Imprint of the Forgotten: Stealthy Membership Inference In Unlearned Graph Neural Networks

He Zhang, Bang Wu, Xiaoning Liu, Karin Verspoor, Xun Yi

School of Computing Technologies, RMIT University
{he.zhang; bang.wu; xiaoning.liu; karin.verspoor; xun.yi}@rmit.edu.au

Abstract

Graphs effectively model interactions in real-world applications such as social and trade networks, where Graph Neural Networks (GNNs) excel at tasks such as link prediction to enhance user experiences. Despite these benefits, users raise privacy concerns as user data can be exploited to improve GNN performance without consent. Accordingly, various graph unlearning methods have been developed. Prior work shows that comparing models before and after unlearning enables attackers to launch *former membership inference attacks* (FMIA) on unlearned data. However, the imprint of unlearned data left in the unlearned model itself remains underexplored, and existing membership inference methods mainly exploit vulnerability of training data, making them ineffective for identifying unlearned data. To this end, we conducted a theoretical analysis and proposed an attack framework targeting unlearned GNNs by learning the distribution patterns of unlearned data to distinguish them from normal test data. Extensive experiments on four real-world datasets and GNN architectures confirm our framework's effectiveness and reveal significant vulnerabilities in current graph unlearning methods.

Introduction

Graphs, composed of nodes representing entities and links indicating interactions (Liu et al. 2025a; Cheng et al. 2025a), provide effective frameworks for data representation in areas such as social and trade networks (Sharma et al. 2024). Graph Neural Networks (GNNs) excel at modeling such data (Wu et al. 2022b; Wang et al. 2023), and have become prominent by significantly enhancing user experiences in real-world applications (Cheng et al. 2025b; Pan et al. 2025). For instance, in social platforms, nodes denote users and items (e.g., posts), while edges represent interactions (Colacrai et al. 2024; Lin et al. 2024). Well-trained GNNs can serve as recommendation systems (Jin et al. 2023a; Wang et al. 2025a), delivering trusted personalized content through link prediction to enrich the user's life (Vombatkere et al. 2024; Lin, Guo, and Jia 2024; Zhang et al. 2025c).

Although these AI services offer benefits (Jiang et al. 2025b,a; Liu et al. 2024a; Chi, Guo, and Jia 2025), users express privacy concerns (Zhang et al. 2025a; Li et al. 2025a) about the potential misuse of sensitive information (Liu et al. 2025b). Without explicit permission (Eaton 2024), developers may collect user data intentionally or unintentionally to improve model performance, as current data-driven models require extensive and detailed datasets (Zhang et al. 2025b). Previous research shows that links between users are vulnerable to privacy attacks (Zhang et al. 2024b), which leads affected users to request data removal (Hu et al. 2024b; Liu et al. 2024b) and expect GNNs to erase learned information (Said et al. 2023; Zhang et al. 2024a). This expectation also aligns with regulations such as the GDPR, which guaranties the "right to be forgotten".

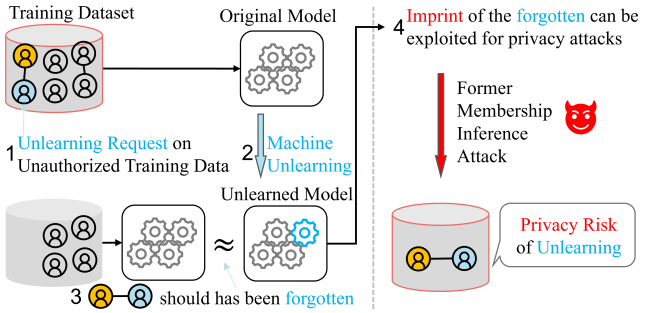

Figure 1: Overview of Privacy Risks in Unlearning.

To address user privacy concerns, various graph unlearning methods have been developed, from the prior-design required SISA (Chen et al. 2022) to plug-and-play solutions like GNNDelete (Cheng et al. 2023). Although these methods aim to protect privacy (e.g., by avoiding membership inference attacks (MIA) (Wu et al. 2022a)), adversaries can still reveal the privacy of unlearning data (Hu et al. 2024a; Li et al. 2025b; Liu et al. 2025c). For example, with access to models before and after unlearning, a method called MIMU (Chen et al. 2021) can infer *if a sample was used in the training dataset and then unlearned (i.e., its former membership information)* by comparing the predictions of both models on identical samples. The revelation of such former membership information poses privacy risks similar to traditional membership inference (Chen et al. 2021).

Unfortunately, existing membership inference methods either rely on impractical assumptions or are ineffective at identifying former membership. For example, MIMU (Chen et al. 2021) requires access to multiple model versions,

which limits its practicality. Other MIA approaches, like LinkTeller (Wu et al. 2022a), depend on issuing multiple queries to the target model, making them easily detectable (Juuti et al. 2019). Moreover, these MIA techniques typically exploit model memorization on training data (Zhang et al. 2023), while overlooking the residual imprints left by unlearning, thereby limiting their ability to identify former membership. Additionally, existing MIA on links focus on node classification tasks, and their underlying rationale cannot be adapted to GNNs for link prediction. Table 1 compares our approach with existing typical methods.

To this end, we propose a former membership inference attack (FMIA) framework targeting unlearned GNNs. Our intuition is the unlearned data exhibits different prediction distributions from normal test data, leaving detectable imprints in the unlearned model. This allows our attack to be both practical and stealthy, as it only requires querying the unlearned model as a benign user. However, identifying unlearned data is non-trivial, especially with low-dimensional outputs (e.g., 2D vectors for link prediction). To address this, we introduce a framework with dual encoders and specially designed loss to enhance the distinguishability of unlearned samples. Our contributions are summarized as follows.

- For the first time, we study the former membership inference attack in the context of GNNs, and formally define the design goal of the link-level privacy attack.
- We theoretically demonstrate the imprint existence of unlearned data and propose a framework with specialized architecture and loss function, enabling practical and stealthy attacks on unlearned GNNs.
- Comprehensive evaluations on four datasets and GNN architectures demonstrate the vulnerability of unlearned GNNs and the effectiveness of our approach.

