Unveiling Graph Copycats: Inference Attacks with Student Models

Paul Agbaje
University of Texas at Arlington
Arlington, Texas, USA
pauloluwatowoju.agbaje@uta.edu

Afia Anjum
University of Texas at Arlington
Arlington, Texas, USA
afia.anjum@uta.edu

Arkajyoti Mitra
University of Texas at Arlington
Arlington, Texas, USA
arkajyoti.mitra@uta.edu

Habeeb Olufowobi
University of Texas at Arlington
Arlington, Texas, USA
habeeb.olufowobi@uta.edu

Abstract

Graph Neural Networks (GNNs) are deep learning models designed to address the complexities of graph-structured, non-Euclidean data. Due to their complexity, knowledge distillation (KD) is often employed to transfer knowledge from a GNN to a simpler, more efficient student model, such as a Multi-Layer Perceptron (MLP), enabling deployment in large-scale industrial applications. However, KD can inadvertently leak sensitive information from the teacher to the student, posing significant privacy risks. We present the first membership inference attacks targeting GNNs in KD pipeline, showing that student MLPs can reveal whether a node appeared in the teacher's training data. Our attacks operate in a black-box setting, requiring access only to the student outputs, and remain effective in cross-dataset scenarios. Experimental evaluations across four GNN models and eight datasets show the effectiveness of our approach, achieving up to 0.9014 precision under low FPR of 1% in cross-dataset settings. These results expose significant vulnerabilities in GNN-based KD frameworks, emphasizing the need for strong security measures during the KD process involving GNNs.

Keywords

Graph Neural Networks, Membership Inference Attack, Distillation

1 Introduction

Deep learning (DL) has shown impressive capabilities across diverse tasks, from computer vision to natural language processing and bioinformatics [8]. However, many DL architectures assume Euclidean input structures, limiting their effectiveness on inherently non-Euclidean data such as social networks, transaction graphs, or biomedical interaction networks [5]. Graph neural networks (GNNs) overcome this limitation by modeling both node features and graph structure, enabling state-of-the-art results in various graph-based applications [31].

Despite their expressiveness, deploying GNNs in real-world systems remains challenging due to their increasing complexity and memory demands. Knowledge distillation (KD)—transferring knowledge from a large teacher model to a lightweight student model—has emerged as a promising strategy for scaling GNN-based solutions [16]. In practice, GNNs are often distilled into simpler, non-graph based models such as Multi-Layer Perceptrons (MLPs), which can be deployed to latency-sensitive or resource-constrained environments [46]. These MLPs eliminate the need for graph structure at inference time, enabling faster, parallelizable, and hardware-friendly predictions.

Beyond efficiency, KD is increasingly used to enhance data privacy [23]. The rationale is intuitive: distilling knowledge from a teacher reduces direct reliance on sensitive training data, potentially mitigating privacy risks. This approach often involves training the teacher model using non-private methods to maximize performance during the distillation process. This trend raises a critical question: **Can student MLPs distilled from GNNs leak private information about the teacher's training data?** Prior work has shown that student models can inadvertently retain sensitive patterns from their teachers [19], making them vulnerable to membership inference attacks (MIAs). However, these findings focus primarily on vision and language domains, where both teacher and student share the same data modality and structure. The privacy implications of distilling GNNs—models with strong inductive structural biases—into MLPs remain unexplored.

This paper presents the *first systematic study* of MIAs against GNNs in KD settings, targeting scenarios where: (i) the student MLP is the only observable model *(black-box access)*, (ii) the teacher GNN, training data, and graph structure are completely hidden, and (iii) the student is trained on a dataset that may differ from the teacher's, simulating realistic cross-dataset deployment. Such settings arise in real-world ML pipelines where only distilled models are shared for deployment or fine-tuning. While this workflow is motivated by privacy concerns, we demonstrate that it can have unintended consequences and introduce new exploitable vulnerabilities.

We focus on GNN-to-MLP KD as an extreme cross-modality case that isolates the impact of removing relational structure at inference. This abstraction exposes leakage mechanisms that arise from the distillation process itself rather than from architectural overlap. While smaller GNN students may retain partial structure by sharing the teacher's modality, our findings demonstrate non-trivial privacy risks that persist even in this most structurally disjoint setting. In this extreme setting, GNNs encode local and global signals through message passing (MP). When this knowledge is distilled to

This work is licensed under the Creative Commons Attribution 4.0 International License. To view a copy of this license visit https://creativecommons.org/licenses/by/4.0/ or send a letter to Creative Commons, PO Box 1866, Mountain View, CA 94042, USA.
Proceedings on Privacy Enhancing Technologies YYYY(X), 1–18
© YYYY Copyright held by the owner/author(s).
https://doi.org/XXXXXXX.XXXXXXX

an MLP—which lacks explicit graph structure—it is unclear whether any of these relational signals persist post-distillation, and critically, whether they can be exploited. The architectural mismatch between teacher and student further complicates privacy auditing. Specifically, does the student preserve latent graph-specific biases despite lacking structural awareness? Conversely, does the teacher's MP obscure membership information by smoothing over individual node features?

Our Contributions. We introduce a modular and statistically rigorous attack framework for auditing privacy risks in knowledge-distilled student models. Our contributions are:

- We formalize the *first black-box privacy attack targeting the training data of a GNN via its student MLP*, requiring no access to the teacher model or graph structure, revealing practical privacy vulnerabilities that persist even in distilled models.
- We develop a modular attack framework that combines feature extraction and classification to infer teacher membership solely from student outputs in KD setting. Empirical results show that student models can leak sensitive training information.
- We demonstrate attack generalizability to *cross-dataset scenarios*, using a quantile regression (QR) method that maintains strong inference performance with low false positive rates (FPR).
- We propose a conformalized quantile regression (CQR) variant that provides tighter guarantees, revealing privacy vulnerabilities under stricter decision thresholds.
- We extend our analysis beyond membership inference through diagnostic *property- and link-inference attacks* by probing whether distilled student models retain semantic or relational patterns from the teacher. These diagnostics reveal that privacy leakage is not confined to individual membership but extends to broader structural and attribute-level information transfer.
- We evaluate our attacks on eight graph datasets and four common GNN architectures, demonstrating consistent vulnerabilities in student models post-KD.

Our findings reveal that KD does not eliminate privacy risks and highlight the need for stronger defenses when deploying student models derived from graph-based teachers.

2 Background and Threat Model

GNN. We define an attributed graph as $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{X})$, where $\mathcal{V}$ is the set of nodes, $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is the set of observed edges, and $\mathcal{X}$ is the attribute matrix. The graph structure is represented by its adjacency matrix $\mathcal{A} \in 0, 1^{|\mathcal{V}|\times|\mathcal{V}|}$, where $\mathcal{A}_{uv} = 1$ if an edge exists between nodes u and v, and $\mathcal{A}_{uv} = 0$ otherwise. Each node u has associated attributes $\mathcal{X}_u \in \mathbb{R}^n$, where n is the dimension of the node feature space. Thus, $\mathcal{X} \in \mathbb{R}^{|\mathcal{V}|\times n}$ represents the attribute matrix for all nodes. A GNN processes $\mathcal{G}$ to generate h-dimensional node representations $\mathcal{Z} \in \mathbb{R}^{|\mathcal{V}|\times h}$, where $\mathcal{Z}_u$ is the embedding for node u. These embeddings enable downstream tasks such as node classification, link prediction, and graph classification. Training and prediction processes for GNNs are detailed in Appendices A and B.

KD is a model compression technique that transfers knowledge from a complex teacher model to a smaller, lightweight student model. In KD, the student replicates the teacher's soft predictions, making it ideal for resource-constrained settings where achieving comparable accuracy with reduced computational overhead is required. KD has been applied to GNNs to enhance efficiency. For instance, TinyGNN [44] distills knowledge from a teacher GNN to a smaller student GNN, achieving scalability with minimal performance loss. However, the iterative message-passing mechanism in GNNs introduces computational and latency constraints for large graphs [39]. To address this, recent methods distill GNNs into simpler models, such as MLPs, bypassing MP. This shift enhances scalability and reduces computational overhead, making it suitable for real-world applications [12, 41, 50]. However, it remains unclear whether such distilled models retain, and potentially leak, privacy attributes of their teachers.

MIAs determine whether a sample was part of a target model training set [7]. These attacks exploit a model's tendency to overfit its data, where they show higher confidence in training examples than unseen examples from the same distribution [3]. In GNNs, MIAs can target individual nodes within a graph, raising privacy concerns, particularly in sensitive applications. These node-level MIAs aim to infer if a specific node was used during the training of the target GNN [15]. Formally, consider a node u, a target GNN f_{tar}, and a private training dataset $\mathcal{G}_{priv}$. The attack is expressed as a hypothesis test:

$$\begin{aligned} H_0 &: \quad u \notin \mathcal{G}_{priv} \quad (u \text{ not in training set}), \\ H_1 &: \quad u \in \mathcal{G}_{priv} \quad (u \text{ in training set}). \end{aligned} \tag{1}$$

To carry out the attack, the adversary defines a decision function $\mathcal{A}(f_{\text{tar}}, u)$ to predict membership as follows:

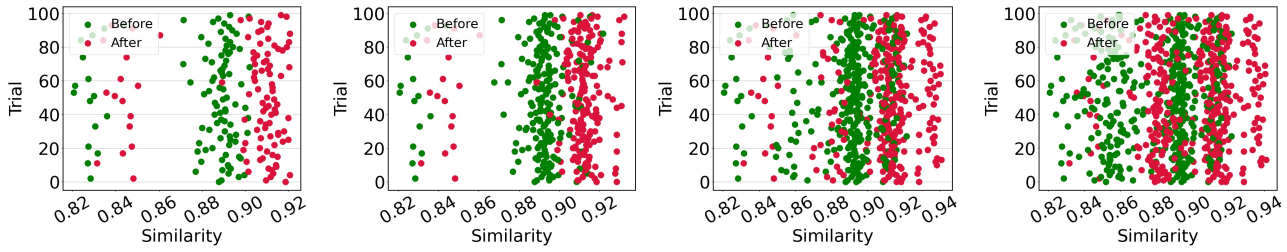
$$\mathcal{A}(f_{\text{tar}}, u) = \begin{cases} H_1 & \text{if } s(u) \geq \gamma, \\ H_0 & \text{otherwise,} \end{cases} \tag{2}$$

where $s(u)$ is a score function (e.g., confidence in predicted label for u), and γ is the membership threshold.

2.1 Motivation

Real-world graphs, such as those in social, medical, and financial networks, often contain private information [14, 21, 43, 47]. Adversaries can exploit models trained on such data and subsequently released to infer specific details about the training data, thereby compromising privacy. For instance, consider a financial system employing GNN for fraud detection/credit risk assessment. If an adversary deduces that a specific user's transaction data contributed to the training, it could expose sensitive patterns, such as involvement in transaction networks, leading to reputational harm or economic/market dynamics manipulation. In healthcare, where GNNs predict outcomes based on patient interaction networks, identifying that a patient's data was used in training could reveal confidential medical details. Such breaches might violate regulations like HIPAA [4], resulting in legal, financial, and ethical consequences.

To assess this risk, we consider a scenario where the original model is distilled into a student model. The student model is released with only API access for querying, and its embeddings can be uploaded to platforms like PyTorch BigGraph [27] for downstream tasks. To explore potential privacy risks, we distilled the teacher (a GNN) into a student (an MLP) and examined whether noticeable differences existed in prediction similarities before and after distillation. To rigorously test, we used student models with architectures



(a) Class 0 (b) Class 1 (c) Class 2 (d) Class 3

Figure 1: Cosine similarity of the output probabilities for the first four classes of Cora, comparing a teacher GCN and a student MLP. Each data point represents the cosine similarity between the probabilities predicted by the teacher GNN and student MLP.

different from the corresponding teacher models. Following the framework by Zhang et al. [46], we conducted experiments using two GNN architectures—GraphSAGE and GCN—and distilled their knowledge into MLPs. Each experiment was repeated for 100 trials.

We measured cosine similarities between the output probabilities of the teacher and student models, both before and after distillation. Fig. 1 shows scatter plots for the first four classes of the Cora dataset [32] using the GCN model. The rightward shift in student similarities post-distillation indicates closer alignment with the teacher's predictions. Additional results can be found in Appendix C. These findings show that the student model's predictions align more closely with those of the teacher model after distillation. This improved alignment suggests the possibility of inferring leaked information by examining the differences in the student's posterior distributions before and after distillation.

One key assumption in MIA is that a model f, trained on a private dataset $\mathcal{G}_{priv}$, may exhibit overconfidence on examples $(u, y) \in \mathcal{G}_{priv}$ [3]. When the teacher model $f_{tar} = f$ and the student model f_{st} align through distillation, as shown in Fig. 1, the student may *inherit this overconfidence*. This inheritance can potentially increase the student's vulnerability to MIA.

Building on these insights and the differences in prediction similarities before and after distillation, we investigate whether an adversary can leverage these differences to execute an MIA against the teacher model using the student's predictions. This line of inquiry is crucial for understanding the potential privacy risks inherent in model distillation and for developing robust privacy-preserving techniques.

Challenges. Jagielski et al. [19] observed that attack performance improves when shadow models mimic target architecture. However, our setup introduces *unique challenges*:

First, in our KD framework, the teacher model is a GNN, while the student model is an MLP. This architectural difference complicates the attack, as the adversary must infer information from the teacher's training set indirectly, despite the student's non-graph-based structure. *Second*, the adversary has no direct access to the teacher model, precluding direct comparison of the teacher's and student's posterior distributions. This constraint renders the attack black-box with respect to the target, necessitating alternative techniques to infer private information without direct access to the teacher's architecture or outputs. *Third*, while some scenarios assume a shared training set for both the teacher and student, we investigate cases where the student is trained on a dataset distinct from the teacher's. Also, the adversary lacks access to the datasets used for training, further limiting their capacity to mimic the teacher's behavior. Despite these constraints, even minimal information leaks can pose significant privacy risks, potentially enabling further attacks. For instance, in domains such as healthcare, such breaches could compromise patient confidentiality and result in severe regulatory and ethical consequences.

