32.

[40] Qi Yuan, Yang Liu, Yateng Tang, Xinhuan Chen, Xuehao Zheng, Qing He, and Xiang Ao. 2025. Dynamic graph learning with static relations for credit risk assessment. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 39. 13133–13141.

[41] Zhikun Zhang, Min Chen, Michael Backes, Yun Shen, and Yang Zhang. 2022. Inference attacks against graph neural networks. In *31st USENIX Security Symposium*. 4543–4560.

[42] Zaixi Zhang, Qi Liu, Zhenya Huang, Hao Wang, Chee-Kong Lee, and Enhong Chen. 2022. Model inversion attacks against graph neural networks. *IEEE Transactions on Knowledge and Data Engineering* 35, 9 (2022), 8729–8741.

[43] Zaixi Zhang, Qi Liu, Zhenya Huang, Hao Wang, Chengqiang Lu, Chuanren Liu, and Enhong Chen. 2021. GraphMI: Extracting private graph data from graph neural networks. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence*. 3749–3755.

[44] Ying Zhao and Jinjun Chen. 2022. A survey on differential privacy for unstructured data content. *Comput. Surveys* 54, 10s (2022), 1–28.

[45] Zhanke Zhou, Chenyu Zhou, Xuan Li, Jiangchao Yao, Quanming Yao, and Bo Han. 2023. On strengthening and defending graph reconstruction attack with markov chain approximation. In *Proceedings of the 40th International Conference on Machine Learning*.

A Theoretical Analysis

Theory 1 (Edge-Level Differential Privacy Guarantee of GRASP). *Let $Agg(X, A) = A^T X$ be the summation aggregation function in GRASP, where the input feature matrix X is row-normalized such that $\forall v \in \mathcal{V} : \|X_v\|_2 = 1$. Then, the edge-level sensitivity of GRASP's aggregation function satisfies $\Delta_{Agg} = 1$. By applying the Gaussian mechanism with noise scale σ to the aggregated embeddings, GRASP guarantees $(\alpha, \alpha/2\sigma^2)$-Rényi Differential Privacy (RDP) at the edge level.*

Proof. We adapt the proof of Lemma 1 and Theorem 1 in [30] to our GRASP method. Let A and A' be adjacency matrices of two edge-level adjacent graphs differing in a single edge (u, v). Without loss of generality, assume $A_{v,u} = 1$ and $A'_{v,u} = 0$. Let $M = \text{Agg}(X, A)$ and $M' = \text{Agg}(X, A')$ denote the aggregation outputs. The Frobenius norm of their difference is bounded as:

$$\|M - M'\|_F = \left(\sum_{i=1}^{N} \|M_i - M'_i\|_2^2 \right)^{1/2} \tag{15}$$

For any node $i \neq u$, the i-th row satisfies $M_i = M'_i$ because the adjacency matrices differ only at entry (v, u). For node u, we have:

$$\|M_u - M'_u\|_2 = \left\| \sum_{j=1}^{N} (A_{j,u} - A'_{j,u}) X_j \right\|_2 = \|(A_{v,u} - A'_{v,u}) X_v\|_2 \tag{16}$$

Substituting $A_{v,u} - A'_{v,u} = 1$ and using the row-normalization condition $\|X_v\|_2 = 1$:

$$\|M_u - M'_u\|_2 = \|X_v\|_2 = 1 \tag{17}$$

Thus, the overall Frobenius norm becomes:

$$\|M - M'\|_F = \left(0 + \cdots + \|M_u - M'_u\|_2^2 + \cdots + 0\right)^{1/2} = 1 \tag{18}$$

This demonstrates the edge-level sensitivity $\Delta_{\text{Agg}} = 1$ for GRASP's aggregation function. Following [27], adding Gaussian noise $\mathcal{N}(0, \sigma^2 I)$ to the aggregated embeddings ensures $(\alpha, \alpha/2\sigma^2)$-RDP per operation. As GRASP applies this mechanism only once, the total privacy budget is:

$$\epsilon(\alpha) = \alpha/2\sigma^2 \tag{19}$$

Therefore, GRASP satisfies $(\alpha, \alpha/2\sigma^2)$-RDP at the edge level, completing the proof. □

B Experimental Details

B.1 Datasets

We evaluate our method on eight widely used graph benchmark datasets spanning multiple domains. Table 2 summarizes their statistical characteristics. Below we describe the contextual semantics of each dataset:

Cora, CiteSeer, and PubMed [31] are citation networks where nodes represent academic papers and edges denote citation relationships. Node features are bag-of-words representations of paper contents, and classes correspond to academic topics.

Computer and Photo [32] are Amazon co-purchase networks where nodes represent products, and edges indicate that two goods are frequently bought together. Features describe product reviews, while classes correspond to product categories.