Preliminaries

Graphs and Link Prediction. $G = \{\mathcal{V}, \mathcal{E}\}$ denotes a graph, where $\mathcal{V}$ is the set of nodes $\{v_1, \ldots, v_{|\mathcal{V}|}\}$, and $\mathcal{E}$ is the set of edges. It can be expressed as $G = \{\mathbf{A}, \mathbf{X}\}$. The matrix $\mathbf{X} \in \mathbb{R}^{|\mathcal{V}| \times d}$ contains node features, with the i-th row depicting features of node v_i, and d is the feature dimension. An edge between v_i and v_j is indicated by $\mathbf{A}_{i,j} = 1$, while $\mathbf{A}_{i,j} = 0$ indicates absence (Wang et al. 2024a, 2025b; Lin et al. 2025; Li et al. 2025c; Zheng et al. 2025). As a common computational task on graphs, e.g., user-item interaction prediction (Jin et al. 2023b; Guo et al. 2025, 2024; Wang et al. 2024b), a link prediction GNN f aims to predict potential links between nodes within G, where each pair of nodes is assigned a score reflecting its probability of having an edge. Given the architecture of a GNN f, its optimal parameter is obtained by training on a subgraph $G_s = \{\mathcal{V}_s, \mathcal{E}_s\} = \{\mathbf{A}_s, \mathbf{X}_s\}$, i.e., $\theta^* = \mathcal{A}(f, G_s) = \operatorname{argmin}_\theta \ell(\mathbf{A}_s, \widetilde{\mathbf{A}}_s)$, where $\widetilde{\mathbf{A}}_s = f_\theta(G_s)$. Here, $\mathcal{A}$ is an algorithm used to determine the optimal parameter θ^* for the model f. ℓ denotes a loss function (e.g., cross-entropy loss) that quantifies the discrepancy between the ground truth $\mathbf{A}_s$ and the predicted $\widetilde{\mathbf{A}}_s$. Once *training from scratch* with G_s is completed, the well-trained f_{θ^*} can predict the existence of edges in G.

Attack Method	FMI	MA	SMAO	BA
Link Stealing		✓	✓	✓
GRA		✓	✓	
MIMU	✓	✓		✓
HRec	✓			
FMIA (Ours)	✓	✓	✓	✓

Table 1: Comparison with Existing Privacy Attack Methods. FMI, MA, SMAO, and BA denote Former Membership Inference, Model-agnostic Attacks, Single Model Access Only, and Black-box Attack, respectively (see Appendix for details on these methods).

Machine Unlearning. Unlike machine learning, which acquires knowledge from data, machine unlearning aims to remove it. For example, in the context of GNNs, the *edge unlearning request* could be defined as a set of edges previously included in training data, i.e. $\mathcal{E}_u = \{(v_1, v_2), \cdots, (v_i, v_j)\}$ and $\mathcal{E}_u \subset \mathcal{E}_s$. Ideally, the model developer should remove the unlearning data $\mathcal{E}_u$ from the original training data $G_s = \{\mathcal{V}_s, \mathcal{E}_s\}$ and retrain the target model from scratch to obtain the retrained model parameter $\theta^*_{re} = \mathcal{A}(f, G_s \setminus \mathcal{E}_u)$. While retraining ensures exact unlearning, its high computational cost motivates the study of approximate unlearning. Concretely, an approximate unlearning method $\mathcal{U}$ aims to achieve $\min \texttt{dis}(\mathbb{P}(\theta^*_{re}), \mathbb{P}(\theta^*_{ul}))$, where $\mathbb{P}(\cdot)$ represents the distribution, $\theta^*_{ul} = \mathcal{U}(\mathcal{A}(f, G_s), G_s, \mathcal{E}_u)$ indicates the model parameter obtained by the unlearning method $\mathcal{U}$, and $\texttt{dis}(\cdot, \cdot)$ measures the distance between two distributions. Note that, instead of solving the above optimization, some methods implement machine unlearning by using $\min \texttt{dis}(f_{\theta^*_{re}}, f_{\theta^*_{ul}})$, which expects the unlearned model and the retrained model to have similar predictions.

Problem Formulation

System Model and Setting. In this paper, we focus on the representative link prediction task. The model developer collects a portion of the links as training data and devises a GNN model with the transductive setting, where the node embeddings of these links are used to predict the links in the test data. Querying the model with a node pair returns a two-dimensional vector, indicating the probabilities of the pair being unconnected or connected. For example, a prediction of $[0.1, 0.9]$ for nodes (v_i, v_j) suggests a connection.

Following previous studies (Wu et al. 2023), we consider that only partial users in the training data have authorized the model developer to collect their data. Due to privacy concerns (e.g., data misuse (Wu et al. 2024)), a set of users whose connections were misused issue unlearning requests on these unauthorized links to the model developer. In response, the model developer will gather unlearning requests and apply a graph unlearning method to forget the knowledge learned from them, resulting in an unlearned GNN model that replaces the original model to serve users.

Threat Model and Attack Setting. After identifying data misuse, data contributors issue unlearning requests to the model developer and receive notifications upon completion. One *Honest-but-Curious* data contributor aims to infer un-

learning requests owned by other data contributors. Upon receiving the unlearning request $\mathcal{E}_u$, the model developer applies a graph unlearning method to obtain the unlearned GNN model $f_{\theta^*_{ul}}$, which is the target model of privacy attackers. Here, we consider the black-box attack setting, i.e., the adversary does not know the architecture and parameter of $f_{\theta^*_{ul}}$ and can only query it with node IDs. Following a previous study (Chen et al. 2021), we assume that the adversary possesses a shadow dataset, which may come from the same or another distribution as the target dataset. The attacker knows the method used for unlearning or is unaware of it; we assess both settings in empirical evaluations.

Attack Goal. The attacker aims to infer the *former membership*, i.e., if two nodes are connected in the previous training data and then this link is included in the unlearning data $\mathcal{E}_u$. Note that identifying the former membership of links in a graph exposes it to threats, which share the same privacy risk as traditional MIAs (Chen et al. 2021). Specifically, the Former Membership ($\texttt{FM}$) inference can be formulated as

$$\texttt{FM}(i,j) = \begin{cases} 1 & \text{if } Atk(\mathbf{P}_{i,j}) > \eta, \\ 0 & \text{otherwise .} \end{cases} \tag{1}$$

where the inference model Atk outputs the inference score for link prediction $\mathbf{P}_{i,j}$ and η is a threshold.

Former Membership Inference Attack

Theoretical Evidence of Unlearning Imprints

To study the imprint of unlearning, the hypothesis test is used to measure the privacy risk of edges under Former Membership Inference Attacks (FMIA).

Hypothesis Test. Given a pair of nodes (v_i, v_j) in the model inference phase, attackers in FMIAs aim to discriminate the following two hypotheses:

- Null hypothesis H_0: For the unlearned model, (v_i, v_j) is not included in the unlearning request $\mathcal{E}_u$. The query result $\mathbf{P}_{i,j} = f_{\theta^*_{ul}}(v_i, v_j)$ belongs to the link prediction distribution of the normal test data.
- Alternative hypothesis H_1: For the unlearned model, (v_i, v_j) is included in the unlearning request $\mathcal{E}_u$. The query result $\mathbf{P}_{i,j} = f_{\theta^*_{ul}}(v_i, v_j)$ belongs to the link prediction distribution of the unlearned data $\mathcal{E}_u$.