2.2 Threat Model

Adversary's Objective. The adversary seeks to determine whether a specific node u was part of the training graph G_{priv} used to train a proprietary GNN, referred to as the *teacher model* f_{tar}. This task constitutes a node-level MIA. Crucially, the adversary cannot interact with the teacher directly and only has query access to a *student model* f_{st}—an MLP—that was trained via KD from the teacher.

Assumptions and Capabilities. We explore a range of realistic adversarial capabilities, outlined along three axes:

- **Query Access.** The student model f_{st} is accessible only through a query interface: the adversary can submit input nodes and obtain posterior probabilities but cannot inspect parameters, gradients, or intermediate representations. For simplicity, we assume that the adversary knows the general architecture (e.g., the number of layers of the model) of f_{st}. This assumption simplifies experimental reproducibility, but is not strictly required for the attack's success. Empirical ablations in Sec. 6.2 confirm that the attack remains effective under architectural mismatches, demonstrating the robustness of the black-box assumption to structural uncertainty. This setting reflects realistic deployment scenarios such as model-as-a-service APIs or released distilled MLPs for downstream applications, where limited background information about model type may be available, but internal access is restricted.
- **Architectural Knowledge.** The adversary possesses partial knowledge of the student model architecture, such as the number of layers or general structure, but lacks hyperparameter details. This enables the adversary to train *shadow models* that approximate the student's behavior. We assume no access to the teacher model's architecture, weights, or training data.

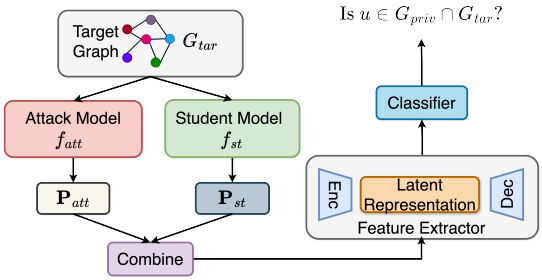


Figure 2: MIA attack pipeline where teacher and student models are trained on the same dataset.

- **Auxiliary Data.** The adversary has access to a a public or synthetically generated *auxiliary graph* G_{aux} sampled from a distribution similar to G_{priv}. This assumption reflects realistic conditions in which comparable graphs are publicly accessible, such as social or co-purchase networks obtainable through public APIs, or molecular graphs available in open repositories [47]. Importantly, G_{aux} functions as a controlled mechanism for uncovering potential leakage rather than a source of leakage itself. Moreover, in less aligned scenarios, such as when the auxiliary or teacher graphs are noisy (Sec. 7), the attack continues to achieve strong membership-inference performance, underscoring that privacy risks persist even under imperfect distributional alignment.

Training Scenarios. We analyze two distinct deployment scenarios that impact attack feasibility:

- **Same-Graph Distillation.** The teacher and student models are trained on the same private graph G_{priv}. This setup captures industrial reuse scenarios, where a GNN is distilled into an MLP and the student is deployed to reduce inference cost.
- **Cross-Graph Distillation.** The teacher is trained on a private graph G_{priv}, and the student is trained on a disjoint graph G_{pub}. This represents stronger privacy assumptions (e.g., data isolation policies or federated settings). Despite no data overlap, we show that the student model can still leak information about G_{priv}.

Attack Surface. Under these assumptions, the adversary's only observable signal is the posterior output of $f_{\text{st}}(u)$ for node u. Our attacks exploit subtle statistical differences in these outputs, induced by KD, to infer whether $u \in G_{\text{priv}}$, despite the student's architectural and training mismatch with the teacher.

Threat Model Tuple Notation. To clearly outline the specific threat scenarios associated with each attack types, we define a binary tuple: (QA, AK, AD, SGD), where each element denotes whether the adversary possesses: QA (query access), AK (partial architectural knowledge), AD (auxiliary data), and SGD (same-graph distillation). A checkmark (✔) indicates the condition holds, while a cross (✘) denote it does not. For example, the tuple (✔, ✔, ✔, ✘) represents a cross-graph setting where the adversary has full attack capabilities, except for access to shared training data.

3 Reconstruction error attack (✔, ✔, ✔, ✔)

Attack Overview. We propose a black-box MIA that exploits student model f_{st} posteriors trained via KD from a private teacher GNN. The key insight is that the *KD transfers confidence patterns from the teacher to the student*, and these patterns can leak membership information about the teacher's training data $\mathcal{G}_{priv}$. Our approach involves comparing the behavior of f_{st} with that of an auxiliary model f_{att} trained on a separate dataset $\mathcal{G}_{aux}$, sampled from the same distribution as $\mathcal{G}_{priv}$. While f_{st} carries inductive biases from the teacher, f_{att} does not. By measuring the divergence in predictions on the target graph $\mathcal{G}_{tar}$, we can obtain latent representations that help infer node membership. Our attack involves two main steps: (1) **Fusion:** Combines f_{st} and f_{att} predictions to generate contrastive features for each node, which are then encoded using a feature extractor. (2) **Classification:** Uses reconstruction error from the feature extractor to identify node representations that deviate from expected manifold patterns, indicating memorization—the persistence of teacher-induced confidence structures rather than explicit sample recall. Note that the student model employs dropout and batch normalization, so the observed memorization and leakage arise from inherited structural bias rather than under-regularization.

3.1 Fusion

The fusion pipeline consists of:

- f_{st}: A student MLP trained via KD from the teacher GNN. It inherits class-boundary information from the teacher, encoding membership-sensitive signals from G_{priv}.
- f_{att}: An auxiliary MLP trained on $\mathcal{G}_{aux}$ without KD. It provides a contrastive baseline for posterior behavior uninfluenced by private data, helping to surface distillation-induced deviations.
- f_{ext}: An autoencoder that compresses combined posteriors from f_{st} and f_{att} into low-dimensional vectors, filtering noise and preserving membership-relevant structure.

Process. We first compute posterior vectors from both models:

$$\mathbf{P}_{att} = f_{att}(\mathcal{G}_{tar}) \qquad \mathbf{P}_{st} = f_{st}(\mathcal{G}_{tar}) \tag{3}$$

We then combine and pass them through an encoder:

$$\mathbf{Z} = f_{ext}^{enc}(\text{combine}(\mathbf{P}_{att}, \mathbf{P}_{st})) \tag{4}$$

We employ five distinct element-wise combination methods (multiplication, ratio, subtraction, mean, and addition). Details of these methods are in Appendix E.

Training the Feature Extractor. The feature extractor is trained on $\mathcal{G}_{aux}$ using Mean Squared Error (MSE) to reconstruct the posteriors of each node in a graph. The goal is to minimize the reconstruction error between the original posteriors and the posteriors produced by the feature extractor decoder. The objective can be formulated as:

$$\min \quad \mathbb{E}_{u \sim \mathcal{G}_{aux}} \text{MSE}(f_{ext}(p_u), p_u) \tag{5}$$

where, $u \sim \mathcal{G}_{aux}$, and $\text{MSE}(f_{ext}(p), p)$ is the MSE between the original posteriors p and the posteriors reconstructed by the feature extractor. This ensures the latent representations preserve fine-grained information in the posterior space.

3.2 Classification

The attack leverages reconstruction error as a proxy for membership. The hypothesis is that member nodes are more difficult to generalize and thus yield higher reconstruction loss. For each node $u \in \mathcal{G}_{tar}$, the reconstruction error is: $\text{error}(u) = \|f_{ext}(p_u) - p_u\|_2^2$.

We define a threshold as the n-th percentile of errors across all test nodes: threshold = Percentile($\{\text{error}(u)\}_{u\in\mathcal{G}_{tar}}, n$). The membership decision is made as:

$$y_{mia}(u) = \begin{cases} 1, & \text{if error}(u) > \text{threshold} \\ 0, & \text{otherwise} \end{cases} \tag{6}$$

The attack is particularly valuable when the adversary wishes to avoid reliance on external classifiers. Reconstruction loss provides a smooth, data-driven indicator of deviation from typical student behavior.

4 Increasing MIA Complexity

Earlier, we assumed a shared training dataset for both teacher and student models, a setting that enables the student to closely match the teacher's posteriors and potentially leak private data, even though the adversary lacks direct access to the dataset. In many privacy-preserving ML workflows, however, the student is trained on a separate, publicly available dataset [19]—limiting direct leakage but raising new inference challenges. Here, we query the student to compute confidence scores, requiring indirect inference of the teacher's training data, which adds complexity to the MIA setting by eliminating leakage from dataset overlap.

While metrics such as AUC and accuracy indicate general attack effectiveness [14, 47–49], they fail to reflect adversaries' ability to identify sensitive individual samples, especially under low false positive (FP) constraints. For example, identifying critical vulnerabilities requires reliably compromising sensitive data subsets with high success rates and low FPR. This focus aligns with practices in sensitive applications where minimizing FP is essential for identifying the most critical vulnerabilities under stringent scenarios. Accuracy and AUC are not directly correlated with FPR, highlighting their limitations in evaluating MIAs [7].

LiRA [7] highlights the importance of evaluating MIA effectiveness under low false positive rates (FPRs) but relies on computationally expensive shadow models. To provide a scalable alternative, we adopt a quantile regression (QR)-based method [3] and extend it as a privacy risk analysis tool for GNN-to-MLP knowledge distillation. Specifically, we evaluate whether posteriors from the student model—trained without access to the teacher's private data or the graph structure—can still leak membership information about the teacher's training set. To compensate for the signal attenuation caused by knowledge distillation, we use an auxiliary model f_{att} to amplify separability between member and non-member samples.

4.1 MIA via Quantile Regression (✔, ✔, ✔, ✘)

Attack Overview. The MIA targets f_{tar}, which is trained on $\mathcal{G}_{priv}$. The attack aims to determine whether a node-label pair (u, y) belongs to $\mathcal{G}_{priv}$, formulated as a hypothesis test:

$$H_0 : (u, y) \sim \mathcal{D} \setminus \mathcal{G}_{\text{priv}} \quad \text{vs.} \quad H_1 : (u, y) \in \mathcal{G}_{\text{priv}} \tag{7}$$

where $\mathcal{D}$ is the unknown underlying data distribution over graph-structured examples from which $\mathcal{G}_{\text{priv}}$ is drawn. Under H_0, (u, y) is sampled from $\mathcal{D}$ but not included in the training graph. Under H_1, the sample is included in training. Inspired by Bertran et al. [3], we apply a threshold to the test statistic $s(u, y)$ to classify whether (u, y) is part of $\mathcal{G}_{priv}$.

Attack Process. The attack utilizes QR model q to classify node-label pairs based on their confidence scores:

$$\mathcal{A}_q(u, y) = \begin{cases} 0 & (u \sim \mathcal{G}) \quad \text{if } s(u, y) < q(u), \\ 1 & (u \sim \mathcal{G}_{priv}) \quad \text{if } s(u, y) \geq q(u). \end{cases} \tag{8}$$

where $q(u)$ is a model q trained on samples $(u, y) \sim \mathcal{G}$, ensuring no overlap with $\mathcal{G}priv$. The adversary, operating with API access to the student model, queries the student to obtain confidence scores and infers membership in $\mathcal{G}_{priv}$ based on these scores.

Challenges with querying the student model. Distilling knowledge from a GNN to an MLP tends to lead to the loss of high-frequency spectral information crucial for distinguishing node features from neighbors [40]. While the MLP may preserve neighborhood smoothing, where node features within the same neighborhood become similar, the inability to differentiate subtle spectral node characteristics reduces the effectiveness of quantile-based attacks directly applied to the student model. Also, teacher-student model mismatch further reduces inference accuracy.

Leveraging an Auxiliary Attack Model f_{att}. To mitigate these challenges, we introduce an attack model, f_{att} trained on an auxiliary graph $\mathcal{G}_{aux'} \sim \mathcal{G}$, disjoint from $\mathcal{G}_{priv}$ but sampled from the same distribution. While f_{att} lacks teacher supervision, it serves as a reference to identify KD-induced biases in f_{st}. While f_{att} does not capture the structural knowledge embedded in the teacher GNN, it is a valuable reference in scenarios where the attacker lacks access to the training data for the teacher or student models. By comparing KD-trained f_{st} with f_{att}, *we reveal patterns that persist from the teacher's training data within the student model.* This comparison relies on confidence scores using the hinge scoring function [7]:

$$s'(u, y) = z_y(x) - \max_{y' \neq y} z_{y'}(x) \tag{9}$$

here $z_y(x)$ is the true class logit and $\max_{y'\neq y} z_{y'}(x)$ is maximum logit for other class. This scoring rule is computationally simpler than the logit-scaled confidence score:

$$s'(u, y) = \log(f_y(u)) - \log(1 - f_y(u)) \tag{10}$$

where $f_y(u)$ is the model's confidence. To amplify differences between f_{st} and f_{att}, we compute:

$$s(u, y) = s'_{f_{st}}(u, y) - s'_{f_{att}}(u, y) \tag{11}$$

$s(u, y)$ leverages subtle confidence patterns retained by f_{st} from its teacher's training data. Despite KD-induced information loss, comparing f_{st} to the independently trained f_{att} enables effective distinction between members and non-members of $\mathcal{G}_{priv}$. This approach highlights vulnerabilities in KD-trained models and offers insights into mitigating risks associated with information leakage. By analyzing relative confidence scores, attackers can infer membership in private datasets, highlighting that KD, despite its compression benefits, can inadvertently preserve identifiable signals—underscoring the importance of privacy-aware student training.

4.2 Balancing Performance and Finite-Sample Guarantees (✔, ✔, ✔, ✘)

QR offers a means to predict a specified FPR and provides asymptotically consistent estimates of conditional quantile functions [28, 35].