CS and Physics [32] are co-authorship networks where nodes represent authors, with edges denoting co-authorship of at least one paper. Node features contain keywords from authors' papers, and classes represent research subfields.

WikiCS [26] is a network of Computer Science-related Wikipedia articles. Nodes represent articles, and edges indicate hyperlinks between them. Features are derived from article embeddings, with classes corresponding to article categories.

Table 2: Statistics of benchmark datasets.

Dataset	Nodes	Edges	Features	Classes
Cora	2,708	5,278	1,433	7
CiteSeer	3,327	4,522	3,703	6
PubMed	19,717	44,324	500	3
Computer	13,752	245,861	767	10
Photo	7,650	119,081	745	8
CS	18,333	81,894	6,805	15
Physics	34,493	247,962	8,415	5
WikiCS	11,701	216,123	300	10

B.2 Hyperparameters

Our proposed GRASP method is model-agnostic and compatible with various graph neural network architectures. Following recent work [23] that systematically optimized hyperparameters for classic GNN baselines, we directly adopt their fine-tuned settings for GCN and GAT backbones without additional modifications. Tables 3 and 4 detail the dataset-specific configurations for GCN and GAT respectively, including normalization strategies, dropout rates, and architectural parameters.

Table 3: GCN hyperparameter settings across datasets.

Dataset	ResNet	Dropout	Layers	Hidden	LR
Cora	No	0.7	3	512	0.001
CiteSeer	No	0.5	2	512	0.001
PubMed	No	0.7	2	256	0.005
Computer	No	0.5	3	512	0.001
Photo	Yes	0.5	6	256	0.001
CS	Yes	0.3	2	512	0.001
Physics	Yes	0.3	2	64	0.001
WikiCS	No	0.5	3	256	0.001

B.3 Metrics

To evaluate the effectiveness of privacy protection against graph reconstruction attacks (GRA), we adopt the AUC (Area Under the ROC Curve) metric. Let $\mathbf{A} \in \{0, 1\}^{N \times N}$ denote the ground-truth adjacency matrix, and $\hat{\mathbf{A}}^*$ represent the predicted adjacency matrix derived from node embeddings $\mathbf{H}_i$ and $\mathbf{H}_j$ through similarity computation and sigmoid activation:

$$\hat{\mathbf{A}}^*_{ij} = \sigma\left(\text{Sim}(\mathbf{H}_i, \mathbf{H}_j)\right), \tag{20}$$

where $\text{Sim}(\cdot)$ measures embedding similarity (e.g., cosine similarity) and $\sigma(\cdot)$ is the sigmoid function. The AUC score is computed by

Table 4: GAT hyperparameter settings across datasets.

Dataset	ResNet	Dropout	Layers	Hidden	LR
Cora	Yes	0.2	3	512	0.001
CiteSeer	Yes	0.5	3	256	0.001
PubMed	No	0.5	2	512	0.01
Computer	No	0.5	2	64	0.001
Photo	Yes	0.5	3	64	0.001
CS	Yes	0.3	1	256	0.001
Physics	Yes	0.7	2	256	0.001
WikiCS	Yes	0.7	2	512	0.001

comparing the predicted probabilities $\hat{\mathbf{A}}^*_{ij}$ against the ground-truth edges $\mathbf{A}_{ij}$ across all node pairs (i, j). This metric quantifies how well the protection method obscures edge existence while considering both true positive and false positive rates across all classification thresholds.

For utility evaluation in node classification tasks, we use the ACC (Accuracy) metric. Let $\mathbf{Y} \in \{0, 1\}^{N \times C}$ be the one-hot encoded ground-truth labels for N nodes and C classes, and $\hat{\mathbf{Y}}$ denote the predicted class probabilities from the model. The classification loss $\mathcal{L}(\theta)$ is computed via cross-entropy:

$$\mathcal{L}(\theta) = -\frac{1}{N} \sum_{i=1}^{N} \sum_{c=1}^{C} \mathbf{Y}_{i,c} \log(\hat{\mathbf{Y}}_{i,c}). \tag{21}$$

The ACC metric is derived by comparing the predicted labels $\arg\max_c \hat{\mathbf{Y}}_{i,c}$ with the ground-truth labels $\arg\max_c \mathbf{Y}_{i,c}$:

$$\text{ACC} = \frac{1}{N} \sum_{i=1}^{N} \mathbb{I}\left(\arg\max_c \hat{\mathbf{Y}}_{i,c} = \arg\max_c \mathbf{Y}_{i,c}\right), \tag{22}$$

where $\mathbb{I}(\cdot)$ is an indicator function. This measures the proportion of correctly classified nodes, reflecting the model's utility preservation under privacy protection.