Given H_0 and H_1, we use the quotation symbol to denote the attacker's decision, i.e., "H_0" and "H_1". Based on the ground-truth and FMIA outcomes, attackers may commit two types of inference errors, indicated by the probabilities $\Pr(\text{“}H_1\text{”} \mid H_0)$ and $\Pr(\text{“}H_0\text{”} \mid H_1)$, where $\Pr(\cdot)$ denotes the probability function. In line with previous studies (Murakonda, Shokri, and Theodorakopoulos 2021), we measure the privacy risk of $f_{\theta^*_{ul}}$ using two statistical measures: the error rate α (false positive rate) and the detection power β (true positive rate). Specifically, α indicates the rate of mistaken identification of H_0, i.e., $\Pr(\text{“}H_1\text{”} \mid H_0)$, while β is the probability of correctly identifying H_1, i.e., $\Pr(\text{“}H_1\text{”} \mid H_1)$.

Vulnerability of Unlearning. In FMIAs, attackers adopt $\texttt{FM}(i,j) = \texttt{FM}(\mathbf{P}_{i,j})$ as the inference score for the node pair (v_i, v_j). Following prior work (Li et al. 2021), we assume $\texttt{FM}(i,j) \sim N(\mu_0^{\texttt{FM}}, \sigma^{\texttt{FM}})$ when $(v_i, v_j) \notin \mathcal{E}u$, and $\texttt{FM}(i,j) \sim N(\mu_1^{\texttt{FM}}, \sigma^{\texttt{FM}})$ if $(v_i, v_j) \in \mathcal{E}u$, with $\mu_1^{\texttt{FM}} \geq \mu_0^{\texttt{FM}}$. Here, $\mu_{0/1}^{\texttt{FM}}$ and $\sigma^{\texttt{FM}}$ denote the mean and variance of former membership scores. According to the *Neyman-Pearson lemma* (Neyman and Pearson 1933), for a fixed α, the likelihood ratio test achieves the highest β among all decision rules, which indicates that the optimal threshold τ for binary inference is defined by the intersection of $N(\mu_0^{\texttt{FM}}, \sigma^{\texttt{FM}})$ and $N(\mu_1^{\texttt{FM}}, \sigma^{\texttt{FM}})$. Note that $z_{1-\alpha} + z_\beta$ reflects the privacy risk for unlearned edges, where $z_{1-\alpha}$ and z_β are the $1-\alpha$ and β quantile of $N(0,1)$, respectively.

Theorem 1. *The imprint of unlearned data can be measured by the vulnerability of an unlearned GNN $f_{\theta^*_{ul}}$ to former membership inference, i.e., $V(f_{\theta^*_{ul}}) = z_{1-\alpha} + z_\beta$. We have*

$$\mathbb{E}(\left\|V(f_{\theta^*_{ul}})\right\|) > 0.$$

Proof. Using the typical message-passing mechanism in graph learning, we calculate the difference between unlearned and normal test data to reach this conclusion (see Appendix: "Proof on Unlearning Imprint" for details). □

Inference Attack Framework

According to Theorem 1, the prediction distribution of unlearned data is different from that of normal test data in the unlearned GNN, which can be exploited as the imprint of unlearning to issue former membership inference attacks. To quantitatively reveal this vulnerability, we propose an attack framework called FMIA, which only needs to query the unlearned GNN model in privacy attacks. Specifically, our framework consists of the following two distinct phases. (1) In the posterior generation phase, for any given pair of nodes (v_i, v_j), the attacker queries the unlearned target model $f_{\theta^*_{ul}}$ to retrieve its posterior $\mathbf{P}_{i,j}$, which reflects the likelihood of a link between nodes v_i and v_j in the downstream tasks. (2) In the inference phase, our attack model determines whether the edge (v_i, v_j) was initially part of the training data before being unlearned in the derivation of the unlearned target model $f_{\theta^*_{ul}}$. Note that our framework is a learning-based method, whose training data is obtained by using the shadow dataset owned by attackers. Details of this common operation can be found in the Appendix. Next, we present the model architecture and loss functions for acquiring a well-trained attack model to capture the imprint of unlearning.

Model Architecture. For GNNs in link prediction, the prediction result $\mathbf{P}_{i,j} \in [0,1]^2$ on two nodes is a vector with two real values. Due to the learning nature of our framework, the limited feature dimension of $\mathbf{P} \in [0,1]^{|D|\times 2}$ poses a challenge to reveal the vulnerability of unlearned GNNs.

To this end, given the training data $D = \{(\mathbf{P}_{i,j}, \mathbf{Y}_{i,j})\}$ ($\mathbf{Y}_{i,j} = 0$ or 1), we propose employing the architecture illustrated in Fig. 2 to enhance the representation of the training samples, thereby increasing the distinguishability between normal test data and unlearned data. Specifically, the inference result is obtained by

$$\widetilde{\mathbf{Y}} = f_C(\mathbf{E}_1 \| \mathbf{E}_2), \tag{2}$$

where $f_C(\cdot)$ denotes a binary classifier, and $\|$ indicates the concatenation operation. Here, $\mathbf{E}_1 \in \mathbb{R}^k$ and $\mathbf{E}_2 \in \mathbb{R}^k$ are

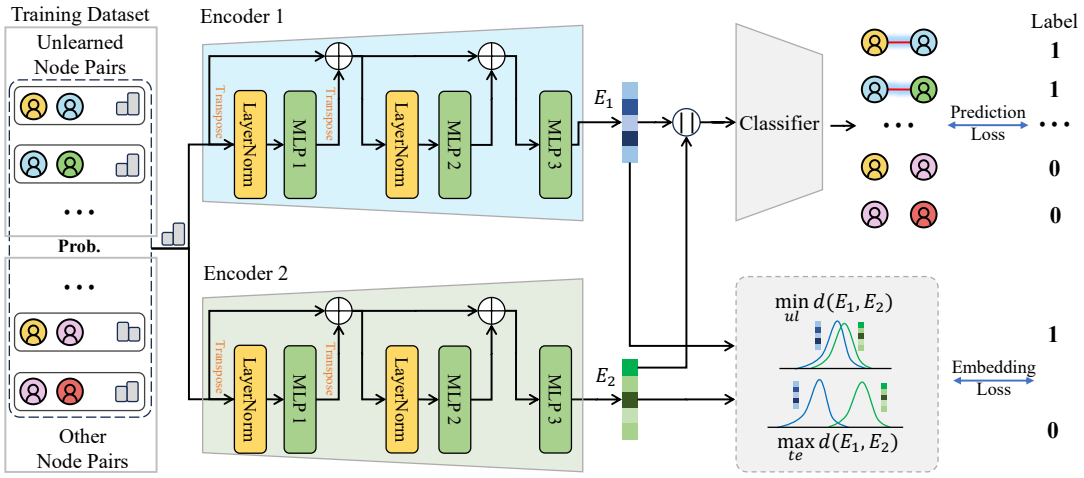

Figure 2: An Overview of the **Former Membership Inference Attack (FMIA)** Framework.