However, its reliance on large-sample behavior can limit its effectiveness when finite-sample guarantees are required for valid coverage at a given FPR. To address this, we integrate conformal prediction for non-asymptotic, distribution-free coverage [29].

This section proposes a hybrid approach that combines the predictive efficiency of QR with the reliability of conformal prediction. By integrating these methods, we enable adversaries to leverage QR's strengths while maintaining statistically sound, distribution-agnostic guarantees, ensuring robust MIA performance across varying sample sizes.

Conformal Prediction. We employ split conformal prediction, which calculates non-conformity scores using a separate calibration set to achieve valid coverage for a specified FPR $\alpha \in [0, 1]$ [26, 37]. These scores quantify the deviation between observed and predicted values and are defined as $R_i = |y_i - \hat{y}_i|$ for $i \in C$, where C is the calibration set $\{(x_i, y_i) : i \in C\}$. The $(1 - \alpha)$ quantile of the empirical distribution of these scores, denoted as ζ, adjusts for finite calibration set size $|C|$ that ensures valid finite-sample coverage:

$$\zeta = \text{Quantile}_{(1-\alpha)\left(1+\frac{1}{|C|}\right)}\left(\{R_i : i \in C\}\right). \tag{12}$$

Given a test point x_{n+1}, we define a prediction interval $\mathcal{I}(x_{n+1})$ associated with the predicted value $\hat{y}_{n+1}$ as:

$$\mathcal{I}(x_{n+1}) = [\hat{y}_{n+1} \pm \zeta] \tag{13}$$

This interval accounts for the uncertainty in the prediction and guarantees valid coverage at the specified FPR α.

Attack Overview. We employ CQR [29] to control FPR while adapting to different regression models. By integrating CQR with an underlying QR model, the adversary achieves valid coverage independent of model performance. The attack partitions data into disjoint training and calibration sets. The training set estimates the $(1 - \alpha)$ quantile of confidence scores $s(u, y)$, and the calibration set adjusts this estimate to meet the target coverage. This two-step process ensures robust inference with formal statistical guarantees.

Attack Process. We begin by randomly splitting the dataset $(u, y) \sim \mathcal{G}$ into a training set $\mathcal{T}$ and a calibration set C, ensuring no overlap. We then train a QR model q on the samples in $\mathcal{T}$ to estimate the $(1 - \alpha)$-quantiles of the scores $s(u, y)$ for each $u \in \mathcal{T}$. For each calibration sample $(u, y) \in C$, we calculate the non-conformity score R_u as the absolute difference between the observed score $s(u, y)$ and the estimated score $q(u)$:

$$R_u = |s(u, y) - q(u)| \tag{14}$$

Following this, we compute ζ_u from the empirical distribution of non-conformity scores in C, as described in Eq. 12. This step calibrates the QR predictions to ensure reliable finite-sample coverage. We can formally define the MIA as a decision function $\mathcal{A}_q : \mathcal{U} \times \mathcal{Y} \to \{0, 1\}$, where:

$$\mathcal{A}_q(u, y) = \begin{cases} 0 \quad (u \sim \mathcal{G}) & \text{if } s(u, y) < q(u) + \zeta_u, \\ 1 \quad (u \sim \mathcal{G}_{priv}) & \text{if } s(u, y) \geq q(u) + \zeta_u. \end{cases} \tag{15}$$

where, $\mathcal{U}$ represents the set of all possible data samples and $\mathcal{Y}$ is the set of possible labels. If $\mathcal{A}_q(u, y) = 0$, the sample is inferred to belong to the public dataset $\mathcal{G}$; if $\mathcal{A}_q(u, y) = 1$, it is inferred to belong to the private dataset $\mathcal{G}_{priv}$.

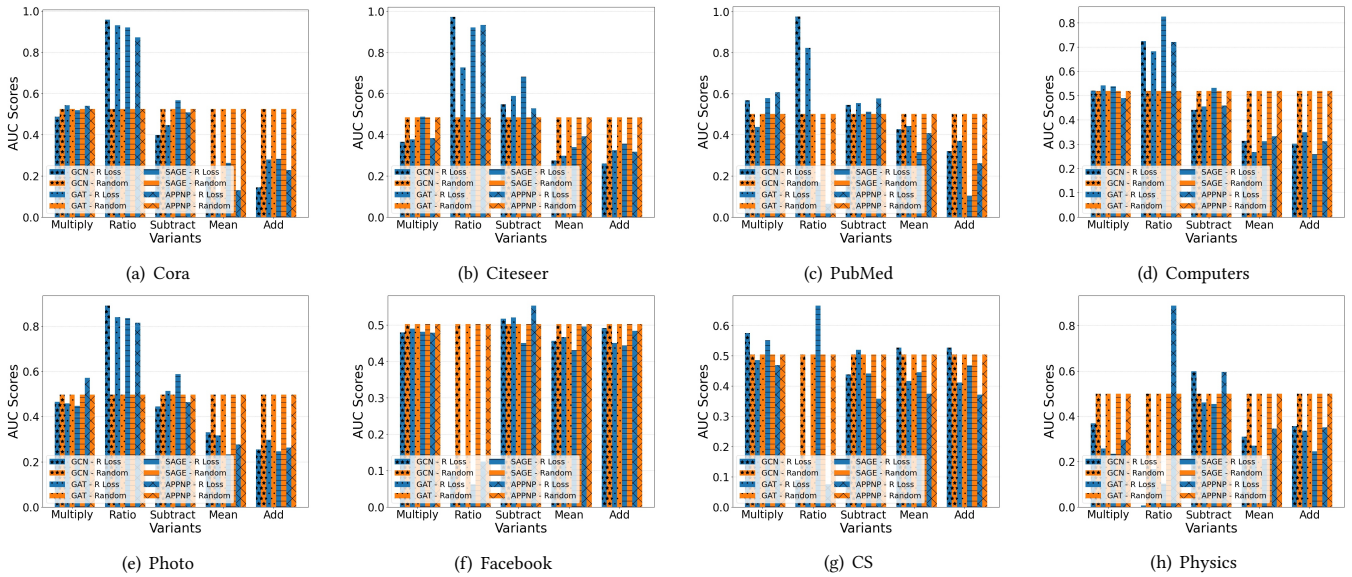
5 Beyond MIA

Although MIA provides the primary lens through which we quantify privacy leakage in knowledge-distilled models, limiting analysis to membership alone risks overlooking broader forms of information transfer. In this section, we extend our framework with two diagnostic analyses –*property inference* and *link theft* [14, 47]—that reveal how privacy leakage propagates beyond individual samples. These extensions serve as analytical probes to test whether distilled student models retain semantic or relational information inherited from the teacher. By demonstrating that student posteriors can still be exploited to infer global graph properties or reconstruct sensitive links, we show that knowledge distillation not only leaks membership information but also transfers higher-order structural and attribute-level dependencies, underscoring broader privacy risks in KD frameworks.

Property Inference (✔, ✘, ✔, ✔). In this attack, the adversary aims to infer node properties, such as degrees, from student posteriors. Unlike prior attacks [47], which rely on graph embeddings, our attack uses a black-box setting where only student posteriors are accessible. We focus on node degree—a critical property often linked to influence or centrality. Identifying high-degree nodes can enable targeted attacks (e.g., compromising key organizational users). We train a classifier to predict discretized node degrees based on posteriors obtained from the student. Specifically, posteriors extracted from an auxiliary dataset $u_{aux} \in G_{aux}$ queried from the student are used as input to train the classifier, with labels based on node degrees. The outputs are discretized into k bins for classification [47]. The classifier maps posteriors to degree bins by training on auxiliary data, enabling inference of critical node properties in constrained settings. Details of the classifier is given in § 6.5.

Link Stealing (✔, ✘, ✘, ✔). This attack aims to infer edge presence between node pairs in the target GNN's training graph using only student outputs. While original methods assume access to the target model's posteriors [2, 14], we investigate whether such attacks remain effective when only the student model's posteriors are available. Our setup assumes the adversary: (i) has no knowledge of node attributes or partial graph structure (unknown graph attributes), and (ii) no auxiliary dataset. We use posteriors from the student model f_{st} to compute distances between node pairs (u, v) using eight metrics: Cosine, Euclidean, Correlation, Chebyshev, Bray-Curtis, Canberra, Manhattan, and Square-Euclidean. These metrics are selected for their ability to capture diverse relational patterns in high-dimensional spaces (details in Table 1, Appendix D). Next, we apply K-means clustering to categorize node pairs into two groups: likely connected (pairs with smaller distances) and likely disconnected (pairs with larger distances).

This clustering-based approach allows the adversary to estimate the edge structure of the underlying graph, without direct access to the teacher model. The success of this attack depends on the alignment between the student and teacher models, which we evaluate in subsequent experiments.



(a) Cora (b) Citeseer (c) PubMed (d) Computers

(e) Photo (f) Facebook (g) CS (h) Physics

Figure 3: Performance of auto MIA across datasets and GNN models. The plots compare attack variants of different combination methods across eight datasets. Each bar represents a unique combination of model and posterior fusion strategy (KMeans, Random, or R Loss).

6 Evaluation

6.1 Experimental Setup

Research Questions. Our evaluation is guided by the following privacy-centered research questions:

- **RQ-1: Do students distilled from GNNs leak teacher membership signals?** We test whether structure-agnostic students preserve exploitable membership signals, even in black-box settings without access to the teacher or graph.
- **RQ-2: Does leakage persist under strict privacy constraints and cross-dataset KD?** We evaluate whether membership remains detectable when students are trained on disjoint datasets and audited under low-FPR thresholds, using QR/CQR to probe high-confidence leakage despite structural mismatch.
- **RQ-3: Under what conditions is leakage most severe?** We examine how leakage varies with datasets, GNN architectures, and KD parameters (e.g., temperature, supervision weight), and identify which private nodes are most at risk.
- **RQ-4: Does KD expose private graph properties beyond membership?** We test whether students reveal semantic or structural cues from the teacher's training graph, such as node degree or edge connectivity.

Models. We evaluate four representative GNNs: GCN [20], GAT [36], GraphSAGE [13], and APPNP [11]. These models represent different architectures and diverse message-passing mechanisms for node classification. We follow Zhang et al.[46] for model architectures and configurations. Hyperparameters and prediction accuracies are reported in Appendix F (Tables 2, 3).

Datasets. Our evaluation spans eight benchmarks across citation, co-purchase, and social networks: Cora, Citeseer, PubMed [24, 32], A-Computers, A-Photo, CS, Physics [33], and Facebook [30]. These datasets vary in size and topology, enabling assessment of attack generalizability [18, 22]. Dataset statistics are in Appendix G.

KD Framework. We adopt the standard KD objective [16] and its GNN→MLP adaptation [46]. For node u, the student prediction $\hat{y}_u$ is trained on both true labels y_u and teacher outputs z_u:

$$\mathcal{L} = \alpha \sum_{u \in \mathcal{G}} \mathcal{L}_{local}(\hat{y}_u, y_u) + (1-\alpha) \sum_{u \in \mathcal{G}} \mathcal{L}_{KD}(\hat{y}_u, z_u) \tag{16}$$

where $\mathcal{L}_{local}$ is the cross-entropy loss between the student predictions $\hat{y}_u$ and the true labels y_u, and $\mathcal{L}_{KD}$ is the Kullback–Leibler (KL) divergence loss between $\hat{y}_u$ and the teacher output z_u. The parameter α balances the contributions of the local supervision and KD losses, and is tuned to optimize student performance.

Implementation. All experiments use PyTorch Geometric with Python 3.11.4 on an NVIDIA RTX A6000 (48GB VRAM). Training settings are in Appendix F.

Data Partitioning. Each dataset $\mathcal{G}$ is split 50/50 into disjoint target ($\mathcal{G}tar$) and auxiliary ($\mathcal{G}aux$) graphs, each further divided into train/val/test (70/15/15). This setup supports systematic evaluation of KD and attack susceptibility.

6.2 Membership Inference Attack

To address **RQ-1**, we evaluate the performance of reconstruction-based membership inference attacks as described in Section 3.

Evaluation Metrics. We evaluate MIA effectiveness using AUC, the standard measure for binary classification across all thresholds [6, 14, 45]. Results are reported as mean AUC. Unless stated otherwise, the KD balancing coefficient is fixed at $\alpha = 0.5$.

Baseline: Random Guessing. We include a random guessing attack, where the adversary predicts membership uniformly at

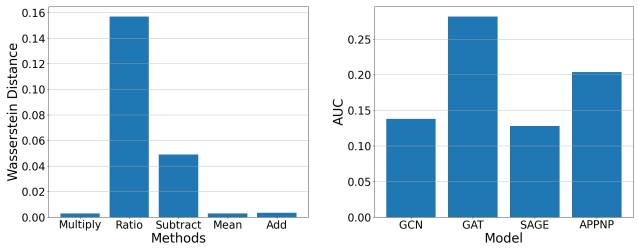
random. This baseline benchmarks attack performance against chance.

Reconstruction Loss Exposes Structured Leakage. The reconstruction attack achieves remarkably high AUC scores across several datasets, attaining 0.96, 0.97, and 0.97 for GCN on Cora, Citeseer, and PubMed, respectively. Notably, the APPNP model on Physics also yields a strong AUC of 0.93, while the SAGE model on CS reaches 0.67. These results indicate that student representations retain latent signals sufficient to reliably differentiate members from non-members. This separability is further enhanced by ratio-based posterior fusion, which magnifies contrastive patterns between membership classes.