two k-dimensional vectors representing the sample embeddings. We obtain two different versions of the embeddings for the input samples as follows:

$$\mathbf{E}_1 = f_{E_1}(\mathbf{P}), \quad \mathbf{E}_2 = f_{E_2}(\mathbf{P}), \tag{3}$$

where $f_{E_1}(\cdot)$ and $f_{E_2}(\cdot)$ represent two encoding modules with the same architecture but distinct parameters. Given $\mathbf{X} = \mathbf{P}$ as the input, $f_E(\cdot) = f_{d\text{-}e} \circ f_{\text{-}s} \circ f_{c\text{-}s}(\cdot)$ is implemented with the following architecture.

$$\begin{aligned} f_{c\text{-}s}(\mathbf{X}) &= \mathbf{X} + \texttt{Tra}(\text{MLP}_1(\texttt{LayNor}(\texttt{Tra}(\mathbf{X})))), \\ f_{i\text{-}s}(\mathbf{X}) &= \mathbf{X} + \text{MLP}_2(\texttt{LayNor}(\mathbf{X})), \\ f_{d\text{-}e}(\mathbf{X}) &= \text{MLP}_3(\mathbf{X}), \end{aligned} \tag{4}$$

where `Tra` indicates the Transpose operation, `LayNor` represents the Layer Normalization operation, and MLP denotes MultiLayer Perceptron with ReLU as the activation function. Here, $f_{c\text{-}s}(\cdot)$ and $f_{i\text{-}s}(\cdot)$ aim to capture cross-sample and inter-sample patterns, and $f_{d\text{-}e}(\cdot)$ indicates a dimensionality expansion operation that maps the sample to embedding space with a higher dimension to facilitate discrimination. Our classifier $f_C(\cdot) = f_{d\text{-}r} \circ f_{i\text{-}s} \circ f_{c\text{-}s}(\cdot)$ has a similar architecture as $f_E(\cdot)$, where $f_{d\text{-}r}(\cdot) = \text{MLP}(\cdot)$ denotes a dimensionality reduction operation to obtain inference results.

Loss Function. Given the above model, all parameters included in it are optimized using the following loss functions. (1) Utility loss. Given the training data $D = \{(\mathbf{P}_{i,j}, \mathbf{Y}_{i,j})\}$ where $N = |D|$, we assume that $(\mathbf{P}_{i,j}, \mathbf{Y}_{i,j})$ is the n-th sample in D. We employ the following cross-entropy loss to optimize the inference performance of our attack model.

$$\ell_c = \frac{-1}{N} \sum_{n=1}^{N} (\mathbf{Y}_{i,j} \log \widehat{\mathbf{Y}}_{i,j} + (1 - \mathbf{Y}_{i,j}) \log(1 - \widehat{\mathbf{Y}}_{i,j})). \tag{5}$$

(2) Embedding loss. Given the n-th sample, we propose to decrease the distance between the two versions of embeddings, i.e., $d_n = d(\mathbf{E}_1[n,:], \mathbf{E}_2[n,:])$, if it is an unlearned sample; conversely, we increase the distance if it is a normal test sample. The embedding difference above improves the distinguishability between unlearned and normal test samples, and the corresponding loss is defined as follows:

$$\ell_e = \frac{1}{N} \sum_{n=1}^{N} (\mathbf{Y}_{i,j} d_n^2 + (1 - \mathbf{Y}_{i,j}) \max(0, m - d_n)^2), \tag{6}$$

where m indicates the margin value used as a threshold.

Given the above two loss functions, the optimal parameter of the attack model is obtained by

$$\omega^* = \text{argmin}_\omega \, \ell_c + \ell_e. \tag{7}$$

With ω^*, our attack model can infer the former membership of node pairs in the target unlearned model $f_{\theta^*_{ul}}$.

Remarks. The dual-encoder design aims to amplify embedding differences between unlearned and normal samples, which better captures the subtle unlearning imprints in the 2D posteriors produced by link-prediction GNNs. Our dual encoders are trained with margin-based contrastive loss (Eq. 6), which reduces the similarity between embeddings from $\mathbf{E}_1$ and $\mathbf{E}_2$ for unlearned samples while increasing it for normal ones. This enhances the ability of the attack model to distinguish unlearned and normal samples in the embedding space and improves our attack effectiveness (see Fig. 7a).

Experiments

Experimental Setup

Datasets, **GNNs**, and **Metric**. We conduct evaluation on four datasets, including Cora (Kipf and Welling 2017), CS (Shchur et al. 2018), DBLP (Bojchevski and Günnemann 2018) and Pubmed (Kipf and Welling 2017), which are commonly used in the graph study. For the GNN architecture, we use representative GCN (Kipf and Welling 2017), GAT (Velickovic et al. 2018), GIN (Xu et al. 2019), and DeepGCN (Li et al. 2019) as the backbone model for the link prediction task. Following existing studies, the privacy risk of unlearned GNNs is measured by the ROC-AUC.

Unlearning Methods. Three representative unlearning methods based on different rationales are used in our evaluations, including Fine-tuning (Yao et al. 2024), GIF (Wu et al.

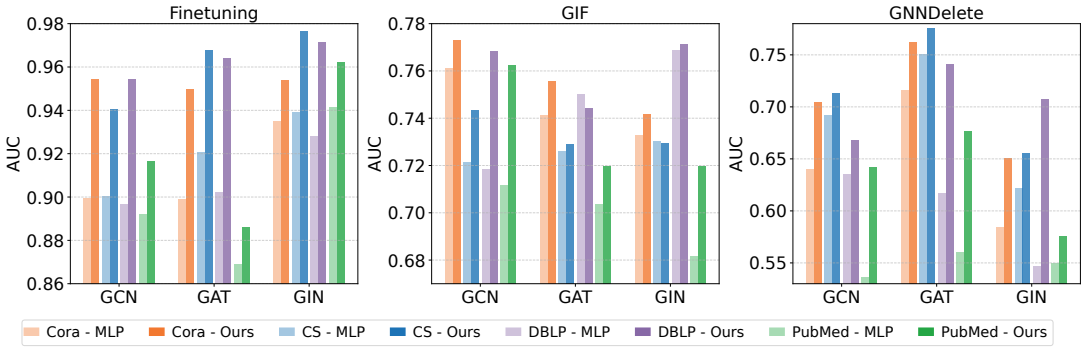

Figure 3: Comparison of Attack Performance (AUC ↑) Between Our FMIA Framework and Baseline Methods. In these figures, higher AUC scores indicate greater vulnerability of the unlearned GNNs to Former Membership Inference Attacks.