To better quantify this separation, we measure the *Wasserstein distance* between the reconstruction loss distributions of member and non-member instances (Fig. 4(a)). Among all posterior fusion strategies, ratio-based fusion produces the *largest separation* in reconstruction loss distributions, suggesting that it accentuates membership-related differences in the input space that the autoencoder encodes into distinct latent representations. As shown in Fig. 4(a), ratio fusion yields the highest Wasserstein separation because multiplicative normalization amplifies teacher-inherited overconfidence signals while preserving relative entropy, whereas additive fusion methods tend to smooth these signals. This amplification enhances the contrast between member and non-member posteriors, leading to greater separability in the latent space and, consequently, higher reconstruction fidelity. The results support the view that knowledge distillation leaves measurable reconstruction fingerprints in the student model—fingerprints that, when magnified through ratio-based fusion and captured by appropriately tuned encoders, can be effectively exploited for membership inference.

Dataset Characteristics and Student Generalization Govern Attack Effectiveness. While the attack performs well on many benchmarks, it yields lower AUC scores on datasets such as Facebook and Physics. However, average-case metrics like AUC may obscure the true nature of vulnerability. An attack might still confidently identify a small subset of member samples even when its AUC is low—particularly under tightly controlled FPRs, which are appropriate in rigorous model audits, where even a few successful inferences may signal significant privacy leakage. Conversely, a high AUC does not guarantee that any single prediction can be made with high confidence. This gap between average-case performance and worst-case vulnerability is critical in security-sensitive settings, where adversaries often prioritize low-FPR identification of high-risk samples over broad membership trends.

Notably, these inconsistencies across datasets can also trace back to differences in how well the student replicates the teacher's decision boundary. For example, on the Facebook dataset, the student model underperforms the teacher by nearly 10%, suggesting incomplete knowledge transfer and a weaker embedding of member-specific artifacts. In contrast, datasets like Cora and PubMed exhibit stronger teacher-student alignment and clearer separation in reconstruction loss. Even when student accuracy is high, as in CS and Physics, the complexity of the data and the regularization effects of distillation may suppress sensitivity to individual training points, thereby reducing the extractable membership signal. These patterns highlight how both dataset geometry and post-distillation generalization behavior jointly modulate the attack surface.



(a) Wasserstein distance (b) No attack model

Figure 4: (a) Wasserstein distance between members and non-members for each combination method. (b) Attack AUCs without f_{att} for reconstruction-loss-based attacks.

Weak-case analysis. We further investigate the weak attack cases using Fisher ratio and nearest-centroid accuracy, which jointly capture global and local structure in the learned feature space. A large Fisher ratio indicates that inter-class variance dominates, implying well-separated class centroids relative to their internal spread. Conversely, a small ratio reflects overlapping clusters or large intra-class variance that obscures decision boundaries. Intuitively, a high Fisher ratio corresponds to strong global class discrimination, whereas a low value signifies entangled embeddings. In addition, we compute the nearest-centroid validation accuracy—a local separability measure that evaluates how tightly samples cluster around their true class centroids. For each class, we compute a centroid from a subset of training nodes and assign each held-out node to its nearest centroid in Euclidean space. The validation accuracy is

$$\text{ValAcc} = \frac{\#\,\text{correctly assigned nodes}}{\text{total validation nodes}}.$$

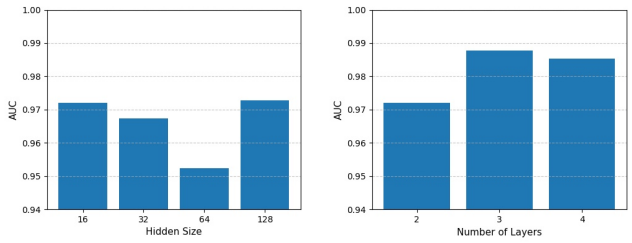
High centroid accuracy reflects compact, well-aligned clusters; low accuracy suggests diffuse or overlapping manifolds.

As shown in Appendix H, Cora and Citeseer exhibit relatively similar Fisher ratios but only moderate centroid accuracy. This indicates that although class centroids are similarly separated, internal scatter remains large, producing diffuse manifolds that rely heavily on structural cues for discrimination. In contrast, Facebook, CS, and Physics display substantially higher centroid accuracy, revealing compact and internally coherent clusters that lie close together globally yet remain locally consistent. In these graphs, the student can approximate the teacher's local decision boundaries more effectively, leading to smaller membership gaps. Structure-dependent datasets such as Cora and Citeseer, however, amplify confidence biases transferred during KD, making them more susceptible. Overall, the analysis shows that attack strength depends not only on graph topology but also on the feature–structure balance governing how membership information is geometrically encoded and transferred between teacher and student models.

Impact of f_{att}. Removing the auxiliary model f_{att} leads to a drastic drop in performance. As shown in Fig. 4(b), the resulting AUC scores on the Cora dataset drop sharply across all GNN architectures. For example, on Cora, the GCN model's AUC drops from 0.96 to

0.13 when f_{att} is omitted. This suggests that f_{att} plays a key role in magnifying posterior discrepancies and shaping them into a space where member and non-member points become separable. Rather than acting as a passive feature extractor, f_{att} functions as a signal amplifier, learning transformations that surface otherwise subtle KD-induced artifacts. This highlights that the attack's success hinges not merely on raw posterior shifts, but on how well the latent geometry of the learned space can be exploited for inference.

The variability in attack performance reflects deeper interactions between dataset complexity, student generalization behavior, and how KD encodes or suppresses sample-specific information. Our results offer an initial lens into the vulnerability of GNNs through the structure-agnostic student model. In the subsequent analysis, we demonstrate that attacks with modest AUC can still expose vulnerable members with high precision—comparable to those with stronger average-case performance.



(a) With hidden sizes (b) With number of layers

Figure 5: Effect of model architecture on attack success for the GCN–Citeseer setup. (a) Attack success with different hidden sizes. (b) Attack success with increasing number of layers.

Effect of architectural disparity. To probe sensitivity to structural mismatch, we perform experiments on the GCN/Citeseer setup that vary hidden sizes (16–128) and introduce layer-count mismatches (2–4). The membership-inference AUC remained high (0.95–0.99), indicating the attack tolerates moderate architectural uncertainty and can succeed even when the adversary lacks precise architectural details of the student model. Our results are shown in Fig. 5.

6.3 Increasing MIA Complexity

To address **RQ-2**, we test whether membership leakage persists when student models are trained under more stringent KD conditions. We focus on the *cross-dataset* setting, where the student is trained on a dataset disjoint from the teacher's training set, and evaluate attacks under strict FPR constraints. This scenario mirrors real-world deployments such as federated learning, cross-institutional collaboration, and public-to-private knowledge transfer, where attackers lack access to training data and internal teacher parameters [1, 17].

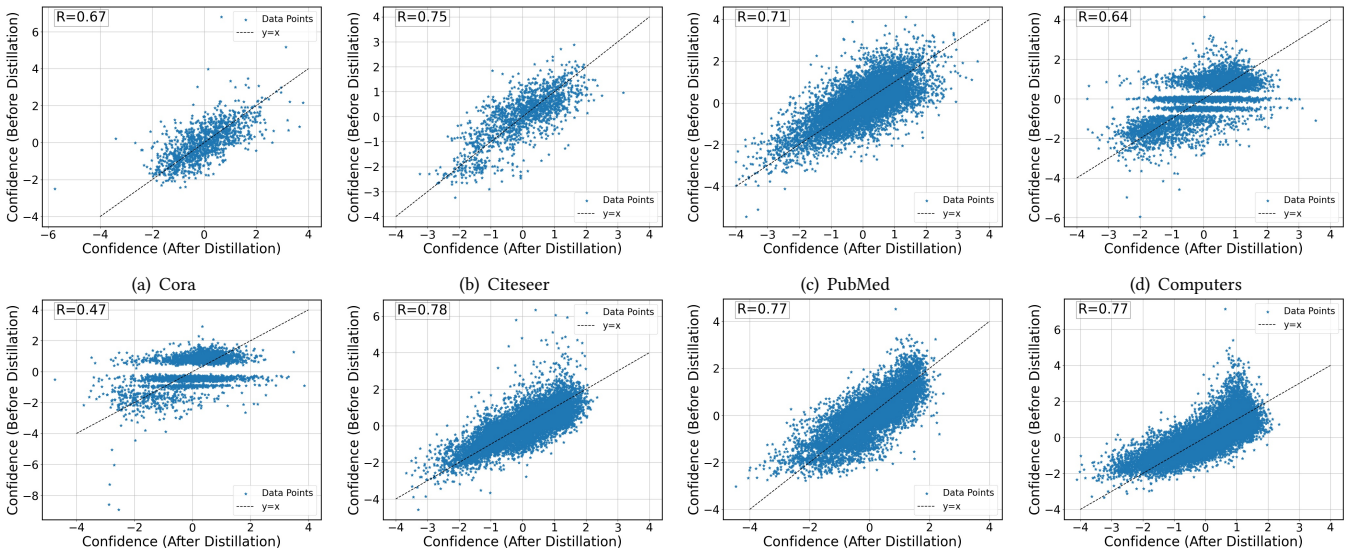
Attack Setup. We adopt the hinge loss scoring function [3, 7] for its stability and correlation with model confidence. MIA is performed using QR and CQR (Section 4). Each trains a parameterized model to predict $\mu(u)$ and $\log \sigma(u)$ for data point u, defining a Gaussian $\mathcal{N}(\mu(u), \sigma(u)^2)$. Membership is decided via quantile thresholds, with results reported at fixed FPRs of 0.01%, 0.5%, and 1%. False positives are evaluated on a held-out set disjoint from teacher, student, and attack training.

Is Distillation Sufficient for Privacy in GNNs? To assess whether KD sufficiently obfuscates membership-related signals, we compare hinge-based confidence scores before and after distillation using GCN models. As shown in Fig. 6, we observe moderate to strong correlations between pre- and post-distillation scores. This suggests that membership-relevant signals in the teacher's logits remain partially preserved in the student's outputs, posing potential privacy risks. The clustering of points near the diagonal line $y = x$ underscores the limited perturbation introduced by KD. However, datasets like A-Photo show weaker correlations and greater variance in post-distillation scores, suggesting increased obfuscation and potentially reduced attack success.

QR Results. We first evaluate QR using the auxiliary attacker model f_{att} trained on student logits. As shown in Table 5 (Appendix J), QR achieves high precision even at low FPR thresholds. For example, on the Facebook dataset—traditionally difficult for MIA—the attack achieves 89% precision at 1% FPR. On datasets with clearer signal separation, such as Citeseer, QR reaches 90.14% precision at the same FPR. QR's conservative design ensures that predictions are made only when confidence is high, yielding sparse but reliable inferences. These results demonstrate that KD does not fully remove membership-dependent structure from student logits, enabling high-confidence attacks.

CQR Results. To improve statistical robustness, we evaluate CQR, which augments QR with calibration based on held-out residuals. This correction guarantees valid finite-sample coverage in regions of posterior uncertainty or sparse calibration support. The ability to provide such guarantees is especially important in privacy auditing, where overstated or uncalibrated confidence could mask rare but severe leakage. As shown in Table 6 (Appendix J), CQR achieves consistently high precision across datasets and architectures. For instance, the GCN model on Cora reaches 89.04% precision at 0.5% FPR, with comparable results for GAT and SAGE. These results confirm that the membership signal survives not only across dataset shifts and architectural gaps, but also under stricter statistical constraints.

Implications. Our results show that membership inference remains effective even when students are trained on datasets disjoint from the teacher's training set and evaluated under strict false-positive constraints. Both QR and CQR reliably flag vulnerable samples, demonstrating that KD systematically encodes membership information in the student's output space. Critically, leakage persists even in datasets that appeared resilient under average-case metrics (e.g., low AUC in **RQ-1**), revealing that models deemed "safe" may still expose sensitive instances under targeted attacks. This underscores a central risk: in privacy-critical domains, protecting the average case is insufficient, as adversaries need only a few compromised nodes to cause harm. The persistence of leakage despite architectural mismatches and absence of graph structure highlights the inadequacy of KD as a privacy-preserving mechanism and raises urgent concerns for deployments in high-stakes settings such as healthcare.



(a) Cora (b) Citeseer (c) PubMed (d) Computers

(e) Photo (f) Facebook (g) CS (h) Physics

Figure 6: Scatter plots of confidence scores pre- and post-distillation across datasets. Each subplot corresponds to a specific dataset, showing the correlation (R) between the confidence scores. The $y = x$ line represents the case where confidence remains unchanged after distillation.

6.4 Further Investigation of KD Leakage

To address **RQ-3**, we diagnose when and why membership inference succeeds by examining architectural settings, distillation regimes, and instance characteristics that heighten privacy risk. We analyze the roles of predictive confidence, graph structure, temperature scaling, and distillation strength in shaping attack success. These insights clarify the mechanics of leakage and highlight vulnerable regions where stronger protections are needed.

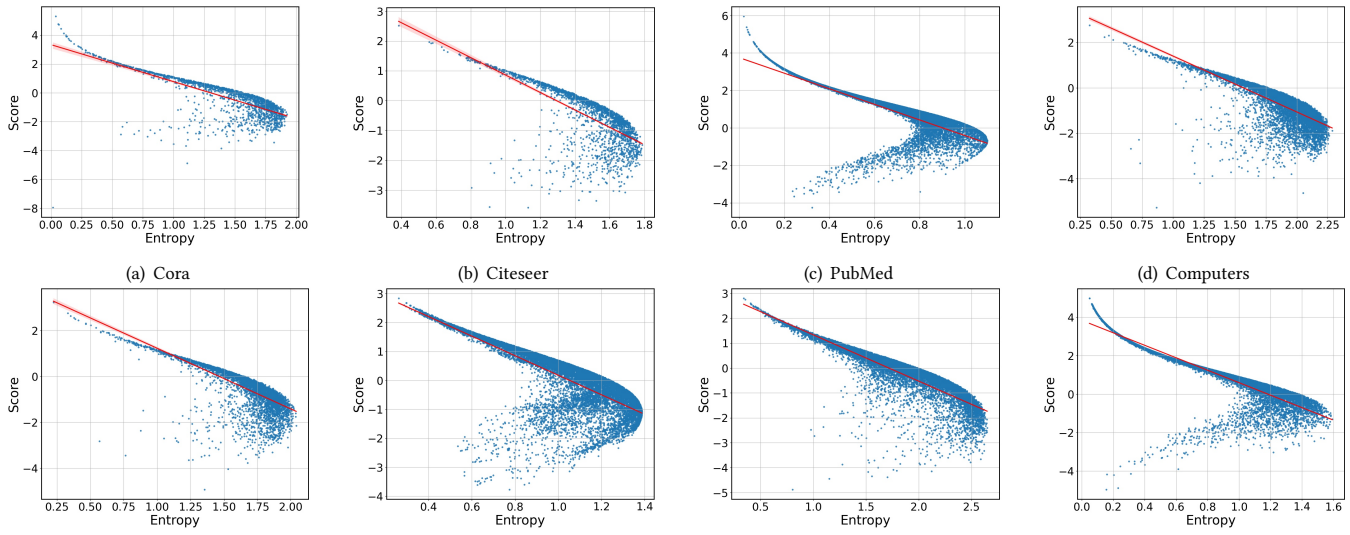
Confidence and Entropy Shape Vulnerability. Fig. 7 illustrates the relationship between predictive entropy and attack scores (via QR). Across datasets, we observe a strong inverse correlation: low-entropy (high-confidence) predictions yield higher membership scores. This reflects QR's design of modeling upper quantiles, making attacks most decisive where students confidently mimic teacher logits. In such regions, the student preserves teacher-induced distinctions between members and non-members, which QR exploits for precise inference. By contrast, high-entropy predictions—associated with uncertain samples—blur the boundary and weaken attack precision. CQR reinforces this analysis through calibrated prediction intervals. As shown in Fig. 8(a), entropy and membership scores remain inversely related, but CQR widens intervals in high-entropy regions, providing conservative calibration where uncertainty is high. This curbs overconfident membership decisions and improves robustness against miscalibration.

Node Degree Does Not Predict Vulnerability. We test whether node degree—a proxy for topological centrality—correlates with node membership vulnerability. As shown in Fig. 8(b), membership scores are diffusely distributed, with both low- and high-degree nodes exhibiting wide score ranges and no clear trend. Unlike teacher GNNs, which exploit connectivity, the student MLP lacks graph structure at inference. Leakage stems from feature–label alignment distilled from the teacher rather than node topology. This decoupling shows that vulnerability persists even when structural cues are absent, underscoring that privacy risks in student models arise from how feature–label patterns are encoded and remain exploitable in graph-free inference settings.

Effect of Temperature Scaling. Temperature scaling smooths or sharpens teacher logits in KD, but its effect on leakage is architecture-dependent. Fig. 9 shows ROC curves at FPR= 0.01% for GCN, GraphSAGE, GAT, and APPNP on CS under varying temperatures. **Low temperatures (0.1–0.5)** amplify leakage in GCN by producing overconfident logits, but reduce vulnerability in GAT and APPNP by suppressing distinctions. **Moderate temperatures (1–2)** yield the strongest attacks across most models, balancing expressiveness and smoothness; GAT and APPNP peak in this range. **High temperatures (4)** reduce leakage in GCN and GAT as logits flatten, yet GraphSAGE and APPNP sustain high AUCs, showing resilience to smoothing. The results show that temperature scaling does not uniformly improve privacy: settings that optimize utility may heighten risk. Adaptive or graph-aware temperature tuning may be needed to balance utility and confidentiality in KD pipelines.

Effect of Distillation Strength. We evaluate how the distillation weight α in Eq. 16 influences membership risk on the CS dataset using CQR across four GNNs, varying α from 0 (no KD) to 1 (full KD). Results are in Fig. 10.

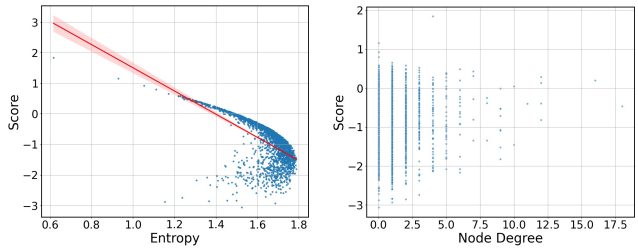
- **Partial distillation** (α = 0.5–0.7) often maximizes leakage. For instance, APPNP peaks in AUC and TPR at this region, as combining teacher supervision with labels preserves strong membership cues.



(a) Cora (b) Citeseer (c) PubMed (d) Computers

(e) Photo (f) Facebook (g) CS (h) Physics

Figure 7: Scatter plot of model confidence scores against predictive entropy for individual data points across each dataset using QR. The red line indicates the fitted linear regression trend.



(a) Confidence vs. Entropy (CQR) (b) Confidence vs. Node Degree

Figure 8: Confidence score analysis on the Citeseer dataset. (a) Confidence vs. entropy using the CQR method, showing the relationship between prediction confidence and uncertainty. (b) Confidence vs. node degree, illustrating how node connectivity correlates with model certainty.

- **Full distillation** ($\alpha = 1.0$) yields mixed outcomes: GAT peaks at this setting, while others show reduced leakage, likely from diminished student-specific generalization.
- **No distillation** ($\alpha = 0$) generally minimizes leakage, as students rely solely on labels without teacher signals.

Summary. Leakage depends not only on architecture but on how α governs student–teacher alignment. Its interaction with feature–label correlations, model capacity, and dataset complexity shapes vulnerability. Privacy auditing must therefore move beyond global metrics to local behaviors tied to calibration, entropy, and training choices. Protecting average-case performance is insufficient, as high-risk samples remain exposed under seemingly benign conditions.

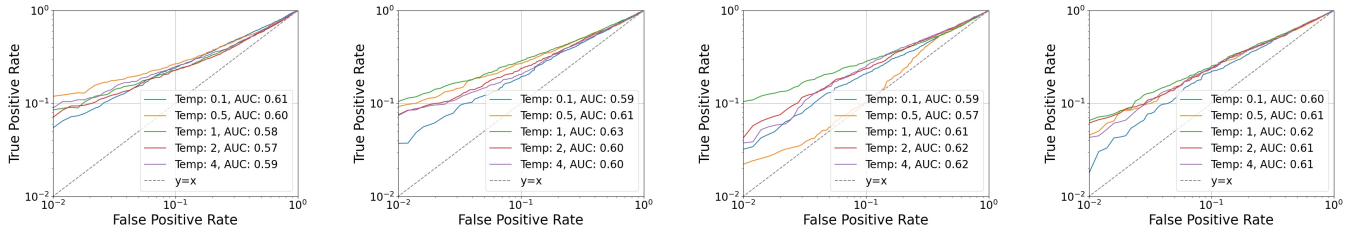
6.5 Beyond MIA: Inference of Private Structural Properties

While MIA identifies whether a node was used in training, real-world risks extend further. Here, we show that KD also leaks semantic and structural properties of the teacher's training graph. Addressing **RQ-4**, we find that student models can act as unintended proxies for inferring sensitive attributes such as node degree and link connectivity—even without graph access or teacher parameters. These results highlight broader privacy risks of KD, especially in graph-free inference settings.

Node Property Inference. We test whether node degree—a structural property of the training graph—can be inferred from student posteriors. Degrees are discretized into three bins (low, medium, high), and classifiers (logistic regression, random forest, gradient boosting, MLP) predict degree labels using only student outputs. Accuracy is the evaluation metric, with random guessing as baseline. We also consider a *nearest-neighbor* variant: each target node is matched to the most similar auxiliary node in posterior space (cosine similarity), and its degree is used as the prediction.

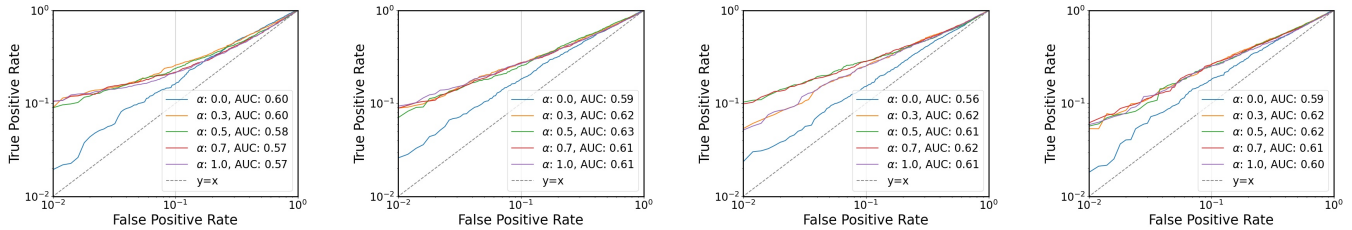
Fig. 11 shows that degrees can be reliably inferred, far surpassing random guessing. On Citeseer, random forest reaches 0.66 accuracy with a GCN teacher, while on A-Photos, Facebook, and Physics accuracies reach 0.96, 0.89, and 0.96, respectively. Even nearest-neighbor performs strongly on some datasets, confirming that structural signals are embedded in posteriors via distillation. Thus, KD transfers graph-derived semantics into students, exposing them to property inference attacks despite lacking topology at inference.

Link Stealing via Posterior Similarity. We test whether KD leaks pairwise structural information—specifically, whether two nodes were connected in the teacher's training graph. The attack infers links by measuring posterior similarity between node pairs, using eight distance metrics (e.g., Cosine, Correlation, Bray–Curtis).



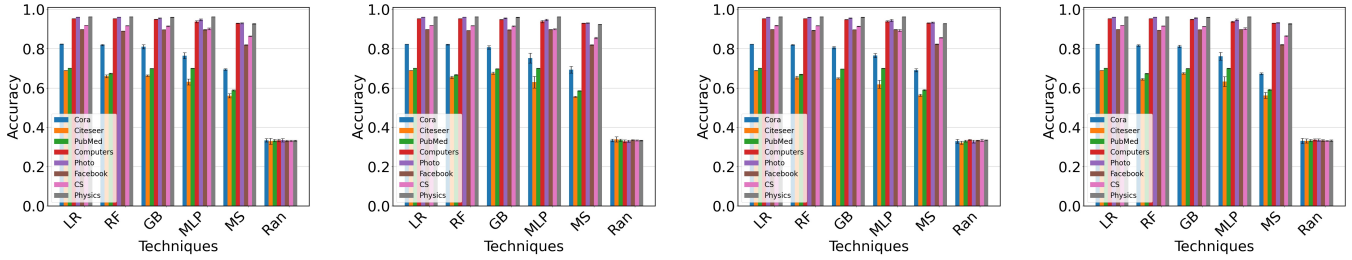
(a) GCN (b) GAT (c) SAGE (d) APPNP

Figure 9: ROC curve of MIA via CQR on the CS dataset, plotted on a log-log scale to highlight performance at low FPRs. The curves compare different temperature scaling values, with the $y = x$ line indicating the baseline where TPR equals FPR.



(a) GCN (b) GAT (c) SAGE (d) APPNP

Figure 10: ROC curve of MIA via CQR on the CS dataset, plotted on a log-log scale to highlight performance at low FPRs. The curves compare different α values for KD, with the $y = x$ line representing the baseline where TPR equals FPR.



(a) GCN (b) GAT (c) SAGE (d) APPNP

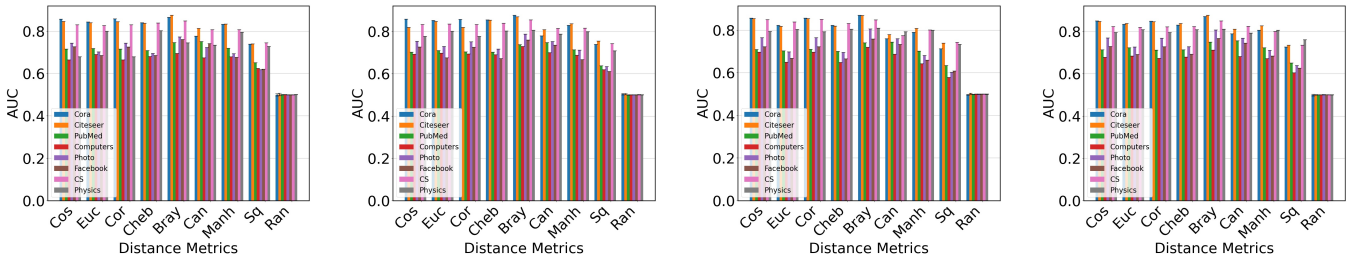
Figure 11: Property inference attacks accuracy on GNN models across eight datasets. The plots compare methods (LR: Logistic Regression, RF: Random Forest, GB: Gradient Boosting, MLP, MS: Most Similar, Ran: Random Baseline). Each subplot represents a distinct GNN architecture.

AUC serves as the evaluation metric. Unlike prior link-stealing attacks [14] that rely on posteriors from the target GNN, our approach uses only student posteriors. Fig. 12 shows strong results: with a GCN teacher on Citeseer, Bray–Curtis achieves AUC 0.875, while with SAGE on CS, Correlation reaches AUC 0.850. These findings indicate that relational inductive biases of the teacher—linked nodes having similar representations—are transferred into the student's output space, even without relational cues at inference.

Implications. Beyond node membership, KD allows leakage of structured information from the teacher's training graph into the student, including connectivity and node-level semantics. This broadens the threat surface, enabling adversaries to reconstruct sensitive properties, such as user roles in social networks, using only student predictions. Crucially, this leakage occurs in a *graph-free* setting, underscoring the inadequacy of KD as a privacy-preserving mechanism. Deployments of student models in public APIs, federated systems, or privacy-critical domains must therefore account for not only membership leakage but the wider spectrum of graph information encoded through distillation.

7 Discussion

In this section, we contextualize our empirical findings and examine their broader implications for privacy in knowledge-distilled GNN–MLP systems. Further details on attack scalability and computational complexity are provided in Appendix L.