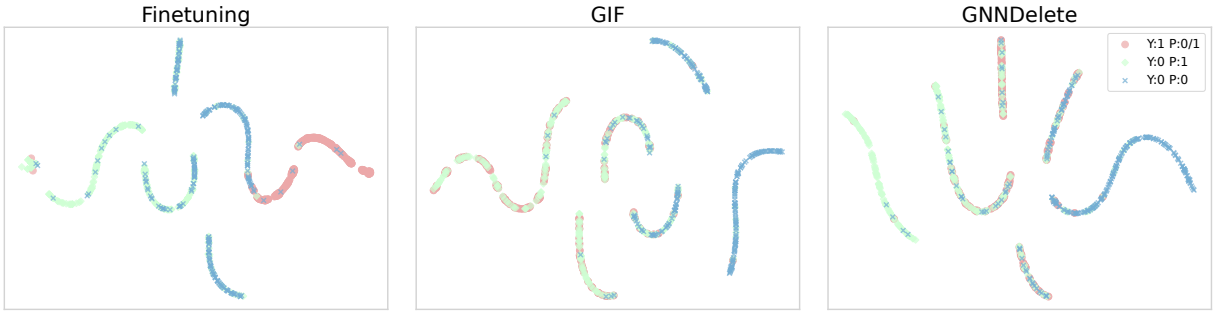

Figure 4: A Visualization of Link Prediction Probabilities for the Unlearned Model on the Cora Dataset with GAT Model. Within these results, pink points correspond to unlearned samples, green points indicate connected node pairs in test dataset, and blue points represent unconnected node pairs in the test dataset. Additionally, P denotes the ground truth connectivity status of each node pair (1 indicating connected pairs and 0 indicating unconnected pairs), whereas Y denotes the ground truth labels regarding whether a node pair is included in the unlearning request $\mathcal{E}_u$ of the unlearned model.

2023), and GNNDelete (Cheng et al. 2023). Unlike the SISA method (Chen et al. 2022), all three methods do not require introducing additional effort in the model training phase, and can directly implement unlearning in a plug-and-play manner after model training. This practicability motivates us to use them in our evaluations. Refer to the Appendix ("Unlearning Methods Used in Evaluations") for the rationale of these unlearning methods, and the utility of the unlearned GNNs is presented in the Appendix.

Baselines. To evaluate the effectiveness of our method, we compare it with the *random classifier* and the vanilla *MLP-based classifier*. In the random baseline, the inference result is generated randomly. In the MLP baseline, it employs the same training data generation pipeline, and then trains a two-layer MLP to implement the privacy attack. Note that the difference between our model and the baseline is the model architecture used as the attacker's model. Refer to the Appendix for more details (e.g., parameters, device, and code).

Attack Performance

We first evaluate the vulnerability of unlearned GNNs against FMIAs, followed by discussing the varying vulnerability of different unlearning methods.

Vulnerability Evaluations. As shown in Fig. 3, our empirical evaluation results confirm that unlearned GNNs are vulnerable to the FMIA. We observe that both our method and the MLP approach outperform the random baseline, whose AUC score is 50%, in privacy attacks due to leveraging additional shadow dataset. Moreover, our method achieves better attack performance than the standard MLP, as the use of specialized model architecture and loss functions potentially improves sample distinguishability.

Varying Levels of Vulnerability. Fig. 3 demonstrates that the unlearned GNN models have different levels of vulnerability against FMIAs. We observe that unlearned models obtained with the fine-tuning method have the highest privacy risk, whereas models via the GNNDelete method have the lowest vulnerability. To investigate the above varying vulnerability, we visualize the distribution of query results on the unlearned model. As shown in Fig. 4, the distinguishability between unlearned samples (i.e., pink points) and other samples (i.e., green and blue points) represents the level of unlearning imprint. Fig. 4 shows that:

- For unlearned GNNs with Fine-tuning, there is only limited distribution overlap between unlearned samples and other samples. Hence, it is easy to distinguish the unlearned node pairs from others, which facilitates the training of privacy attack models.
- For unlearned models using GIF, the overlapping focuses on unlearned samples and connected node pairs. Thus,

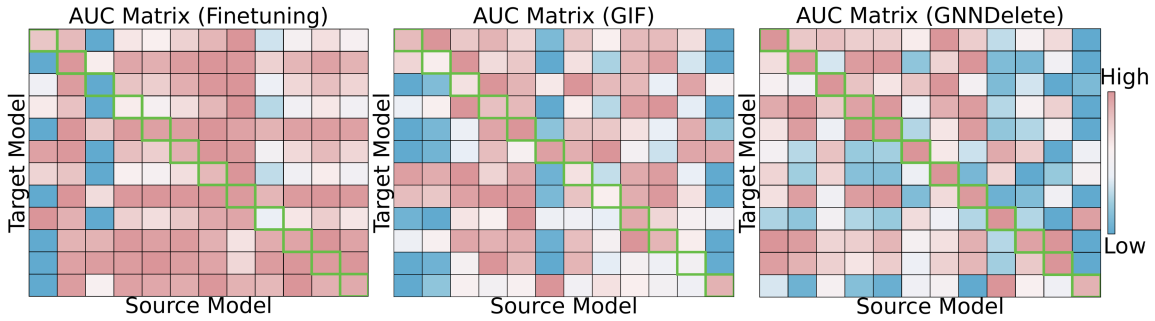

Figure 5: Attack Transferability Across Datasets and Models. The left, middle, and right matrices represent the attack results (AUC%) using Fine-tuning, GIF, and GNNDelete, respectively. The rows (top to bottom) and columns (left to right) follow the same indexing order based on the combinations of (Cora/CS/DBLP/PubMed, GCN/GAT/GIN).