(a) GCN (b) GAT (c) SAGE (d) APPNP

Figure 12: Performance of link stealing attack on GNN models across eight datasets using various distance metrics (Cos: Cosine, Euc: Euclidean, Cor: Correlation, CheB: Chebyshev, Bray: Bray-Curtis, Can: Canberra, Mahn: Mahalanobis, Sq: Squared Euclidean, Ran: Random Baseline). Each subplot corresponds to a distinct GNN model architecture.

Interpreting Leakage Patterns. Our evaluation shows that membership leakage persists despite architectural mismatch and dataset disjointness, undermining the assumption that KD inherently provides privacy through abstraction or compression. The student model still encodes fine-grained confidence patterns that correlate with training set membership. Notably, while KD transfers structural information—as demonstrated by the effectiveness of link-stealing attacks—the signal exploited by node-level MIAs is driven more by feature–label alignment than by topological cues. This distinction is important: node-level MIAs succeed by leveraging feature–boundary alignment, whereas link-stealing attacks succeed by exploiting relational biases distilled from the teacher.

Evaluating Input Perturbation. We examine input feature perturbation as a classical defense to reduce overfitting and obfuscate fine-grained memorization. For each node x_u, we compute the perturbed input: $\tilde{x}_u = x_u + \eta, \quad \eta \sim \text{Laplace}(0, b)$, where b controls the noise scale. By injecting Laplace noise into each input vector, the objective is to blur sample-specific information that could manifest in the student's posteriors. Perturbation can reduce the sensitivity of the model to individual training points, particularly when the model's decision surface is locally sharp. However, in KD, the student is trained to approximate the softened posteriors of the teacher, which encode the teacher's confidence patterns and semantic structure rather than purely local fits. As a result, the posteriors may remain smooth but semantically meaningful, and small perturbations to the input may not significantly distort the underlying confidence traces.

Empirically, varying the noise scale yields only minor changes in AUC (Appendix K) while preserving near-identical precision; only large noise substantially reduces attack performance. This highlights a trade-off: small noise fails to shift the latent space, while large noise degrades task accuracy. The student's robustness to mild perturbations may also stem from KD's regularizing effect. Thus, perturbation alone is insufficient to disrupt the higher-order statistical cues driving leakage. Effective defenses likely require complementary strategies that directly reshape posterior distributions without affecting predictive performance, such as entropy regularization, logit masking, or privacy-preserving distillation objectives.

8 Related Work

MIAs. Shokri et al. [34] introduced MIAs, showing that overfitting increases susceptibility. Carlini et al. [7] proposed a likelihood ratio attack, which was later extended by Galichin [10] to include KD, enhancing the attack's effectiveness in a black-box setting. However, these methods are computationally intensive, particularly for large models. Bertran et al. [3] addressed this with a scalable QR-based attack, but its applicability to GNNs remains largely unexplored. Jagielski et al. [19] revealed that student models trained via KD can leak teacher model information, even without direct querying. Building on this, we examine the vulnerability of GNNs in KD setups, revealing that GNNs are equally prone to MIAs.

MIAs on GNNs. GNNs have been shown to be vulnerable to MIAs [25]. Duddu et al. [9] exploited graph embeddings to differentiate training and testing data. Olatunji et al. [25] found that structural information amplifies GNN vulnerability, and He et al. [15] showed successful node-level MIAs with minimal background knowledge. Beyond MIAs, Zhang et al. [47] highlighted the broader privacy risks in GNNs, including property and subgraph inference, and graph reconstruction attacks. Wang and Wang [38] and Wu et al. [42] proposed structure-based and link-stealing attacks, respectively, demonstrating the breadth of privacy concerns in graph data.

In contrast to prior work, we study node-level MIAs in KD settings, asking whether student MLPs distilled from GNNs leak information despite architectural differences. Our results reveal concrete privacy risks of KD in graph learning, with implications for sensitive applications.

9 Conclusion and Future Work

This paper investigates the vulnerability of student models to MIAs in KD frameworks with GNN teachers. We demonstrated that adversaries can exploit student MLP outputs to infer teacher training membership, with attacks effective in same-dataset settings and robust under cross-dataset conditions. Experiments under stricter bounds further confirm attack reliability, underscoring the need for stronger defenses. Our results show that student MLPs leak information about their teacher GNNs despite architectural differences. Future work will extend this analysis to broader tasks such as link prediction and graph classification.

Acknowledgments

We thank our revision editor and the anonymous reviewers for their comments and suggestions. This work has been partially supported by the National Science Foundation (NSF) under grant CNS-2443252. Any findings, conclusions, and recommendations expressed in this paper are those of the authors and do not necessarily reflect the views of the NSF.

References

[1] Kawa Atapour, S Jamal Seyedmohammadi, Jamshid Abouei, Arash Mohammadi, and Konstantinos N Plataniotis. 2024. FedD2S: Personalized data-free federated knowledge distillation. *arXiv preprint arXiv:2402.10846* (2024).

[2] Michael Backes, Mathias Humbert, Jun Pang, and Yang Zhang. 2017. walk2friends: Inferring social links from mobility profiles. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*. 1943–1957.

[3] Martin Bertran, Shuai Tang, Aaron Roth, Michael Kearns, Jamie H Morgenstern, and Steven Z Wu. 2024. Scalable membership inference attacks via quantile regression. *Advances in Neural Information Processing Systems* 36 (2024).

[4] ASA DIX LEGAL BRIEF. 2018. THE HEALTH INSURANCE PORTABILITY AND ACCOUNTABILITY ACT. *JOINT READINESS CENTER LEGAL SECTION* (2018).

[5] Michael M Bronstein, Joan Bruna, Yann LeCun, Arthur Szlam, and Pierre Vandergheynst. 2017. Geometric deep learning: going beyond euclidean data. *IEEE Signal Processing Magazine* 34, 4 (2017), 18–42.

[6] Lei Cai, Jundong Li, Jie Wang, and Shuiwang Ji. 2021. Line graph neural networks for link prediction. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 44, 9 (2021), 5103–5113.

[7] Nicholas Carlini, Steve Chien, Milad Nasr, Shuang Song, Andreas Terzis, and Florian Tramer. 2022. Membership inference attacks from first principles. In *2022 IEEE Symposium on Security and Privacy (SP)*. IEEE, 1897–1914.

[8] Shaveta Dargan, Munish Kumar, Maruthi Rohit Ayyagari, and Gulshan Kumar. 2020. A survey of deep learning and its applications: a new paradigm to machine learning. *Archives of computational methods in engineering* 27 (2020), 1071–1092.

[9] 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.

[10] Andrey V Galichin, Mikhail Pautov, Alexey Zhavoronkin, Oleg Y Rogov, and Ivan Oseledets. 2024. GLiRA: Black-Box Membership Inference Attack via Knowledge Distillation. *arXiv preprint arXiv:2405.07562* (2024).

[11] Johannes Gasteiger, Aleksandar Bojchevski, and Stephan Günnemann. 2018. Predict then propagate: Graph neural networks meet personalized pagerank. *arXiv preprint arXiv:1810.05997* (2018).

[12] Zhichun Guo, William Shiao, Shichang Zhang, Yozen Liu, Nitesh V Chawla, Neil Shah, and Tong Zhao. 2023. Linkless link prediction via relational distillation. In *International Conference on Machine Learning*. PMLR, 12012–12033.

[13] Will Hamilton, Zhitao Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. *Advances in neural information processing systems* 30 (2017).

[14] Xinlei He, Jinyuan Jia, Michael Backes, Neil Zhenqiang Gong, and Yang Zhang. 2021. Stealing links from graph neural networks. In *30th USENIX security symposium (USENIX security 21)*. 2669–2686.

[15] 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).

[16] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531* (2015).

[17] Dong Huang, Xiucai Ye, and Tetsuya Sakurai. 2022. Knowledge distillation-based privacy-preserving data analysis. In *Proceedings of the Conference on Research in Adaptive and Convergent Systems*. 15–20.

[18] Kexin Huang, Ying Jin, Emmanuel Candes, and Jure Leskovec. 2024. Uncertainty quantification over graph with conformalized graph neural networks. *Advances in Neural Information Processing Systems* 36 (2024).

[19] Matthew Jagielski, Milad Nasr, Katherine Lee, Christopher A Choquette-Choo, Nicholas Carlini, and Florian Tramer. 2024. Students parrot their teachers: Membership inference on model distillation. *Advances in Neural Information Processing Systems* 36 (2024).

[20] Thomas N Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907* (2016).

[21] Wanyu Lin, Baochun Li, and Cong Wang. 2022. Towards private learning on decentralized graphs with local differential privacy. *IEEE Transactions on Information Forensics and Security* 17 (2022), 2936–2946.

[22] Sitao Luan, Chenqing Hua, Minkai Xu, Qincheng Lu, Jiaqi Zhu, Xiao-Wen Chang, Jie Fu, Jure Leskovec, and Doina Precup. 2024. When do graph neural networks help with node classification? investigating the homophily principle on node distinguishability. *Advances in Neural Information Processing Systems* 36 (2024).

[23] Federico Mazzone, Leander van den Heuvel, Maximilian Huber, Cristian Verdecchia, Maarten Everts, Florian Hahn, and Andreas Peter. 2022. Repeated knowledge distillation with confidence masking to mitigate membership inference attacks. In *Proceedings of the 15th ACM Workshop on Artificial Intelligence and Security*. 13–24.

[24] Galileo Namata, Ben London, Lise Getoor, Bert Huang, and U Edu. 2012. Query-driven active surveying for collective classification. In *10th international workshop on mining and learning with graphs*, Vol. 8. 1.

[25] 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 (TPS-ISA)*. IEEE, 11–20.

[26] Harris Papadopoulos. 2008. Inductive conformal prediction: Theory and application to neural networks. In *Tools in artificial intelligence*. Citeseer.

[27] PyTorch BigGraph. 2019. PyTorch BigGraph. https://github.com/facebookresearch/PyTorch-BigGraph. Accessed: 2025-01-09.

[28] Zhongjun Qu, Jungmo Yoon, and Pierre Perron. 2024. Inference on conditional quantile processes in partially linear models with applications to the impact of unemployment benefits. *Review of Economics and Statistics* 106, 2 (2024), 521–541.

[29] Yaniv Romano, Evan Patterson, and Emmanuel Candes. 2019. Conformalized quantile regression. *Advances in neural information processing systems* 32 (2019).

[30] Benedek Rozemberczki, Carl Allen, and Rik Sarkar. 2021. Multi-scale attributed node embedding. *Journal of Complex Networks* 9, 2 (2021), cnab014.

[31] Michael Schlichtkrull, Thomas N Kipf, Peter Bloem, Rianne Van Den Berg, Ivan Titov, and Max Welling. 2018. Modeling relational data with graph convolutional networks. In *The Semantic Web: 15th International Conference, ESWC 2018, Heraklion, Crete, Greece, June 3–7, 2018, Proceedings 15*. Springer, 593–607.

[32] 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.

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

[34] 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 (SP)*. IEEE, 3–18.

[35] Ichiro Takeuchi, Quoc V Le, Timothy D Sears, Alexander J Smola, and Chris Williams. 2006. Nonparametric quantile estimation. *Journal of machine learning research* 7, 7 (2006).

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

[37] Vladimir Vovk, Alexander Gammerman, and Glenn Shafer. 2005. *Algorithmic learning in a random world*. Vol. 29. Springer.

[38] Xiuling Wang and Wendy Hui Wang. 2024. Subgraph Structure Membership Inference Attacks against Graph Neural Networks. *Proceedings on Privacy Enhancing Technologies* (2024).

[39] Yiwei Wang, Bryan Hooi, Yozen Liu, and Neil Shah. 2023. Graph Explicit Neural Networks: Explicitly Encoding Graphs for Efficient and Accurate Inference. In *Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining*. 348–356.

[40] Lirong Wu, Haitao Lin, Yufei Huang, Tianyu Fan, and Stan Z Li. 2023. Extracting low-/high-frequency knowledge from graph neural networks and injecting it into mlps: An effective gnn-to-mlp distillation framework. In *Proceedings of the AAAI Conference on Artificial Intelligence*. 10351–10360.

[41] Lirong Wu, Haitao Lin, Yufei Huang, and Stan Z Li. 2023. Quantifying the Knowledge in GNNs for Reliable Distillation into MLPs. *arXiv preprint arXiv:2306.05628* (2023).

[42] Yixin Wu, Xinlei He, Pascal Berrang, Mathias Humbert, Michael Backes, Neil Zhenqiang Gong, and Yang Zhang. 2024. Link Stealing Attacks Against Inductive Graph Neural Networks. *arXiv preprint arXiv:2405.05784* (2024).

[43] Xiaojun Xu, Qingying Hao, Zhuolin Yang, Bo Li, David Liebovitz, Gang Wang, and Carl A Gunter. 2023. How to cover up anomalous accesses to electronic health records. In *32nd USENIX Security Symposium (USENIX Security 23)*. 229–246.

[44] Bencheng Yan, Chaokun Wang, Gaoyang Guo, and Yunkai Lou. 2020. Tinygnn: Learning efficient graph neural networks. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 1848–1856.

[45] Muhan Zhang and Yixin Chen. 2018. Link prediction based on graph neural networks. *Advances in neural information processing systems* 31 (2018).

[46] Shichang Zhang, Yozen Liu, Yizhou Sun, and Neil Shah. 2021. Graph-less neural networks: Teaching old mlps new tricks via distillation. *arXiv preprint arXiv:2110.08727* (2021).

[47] Zhikun Zhang, Min Chen, Michael Backes, Yun Shen, and Yang Zhang. 2022. Inference attacks against graph neural networks. In *31st USENIX Security Symposium (USENIX Security 22)*. 4543–4560.