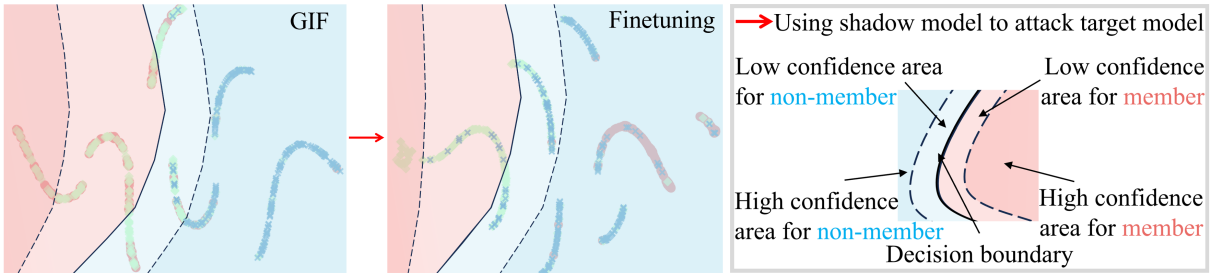

Figure 6: Decision Boundary Illustration in Transfer Attacks With Different Unlearning Methods (From GIF to Fine-tuning). These data distributions are derived from the DBLP dataset using the GAT model.

compared with Fine-tuning, it is harder for the attack model to discriminate the unlearned samples from others.

- For unlearned GNNs with GNNDelete, the distribution of unlearned samples overlaps with both connected node pairs and unconnected node pairs in the test data. Thus, it is more challenging to accurately identify the unlearned samples, which subsequently makes the unlearned models with GNNDelete have the lowest privacy risk.

The above varying degree of distribution overlap can be attributed to the rationale difference of unlearning methods, refer to the Appendix ("Prediction Distribution of Unlearned GNNs") for more discussions.

Attack Transferability

To further reveal the privacy risks of unlearned GNNs, we assess their vulnerability using attack models from different datasets, architectures, and unlearning methods.

Cross Datasets & Models. In this setting, given a target unlearned model with the specific dataset, GNN backbone, and unlearning technique, we use the attack models obtained from different datasets and/or GNN architectures to perform inference attacks. As shown in Fig. 5, for a specific target unlearned model (i.e., a row in the matrix), the attack results obtained by using different attack models are colored by considering the current row results. We observe that

(1) Unlearned GNNs are vulnerable to transfer attacks, evidenced by the AUC score in almost all cases being higher than 50%. When the shadow and target models use the same dataset and GNN architecture, the attack performance is generally higher, as shown by the diagonal squares in the matrices. In transfer scenarios (i.e., non-diagonal squares), attack effectiveness generally remains similar or only slightly decreases compared to the non-transfer cases.

(2) Different unlearning methods have varying AUC drops between non-transfer and transfer settings. That is, when comparing the results in the same column, the generalization abilities of an attack model with different unlearning methods are different. Using the diagonal results as reference, the AUC drop with Fine-tuning is lowest in most cases, while that associated with GNNDelete is the largest. From the perspective of machine learning, the varying generalization capability can be attributed to the quality of training data and the distribution shift of test data during the inference phase of attack models. For example, unlearned GNNs using Fine-tuning display a highly consistent unlearning imprint, where unlearned samples are typically well separated from other samples. This clear and consistent separation leads to the limited drop in attack performance during transfer attacks. See the Appendix ("Discussion on Transfer Attacks") for more details on performance drop differences.

Cross Unlearning Methods. We evaluate the vulnerability of the target unlearned GNN with one specific unlearning method by using attack models associated with different unlearning methods (see Appendix for more details).

We observe that better attack results are generally obtained when the target model and the shadow model share the same unlearning method, while cases with changed un-

learning methods have demonstrated various inference behaviors. For example, when using the attack model with the GIF method to attack unlearned GNNs with the Fine-tuning method, we observe inverted inference results (e.g., 13.3%). As shown in Fig. 6, the decision boundary of attack model using GIF cuts through the green samples, with blue samples located in the non-member region. However, due to a distribution shift in the target model, red samples are separated from green by blue samples and are misclassified as non-members. This leads to inference results that are significantly inverted. Note that for inverted AUCs (e.g., 13.3%), attackers can flip predictions to improve inference accuracy. More discussion can be found in the Appendix.

Ablation Study

Attack Model Variants. We use different variants of our method to attack unlearned GNNs, including (1) without using *encoder* modules and (2) using different distance functions in Equation (6). As shown in Fig. 7-(a), all variants generally have better attack performance than the MLP-based method, which can be attributed to the enhanced learning capacity of our attack framework. It also suggests that specific versions of our framework excel over others depending on the dataset, GNN, or unlearning strategy used, allowing the adversary to utilize the corresponding variant for optimal attack performance. Evaluations on other cases (e.g., datasets) can be found in the Appendix.

Different Size of Unlearning Data. Fig. 7-(b),(c),(d) illustrate the vulnerability of unlearned GNNs when varying the size of unlearned data. We observe that:

- For the Fine-tuning, the privacy risk of unlearned GNNs decreases as the size increases. This is because the overfitting on the unlearned data is alleviated as the size grows, weakening the overfitting-based unlearning imprints. This aligns with previous research (Shokri et al. 2017) on membership inference attacks, which shows that larger data size reduces privacy risks.
- For the GIF method, the vulnerability of unlearned GNNs is nearly constant. Although more data are unlearned, the gradient value of unlearned data still remains limited (almost zero), since the model before unlearning was originally optimized for these unlearning samples. Therefore, it cannot significantly update model parameters and lead to nearly constant vulnerability.
- For the GNNDelete method, the privacy risk of unlearned GNNs increases as the size increases. GNNDelete modifies the predictions (e.g., link existence probabilities) of unlearned data to match those of the remaining data. However, as more data are unlearned, the link existence probability in the remaining data decreases, diverging further from the distribution of general test data. This enlarged divergence makes it easier for an adversary to infer the former membership of unlearned data.

Vulnerability of GNN with Varying Layers. We further evaluate the vulnerability of deeper GNNs by using the DeepGCN model (Li et al. 2019). Here we focus on the Fine-tuning method, as resource-intensive GIF and architecture-modifying GNNDelete are not suitable on deeper models, especially on edge devices with limited resources. Table 2 shows that deeper GNNs are still vulnerable to FMIAs. Moreover, shallower models (e.g., DeepGCN with PubMed) generally exhibit higher privacy risks compared to deeper models. This is because shallower models, with fewer parameters, are more prone to overfitting and tend to memorize unlearned samples more than deeper models.

Refer to the Appendix for more discussions (e.g., transfer attacks, potential mitigation) on the FMIA.

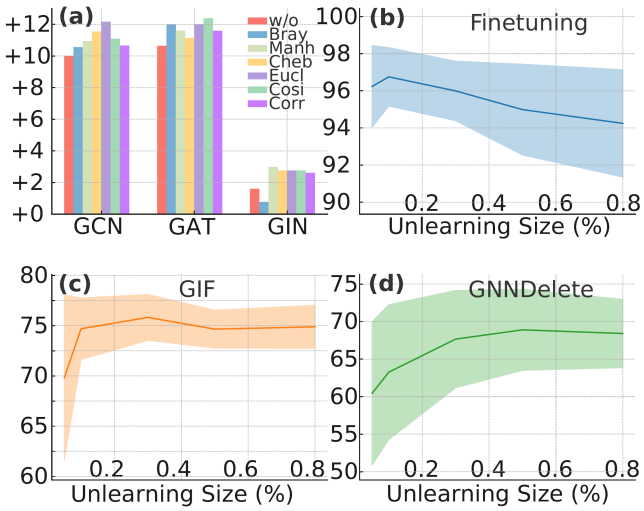

Figure 7: (a) Improvement in Attack Performance (AUC%) Compared to the MLP Attack Model (PubMed, GNNDelete). "w/o" indicates that the *encoder* modules are removed. The other bars represent our method using different distance functions in Equation (6). (b) (c) (d) Privacy risks (AUC%) w.r.t. different amounts of unlearned data. Results are obtained using all (dataset, GNN) combinations.

# Layer	Cora	CS	DBLP	PubMed
56	97.43	98.73	97.70	93.44
28	98.48	98.83	96.80	92.93
14	97.41	97.91	98.25	95.91
7	98.84	99.28	98.27	95.45
3	98.66	99.17	98.52	96.18

Table 2: Vulnerability (AUC% ↑) of DeepGCN.

Conclusion

This paper investigates the overlooked privacy risk of unlearned GNNs against former membership inference attacks. Existing graph unlearning methods, though designed to protect privacy, leave unintended imprints that attackers can exploit. To reveal this risk, we conduct a theoretical analysis and propose a practical and stealthy framework that performs privacy attacks by querying only the unlearned model. Extensive evaluations demonstrate the effectiveness of our method against various unlearning methods. Our future work includes exploring additional security risks and developing more secure graph unlearning strategies.

Acknowledgments

This research was supported in part by the Australian Research Council (ARC) under projects DP190102835, DP220102803, LP220200649, DP240102140, and LP250100307.

References

Bojchevski, A.; and Günnemann, S. 2018. Deep Gaussian Embedding of Graphs: Unsupervised Inductive Learning via Ranking. In *ICLR (Poster)*. OpenReview.net.

Chen, M.; Zhang, Z.; Wang, T.; Backes, M.; Humbert, M.; and Zhang, Y. 2021. When Machine Unlearning Jeopardizes Privacy. In *CCS*, 896–911. ACM.

Chen, M.; Zhang, Z.; Wang, T.; Backes, M.; Humbert, M.; and Zhang, Y. 2022. Graph Unlearning. In *CCS*, 499–513. ACM.

Cheng, J.; Dasoulas, G.; He, H.; Agarwal, C.; and Zitnik, M. 2023. GNNDelete: A General Strategy for Unlearning in Graph Neural Networks. In *ICLR*. OpenReview.net.

Cheng, X.; Zhang, Z.; Wang, J.; Fang, L.; He, C.; Guan, Q.; Pan, S.; and Luo, W. 2025a. Education-Oriented Graph Retrieval-Augmented Generation for Learning Path Recommendation. *CoRR*, abs/2506.22303.

Cheng, X.; Zhou, X.; Fang, L.; He, C.; Zhou, Y.; Luo, W.; Gong, Z.; and Guan, Q. 2025b. NR4DER: Neural Reranking for Diversified Exercise Recommendation. In *SIGIR*, 1738–1747. ACM.

Chi, G.; Guo, L.; and Jia, C. 2025. A Local Search Algorithm for the Radius-Constrained k-Median Problem. *Theory Comput. Syst.*, 69(1): 11.

Colacrai, E.; Cinus, F.; Morales, G. D. F.; and Starnini, M. 2024. Navigating Multidimensional Ideologies with Reddit's Political Compass: Economic Conflict and Social Affinity. In *WWW*, 2582–2593. ACM.

Eaton, K. 2024. LinkedIn Secretly Training its AIs on User Data. https://www.inc.com/kit-eaton/linkedin-secretlytraining-its-ais-on-user-data.html. Accessed: Mar 31, 2025.

Guo, L.; Jia, C.; Liao, K.; Lu, Z.; and Xue, M. 2024. Efficient Constrained K-center Clustering with Background Knowledge. In *AAAI*, 20709–20717. AAAI Press.

Guo, L.; Jia, C.; Liao, K.; Lu, Z.; and Xue, M. 2025. Near-Optimal Algorithms for Instance-Level Constrained k-Center Clustering. *IEEE Trans. Neural Networks Learn. Syst.*, 36(10): 18844–18858.

Hu, H.; Wang, S.; Dong, T.; and Xue, M. 2024a. Learn What You Want to Unlearn: Unlearning Inversion Attacks against Machine Unlearning. In *SP*, 3257–3275. IEEE.

Hu, Y.; Lou, J.; Liu, J.; Ni, W.; Lin, F.; Qin, Z.; and Ren, K. 2024b. ERASER: Machine Unlearning in MLaaS via an Inference Serving-Aware Approach. In *CCS*, 3883–3897. ACM.

Jiang, J.; Ding, H.; Wang, H.; and Yan, R. 2025a. Deep Spiking Neural Networks Driven by Adaptive Interval Membrane Potential for Temporal Credit Assignment Problem. *IEEE Trans. Emerg. Top. Comput. Intell.*, 9(1): 717–728.

Jiang, J.; Wang, L.; Jiang, R.; Fan, J.; and Yan, R. 2025b. Adaptive Gradient Learning for Spiking Neural Networks by Exploiting Membrane Potential Dynamics. In *IJCAI*, 4164–4172. ijcai.org.

Jin, D.; Wang, L.; Zhang, H.; Zheng, Y.; Ding, W.; Xia, F.; and Pan, S. 2023a. A survey on fairness-aware recommender systems. *Inf. Fusion*, 100: 101906.

Jin, D.; Wang, L.; Zheng, Y.; Song, G.; Jiang, F.; Li, X.; Lin, W.; and Pan, S. 2023b. Dual Intent Enhanced Graph Neural Network for Session-based New Item Recommendation. In *WWW*, 684–693. ACM.