[48] Zaixi Zhang, Jinyuan Jia, Binghui Wang, and Neil Zhenqiang Gong. 2021. Backdoor attacks to graph neural networks. In *Proceedings of the 26th ACM Symposium*

on Access Control Models and Technologies. 15–26.
[49] Zhiwei Zhang, Minhua Lin, Enyan Dai, and Suhang Wang. 2024. Rethinking graph backdoor attacks: A distribution-preserving perspective. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 4386–4397.
[50] Wenqing Zheng, Edward W Huang, Nikhil Rao, Sumeet Katariya, Zhangyang Wang, and Karthik Subbian. 2021. Cold brew: Distilling graph node representations with incomplete or missing neighborhoods. *arXiv preprint arXiv:2111.04840* (2021).

A Training a Graph Neural Network

Given an attributed graph $\mathcal{G}$, GNNs iteratively update node embeddings by aggregating information from neighboring nodes and edges. For Graph Convolutional Networks (GCNs), the node embeddings are updated at each layer using the following aggregation rule:

$$\mathcal{Z}^{l+1} = \sigma(\tilde{\mathcal{D}}^{-\frac{1}{2}}\tilde{\mathcal{A}}\tilde{\mathcal{D}}^{-\frac{1}{2}}\mathcal{Z}^{l}\mathcal{W}^{l}) \tag{17}$$

where σ is an activation function, $\tilde{\mathcal{A}} = \mathcal{A} + \mathcal{I}$ is the adjacency matrix with added self-loops (identity matrix $\mathcal{I}$), and $\tilde{\mathcal{D}}$ is the degree matrix of $\tilde{\mathcal{A}}$. $\mathcal{W}^l$ is the weight matrix for the l-th layer, and $\mathcal{Z}^k$ is the node representation at the l-th layer. At the input layer ($L = 0$), $\mathcal{Z}^0 = \mathcal{X}$. Normalization with $\tilde{\mathcal{D}}^{-\frac{1}{2}}$ ensures stability during training by scaling the contributions of neighboring nodes.

Beyond GCNs, other GNN variants employ different aggregation strategies. For example, Graph Attention Networks (GATs) use attention coefficients to weigh the importance of neighboring nodes dynamically, enabling the model to prioritize informative neighbors [36]. GraphSAGE employs sampling-based aggregation with diverse functions, such as mean or max pooling, to efficiently process large-scale graphs and enable inductive learning [13]. These variants enhance the expressiveness and scalability of GNNs, making them suitable for complex applications.

B Prediction in Graph Neural Networks

The learned embedding $\mathcal{Z}$ from a GNN can be utilized for various tasks, such as node classification in social network analysis or fraud detection. For node classification, the classifier functions similarly to a standard Multi-Layer Perceptron (MLP). The final representation $\mathcal{Z}^l$ of the model is passed through a Softmax layer to obtain the probability distribution across the C available classes for each node:

$$\tilde{\mathcal{Y}} = \text{Softmax}(\mathcal{Z}^l) \tag{18}$$

here $\tilde{\mathcal{Y}} = {y_i}_{j=1}^{N}$, where y_i denotes the predicted class of the i-th node, and N is the total number of nodes in the graph.

C Motivation: Other Results on the Similarity Between Student and Teacher Probabilities

In this section, we show additional results for the similarity between student and teacher output probabilities. In Fig. 13, we show the scatter plots for the 6 classes of the Cora dataset. Fig. 14 shows the similarity between the output probabilities of the last three classes of Cora using a teacher GCN. Additionally, Fig. 15 illustrates this similarity across all classes for the 100th trial.

Table 1: Distance metrics definition. u and v here represent two vectors, u_i and v_i represent the i-th components of vectors u and v, respsectively.

Metrics	Definition
Cosine	$1 - \frac{u^\top v}{\|u\|_2\|v\|_2}$
Euclidean	$\|u - v\|_2 = \sqrt{\sum_{i=1}^{n}(u_i - v_i)^2}$
Correlation	$1 - \frac{\sum_{i=1}^{n}(u_i-\bar{u})(v_i-\bar{v})}{\sqrt{\sum_{i=1}^{n}(u_i-\bar{u})^2}\sqrt{\sum_{i=1}^{n}(v_i-\bar{v})^2}}$
Chebyshev	$\max_{i=1,\ldots,n}\|u_i - v_i\|$
Bray-Curtis	$\frac{\sum_{i=1}^{n}\|u_i-v_i\|}{\sum_{i=1}^{n}(u_i+v_i)}$
Manhattan	$\|u - v\|_1 = \sum_{i=1}^{n}\|u_i - v_i\|$
Canberra	$\sum_{i=1}^{n}\frac{\|u_i-v_i\|}{\|u_i\|+\|v_i\|}$
Squared Euclidean	$\|u - v\|_2^2 = \sum_{i=1}^{n}(u_i - v_i)^2 = u^\top u + v^\top v - 2u^\top v$

Table 2: Hyperparameters for the GNNs. * indicates that the hyperparameter values are specific for our experiment. All others follow the hyperparameters used in the graph-less neural network paper for KD [46].

Parameter	GCN	GAT	SAGE	APPNP	STUDENT
No. of Layers	2	2	2	2	2
Hidden Dim	128	64	64	64	16*
Learning Rate	0.01	0.01	0.01	0.01	0.01
Weight Decay*	$5e-4$	$5e-4$	$5e-4$	$5e-4$	$5e-4$
Dropout	0	0.8	0.6	0.5	0.5
Attention heads	–	8	–	–	–
Power Iterations	–	–	–	10	–

D Distance Metrics

Table 1 provides the definitions for the distance metrics used in our experiments. These are consistent with the metrics applied in the original link-stealing attack paper [14].

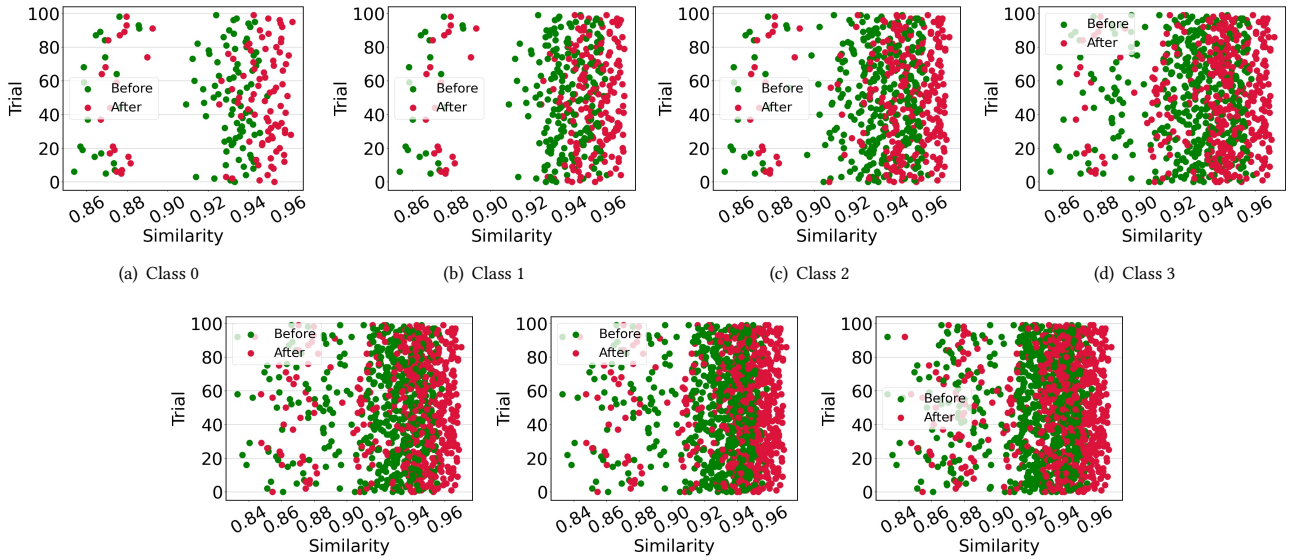
E Combination Methods

We employ five distinct element-wise combination methods:

- **Multiplication:** Computes the element-wise product of $\mathbf{P}_{att}$ and $\mathbf{P}_{st}$, given by $\mathbf{P}_{\text{combined}} = \mathbf{P}_{att} \odot \mathbf{P}_{st}$.
- **Ratio:** Divides each element of $\mathbf{P}_{att}$ by the corresponding element in $\mathbf{P}_{st}$, ensuring numerical stability by adding a small constant ϵ to the denominator. The ratio is given by $\mathbf{P}_{\text{combined}} = \frac{\mathbf{P}_{att}}{\mathbf{P}_{st}+\epsilon}, \quad \epsilon = 10^{-10}$.
- **Subtraction:** Subtracts $\mathbf{P}_{st}$ from $\mathbf{P}_{att}$ element-wise, expressed as $\mathbf{P}_{\text{combined}} = \mathbf{P}_{att} - \mathbf{P}_{st}$.
- **Mean:** Averages $\mathbf{P}_{att}$ and $\mathbf{P}_{st}$ element-wise, given by $\mathbf{P}_{\text{combined}} = \frac{\mathbf{P}_{att}+\mathbf{P}_{st}}{2}$.
- **Addition:** Computes the element-wise sum of $\mathbf{P}_{att}$ and $\mathbf{P}_{st}$. The sum operation can be expressed as $\mathbf{P}_{\text{combined}} = \mathbf{P}_{att} + \mathbf{P}_{st}$.

F Model Hyperparameters and Performance

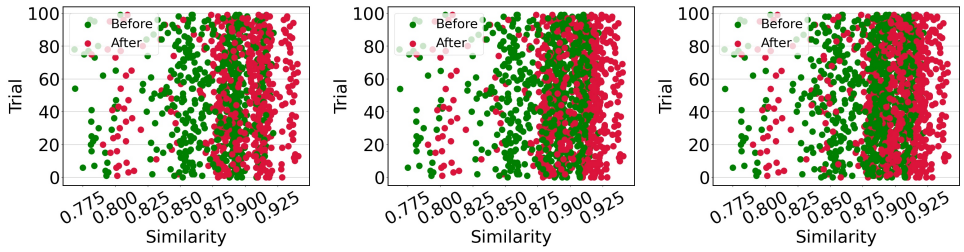
In Table 2, we show the hyperparameters that we use for the target models. Table 3 shows the test accuracy of the teacher and student models.



(a) Class 0 (b) Class 1 (c) Class 2 (d) Class 3

(e) Class 4 (f) Class 5 (g) Class 6

Figure 13: Cosine similarity of output probabilities between a teacher GraphSAGE and a student MLP for all seven classes in Cora.

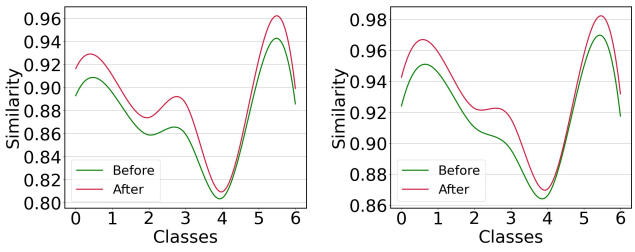


(a) Class 4 (b) Class 5 (c) Class 6

Figure 14: Probability similarity of the last three classes of Cora from the output of a student MLP and a teacher GCN.

Table 3: Test accuracies for target models and the student models.

Datasets	GCN	GAT	SAGE	APNNP
Cora	0.877	0.847	0.852	0.867
Student	0.877	0.847	0.847	0.857
Citeseer	0.755	0.755	0.747	0.767
Student	0.747	0.755	0.743	0.763
PubMed	0.854	0.854	0.880	0.852
Student	0.866	0.875	0.0.875	0.873
A-Computers	0.662	0.670	0.794	0.659
Student	0.699	0.713	0.834	0.729
A-Photo	0.803	0.855	0.929	0.848
Student	0.831	0.904	0.948	0.873
Facebook	0.844	0.861	0.855	0.845
Student	0.747	0.766	0.769	0.750
CS	0.885	0.902	0.937	0.887
Student	0.904	0.906	0.932	0.895
Physics	0.945	0.946	0.957	0.942
Student	0.930	0.947	0.957	0.946



(a) GCN-MLP (b) GraphSAGE-MLP

Figure 15: Cosine similarity of class probabilities for the 100th trial on Cora between teacher and student models.

Table 4: Dataset statistics.

Dataset	Type	# Nodes	# Edges	# Features	# Classes
Cora	Citation	2, 708	10, 556	1, 433	7
CiteSeer	Citation	3, 327	9, 104	3, 703	6
PubMed	Citation	19, 717	88, 648	400	3
Amazon Computers	Co-purchase	13, 752	491, 722	767	10
Amazon Photo	Co-purchase	7, 650	238, 162	745	8
Facebook	Social	22, 470	342, 004	128	4
CS	Co-Author	18, 333	153, 788	6, 805	15
Physics	Co-Author	34, 493	495, 924	8, 415	5

G Dataset Details

Table 4 presents the statistics for each dataset, including the graph type, total number of nodes, edges, features, and the number of classes within each graph.

H Fisher ratio and nearest-centroid validation accuracy.

Fig. 16 shows the Fisher ratio and nearest-centroid validation accuracy across datasets.

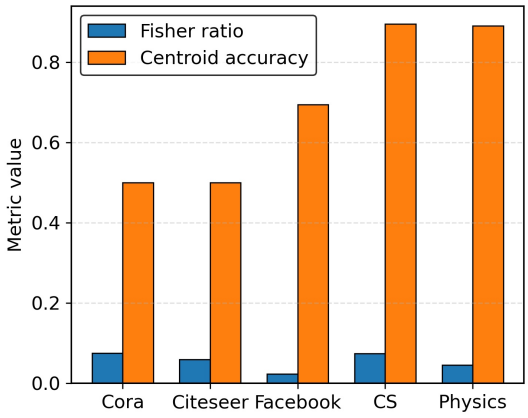
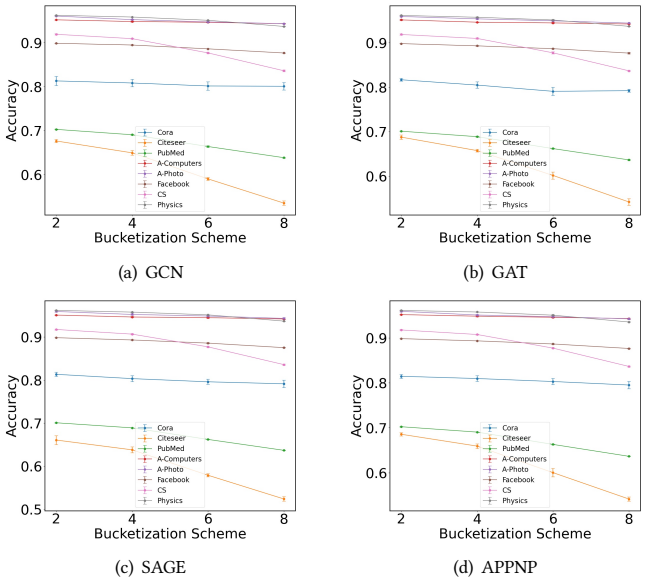


Figure 16: Fisher ratio and nearest-centroid validation accuracy across datasets, illustrating the global and local separability that governs weak-case attack performance.

I Effect of Bucketization Scheme on Property Inference Attack

We evaluate the effect of different bucketization schemes on the performance of the property inference attack, with results presented in Fig. 17 for the gradient boosting classifier. Our results show that, across all four models, the attack's performance declines as the number of buckets, k, increases. This is expected, as increasing k makes it more challenging for the attacker due to the finer distinctions between node classes. For our evaluation in Section 6.5, we chose $k = 3$ to provide the attacker with a broader but still informative classification. This enables the attacker to group nodes into three levels of importance: mildly important (low-degree), moderately important (moderate-degree), and highly important (high-degree) nodes.



(a) GCN (b) GAT

(c) SAGE (d) APPNP

Figure 17: Performance of the property inference attack with different models across eight datasets using different bucketization schemes.

J Attack via QR and CQR Evaluation

Table 5 shows the precision and AUC values for models across datasets at different FPR thresholds. Table 6 shows the precision and AUC values for models across datasets at different FPR thresholds.

K Defense Evaluation

Table 7 presents the results of the defense strategy across all models and datasets. The findings are consistent with the observations discussed in Section 7 and indicate that, while introducing noise reduces the attack's discriminative power, the reduction is not substantial. Furthermore, the precision remains largely unchanged compared to scenarios without noise, across all models and datasets.

L Complexity and Scaling.

The attacks rely solely on student posteriors, resulting in a runtime that scales linearly with the number of nodes and remains independent of graph density or edge count. For the reconstruction-loss attack, training an autoencoder on n nodes for E epochs with posterior dimension C, hidden widths (h_1, h_2), and latent size k incurs a cost of $\Theta(En(2Ch_1 + h_1h_2 + 2h_2k))$. In the CQR attack, the dominant cost arises from CatBoost training and calibration, given by $\Theta(T_{\text{cb}}n_{\text{tr}}Fd)$ and $\Theta(T_{\text{cb}}(n_{\text{val}} + n_{\text{te}})d)$, where T_{cb}, d, and F denote the number of trees, tree depth, and feature width, and n_{tr}, n_{val}, and n_{te} correspond to the training, validation, and test splits. The QR variant follows the same form but omits the calibration term. Overall, both attacks are computationally lightweight, with cost growth linear in the number of accessible samples rather than the size or connectivity of the underlying graph.

Table 5: Precision and AUC values for models across datasets at FPR thresholds 0.1%, 0.5%, and 1%, using QR.

Model	FPR	Cora		Citeseer		PubMed		A-Computers		A-Photo		Facebook		CS		Physics	
		Prec.	AUC	Prec.	AUC	Prec.	AUC	Prec.	AUC	Prec.	AUC	Prec.	AUC	Prec.	AUC	Prec.	AUC
GCN	0.1%	0.8903	0.6842	0.8906	0.7865	0.8889	0.5779	0.8894	0.5843	0.8889	0.6035	0.8890	0.5131	0.8888	0.5861	0.8888	0.5571
	0.5%	0.8926	0.6520	0.8906	0.7857	0.8888	0.5794	0.8899	0.5776	0.8889	0.6122	0.8890	0.5144	0.8888	0.5877	0.8888	0.5598
	1%	0.8891	0.7030	0.8902	0.7913	0.8888	0.5786	0.8893	0.5801	0.8889	0.6056	0.8889	0.5269	0.8889	0.5821	0.8888	0.5584
GAT	0.1%	0.8883	0.6958	0.8926	0.7261	0.8888	0.6272	0.8888	0.5586	0.8887	0.5619	0.8889	0.5710	0.8889	0.6134	0.8888	0.5416
	0.5%	0.8873	0.7129	0.9003	0.6403	0.8889	0.6330	0.8890	0.5550	0.8892	0.5671	0.8889	0.5679	0.8888	0.6360	0.8888	0.5338
	1%	0.8885	0.6895	0.9014	0.6849	0.8888	0.6336	0.8891	0.5490	0.8889	0.5457	0.8889	0.5657	0.8889	0.6246	0.8888	0.5405
SAGE	0.1%	0.8883	0.7519	0.8964	0.6735	0.8888	0.6151	0.8887	0.5497	0.8895	0.5260	0.8889	0.5768	0.8888	0.6499	0.8889	0.5443
	0.5%	0.8893	0.7451	0.8899	0.7273	0.8888	0.6149	0.8889	0.5470	0.8889	0.5801	0.8889	0.5751	0.8889	0.6434	0.8888	0.5436
	1%	0.8895	0.7235	0.8999	0.7012	0.8888	0.6102	0.8887	0.5484	0.8888	0.5843	0.8889	0.5814	0.8892	0.6349	0.8888	0.5445
APPNP	0.1%	0.8905	0.6157	0.8889	0.7812	0.8888	0.6030	0.8888	0.5629	0.8891	0.5616	0.8889	0.5517	0.8889	0.6492	0.8888	0.5498
	0.5%	0.8898	0.5912	0.8889	0.7525	0.8888	0.5983	0.8887	0.5657	0.8890	0.5687	0.8889	0.5455	0.8889	0.6338	0.8888	0.5505
	1%	0.8901	0.6219	0.8888	0.7676	0.8888	0.5996	0.8887	0.5633	0.8890	0.5649	0.8889	0.5499	0.8888	0.6297	0.8888	0.5517

Table 6: Precision and AUC values for models across datasets at three FPR thresholds (0.1%, 0.5%, and 1%) using CQR.

Model	FPR	Cora		Citeseer		PubMed		A-Computers		A-Photo		Facebook		CS		Physics	
		Prec.	AUC	Prec.	AUC	Prec.	AUC	Prec.	AUC	Prec.	AUC	Prec.	AUC	Prec.	AUC	Prec.	AUC
GCN	0.1%	0.8891	0.6531	0.8891	0.7670	0.8889	0.5656	0.8884	0.5739	0.8887	0.5855	0.8890	0.5254	0.8888	0.5811	0.8888	0.5549
	0.5%	0.8904	0.7095	0.8910	0.7995	0.8889	0.5664	0.8909	0.5807	0.8889	0.5812	0.8891	0.5256	0.8888	0.5894	0.8888	0.5508
	1%	0.8901	0.6805	0.8892	0.7732	0.8889	0.5660	0.8902	0.5813	0.8889	0.5894	0.8888	0.5214	0.8888	0.5794	0.8888	0.5565
GAT	0.1%	0.8885	0.6941	0.8988	0.6754	0.8888	0.6092	0.8906	0.5530	0.8891	0.5580	0.8889	0.5556	0.8889	0.6109	0.8888	0.5344
	0.5%	0.8895	0.7370	0.8957	0.7837	0.8888	0.6194	0.8911	0.5475	0.8888	0.5560	0.8889	0.5606	0.8889	0.6315	0.8888	0.5322
	1%	0.8883	0.6934	0.9031	0.6825	0.8888	0.6198	0.8891	0.5479	0.8890	0.5492	0.8889	0.5559	0.8888	0.6164	0.8888	0.5340
SAGE	0.1%	0.8882	0.7072	0.8893	0.7449	0.8888	0.6074	0.8890	0.5459	0.8884	0.5343	0.8889	0.5609	0.8887	0.6126	0.8888	0.5381
	0.5%	0.8911	0.7356	0.8930	0.7572	0.8888	0.6089	0.8940	0.5435	0.8879	0.5562	0.8889	0.5587	0.8895	0.6262	0.8888	0.5391
	1%	0.8900	0.7618	0.8889	0.7691	0.8888	0.6075	0.8887	0.5523	0.8884	0.5745	0.8889	0.5640	0.8888	0.6202	0.8888	0.5445
APPNP	0.1%	0.8910	0.6014	0.8894	0.7542	0.8888	0.5877	0.8887	0.5514	0.8890	0.5481	0.8889	0.5338	0.8889	0.6063	0.8888	0.5464
	0.5%	0.8980	0.6393	0.8893	0.7376	0.8888	0.5879	0.8887	0.5465	0.8889	0.5499	0.8889	0.5365	0.8890	0.6178	0.8888	0.5452
	1%	0.8898	0.5937	0.8888	0.7528	0.8888	0.5892	0.8887	0.5542	0.8889	0.5474	0.8889	0.5374	0.8888	0.6160	0.8888	0.5463

Table 7: Precision and AUC values for different models across datasets at different noise scales (0.1, 1, 5, and 10).

Model	Noise scales	Cora		Citeseer		PubMed		A-Computers		A-Photo		Facebook		CS		Physics	
		Prec.	AUC	Prec.	AUC	Prec.	AUC	Prec.	AUC	Prec.	AUC	Prec.	AUC	Prec.	AUC	Prec.	AUC
GCN	No Noise	0.8901	0.6805	0.8892	0.7732	0.8889	0.5660	0.8902	0.5813	0.8889	0.5894	0.8888	0.5214	0.8888	0.5794	0.8888	0.5565
	0.1	0.8892	0.6305	0.8898	0.7955	0.8888	0.5602	0.8899	0.5767	0.8889	0.5863	0.8890	0.5153	0.8888	0.5716	0.8889	0.5507
	1	0.8878	0.6990	0.8894	0.7863	0.8888	0.5618	0.8899	0.5746	0.8889	0.5815	0.8888	0.5131	0.8888	0.5761	0.8889	0.5670
	5	0.8887	0.7058	0.8886	0.7836	0.8888	0.5638	0.8885	0.5822	0.8889	0.5790	0.8890	0.5167	0.8888	0.5804	0.8888	0.5628
	10	0.8879	0.7329	0.8891	0.7915	0.8888	0.5626	0.8889	0.5803	0.8889	0.5816	0.8890	0.5169	0.8889	0.5695	0.8888	0.5613
GAT	No Noise	0.8883	0.6934	0.9031	0.6825	0.8888	0.6198	0.8891	0.5479	0.8890	0.5492	0.8889	0.5559	0.8888	0.6164	0.8888	0.5340
	0.1	0.8877	0.6046	0.8906	0.7625	0.8888	0.6180	0.8889	0.5514	0.8889	0.5560	0.8889	0.5509	0.8888	0.5675	0.8888	0.5303
	1	0.8873	0.5987	0.8896	0.7880	0.8888	0.6150	0.8907	0.5500	0.8889	0.5484	0.8889	0.5522	0.8887	0.6041	0.8888	0.5322
	5	0.8879	0.6650	0.8887	0.7403	0.8888	0.6120	0.8888	0.5498	0.8889	0.5528	0.8889	0.5515	0.8888	0.6046	0.8888	0.5306
	10	0.8875	0.5724	0.8890	0.7492	0.8888	0.6138	0.8890	0.5449	0.8889	0.5438	0.8889	0.5523	0.8888	0.5989	0.8888	0.5308
SAGE	No Noise	0.8890	0.7618	0.8889	0.7691	0.8888	0.6075	0.8888	0.5523	0.8887	0.5745	0.8888	0.5640	0.8888	0.6202	0.8888	0.5445
	0.1	0.8890	0.6364	0.8880	0.7766	0.8888	0.5879	0.8900	0.5502	0.8886	0.5494	0.8889	0.5550	0.8889	0.6231	0.8888	0.5383
	1	0.8888	0.6271	0.8897	0.6867	0.8888	0.5919	0.8886	0.5372	0.8883	0.5480	0.8889	0.5539	0.8888	0.6154	0.8888	0.5462
	5	0.8880	0.7166	0.8909	0.7254	0.8888	0.5908	0.8887	0.5469	0.8887	0.5586	0.8889	0.5537	0.8888	0.5983	0.8888	0.5267
	10	0.8879	0.7243	0.8891	0.7245	0.8888	0.5983	0.8886	0.5458	0.8889	0.5644	0.8889	0.5511	0.8888	0.6260	0.8888	0.5434
APPNP	No Noise	0.8898	0.5937	0.8898	0.7528	0.8888	0.5892	0.8887	0.5542	0.8889	0.5474	0.8889	0.5374	0.8888	0.6160	0.8888	0.5463
	0.1	0.8938	0.5948	0.8903	0.7098	0.8889	0.5808	0.8887	0.5538	0.8889	0.5460	0.8889	0.5262	0.8888	0.5996	0.8888	0.5389
	1	0.8915	0.6352	0.8894	0.7481	0.8888	0.5821	0.8887	0.5545	0.8888	0.5451	0.8889	0.5246	0.8888	0.5961	0.8888	0.5521
	5	0.8911	0.6241	0.8893	0.7419	0.8888	0.5816	0.8887	0.5525	0.8889	0.5466	0.8889	0.5262	0.8888	0.5891	0.8888	0.5535
	10	0.8931	0.6225	0.8893	0.6869	0.8888	0.5855	0.8887	0.5555	0.8890	0.5504	0.8889	0.5254	0.8888	0.5878	0.8888	0.5514