Juuti, M.; Szyller, S.; Marchal, S.; and Asokan, N. 2019. PRADA: Protecting Against DNN Model Stealing Attacks. In *EuroS&P*, 512–527. IEEE.

Kipf, T. N.; and Welling, M. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *ICLR (Poster)*. OpenReview.net.

Li, G.; Müller, M.; Thabet, A. K.; and Ghanem, B. 2019. DeepGCNs: Can GCNs Go As Deep As CNNs? In *ICCV*, 9266–9275. IEEE.

Li, H.; Bai, L.; Ye, Q.; Hu, H.; Xiao, Y.; Zheng, H.; and Xu, J. 2025a. A Sample-Level Evaluation and Generative Framework for Model Inversion Attacks. In *AAAI*, 18287–18295. AAAI Press.

Li, N.; Zhou, C.; Gao, Y.; Chen, H.; Zhang, Z.; Kuang, B.; and Fu, A. 2025b. Machine unlearning: Taxonomy, metrics, applications, challenges, and prospects. *IEEE Transactions on Neural Networks and Learning Systems*.

Li, P.; Wang, Y.; Zhao, H.; Hong, P.; and Liu, H. 2021. On Dyadic Fairness: Exploring and Mitigating Bias in Graph Connections. In *ICLR*. OpenReview.net.

Li, S.; Liu, Y.; Wen, Q.; Zhang, C.; and Pan, S. 2025c. Assemble Your Crew: Automatic Multi-agent Communication Topology Design via Autoregressive Graph Generation. *CoRR*, abs/2507.18224.

Lin, X.; Cao, Y.; Sun, N.; Zou, L.; Zhou, C.; Zhang, P.; Zhang, S.; Zhang, G.; and Wu, J. 2025. Conformal Graph-level Out-of-distribution Detection with Adaptive Data Augmentation. In *WWW*, 4755–4765. ACM.

Lin, X.; Zhang, W.; Shi, F.; Zhou, C.; Zou, L.; Zhao, X.; Yin, D.; Pan, S.; and Cao, Y. 2024. Graph Neural Stochastic Diffusion for Estimating Uncertainty in Node Classification. In *ICML*. OpenReview.net.

Lin, Z.; Guo, L.; and Jia, C. 2024. Streaming Fair k-Center Clustering over Massive Dataset with Performance Guarantee. In *PAKDD (3)*, volume 14647 of *Lecture Notes in Computer Science*, 105–117. Springer.

Liu, Y.; Zhang, G.; Wang, K.; Li, S.; and Pan, S. 2025a. Graph-Augmented Large Language Model Agents: Current Progress and Future Prospects. *CoRR*, abs/2507.21407.

Liu, Z.; Chen, M.; Hua, Y.; Chen, Z.; Liu, Z.; Liang, L.; Chen, H.; and Zhang, W. 2024a. UniHR: Hierarchical Representation Learning for Unified Knowledge Graph Link Prediction. *CoRR*, abs/2411.07019.

Liu, Z.; Gan, C.; Wang, J.; Zhang, Y.; Bo, Z.; Sun, M.; Chen, H.; and Zhang, W. 2025b. OntoTune: Ontology-Driven Self-training for Aligning Large Language Models. In *WWW*, 119–133. ACM.

Liu, Z.; Wang, T.; Huai, M.; and Miao, C. 2024b. Backdoor Attacks via Machine Unlearning. In *AAAI*, 14115–14123. AAAI Press.

Liu, Z.; Ye, H.; Chen, C.; Zheng, Y.; and Lam, K. 2025c. Threats, Attacks, and Defenses in Machine Unlearning: A Survey. *IEEE Open J. Comput. Soc.*, 6: 413–425.

Murakonda, S. K.; Shokri, R.; and Theodorakopoulos, G. 2021. Quantifying the Privacy Risks of Learning High-Dimensional Graphical Models. In *AISTATS*, volume 130 of *Proceedings of Machine Learning Research*, 2287–2295. PMLR.

Neyman, J.; and Pearson, E. S. 1933. IX. On the problem of the most efficient tests of statistical hypotheses. *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character*, 231(694-706): 289–337.

Pan, J.; Liu, Y.; Zheng, X.; Zheng, Y.; Liew, A. W.; Li, F.; and Pan, S. 2025. A Label-free Heterophily-guided Approach for Unsupervised Graph Fraud Detection. In *AAAI*, 12443–12451. AAAI Press.

Said, A.; Derr, T.; Shabbir, M.; Abbas, W.; and Koutsoukos, X. D. 2023. A Survey of Graph Unlearning. *CoRR*, abs/2310.02164.

Sharma, K.; Lee, Y.; Nambi, S.; Salian, A.; Shah, S.; Kim, S.; and Kumar, S. 2024. A Survey of Graph Neural Networks for Social Recommender Systems. *ACM Comput. Surv.*, 56(10): 265.

Shchur, O.; Mumme, M.; Bojchevski, A.; and Günnemann, S. 2018. Pitfalls of Graph Neural Network Evaluation. *CoRR*, abs/1811.05868.

Shokri, R.; Stronati, M.; Song, C.; and Shmatikov, V. 2017. Membership Inference Attacks Against Machine Learning Models. In *IEEE Symposium on Security and Privacy*, 3–18. IEEE Computer Society.

Velickovic, P.; Cucurull, G.; Casanova, A.; Romero, A.; Liò, P.; and Bengio, Y. 2018. Graph Attention Networks. In *ICLR (Poster)*. OpenReview.net.

Vombatkere, K.; Mousavi, S.; Zannettou, S.; Roesner, F.; and Gummadi, K. P. 2024. TikTok and the Art of Personalization: Investigating Exploration and Exploitation on Social Media Feeds. In *WWW*, 3789–3797. ACM.

Wang, L.; He, D.; Zhang, H.; Liu, Y.; Wang, W.; Pan, S.; Jin, D.; and Chua, T. 2024a. GOODAT: Towards Test-Time Graph Out-of-Distribution Detection. In *AAAI*, 15537–15545. AAAI Press.

Wang, L.; Zheng, Y.; Jin, D.; Li, F.; Qiao, Y.; and Pan, S. 2024b. Contrastive Graph Similarity Networks. *ACM Trans. Web*, 18(2): 17:1–17:20.

Wang, Y.; Zhao, Y.; Wang, D. Z.; and Li, L. 2023. GALOPA: Graph Transport Learning with Optimal Plan Alignment. In *NeurIPS*.

Wang, Y.; Zhao, Y.; Wang, Z.; Li, L.; Wang, J.; Li, F.; Huang, M.; Pan, S.; and Wang, X. 2025a. Equivalence is All: A Unified View for Self-supervised Graph Learning. In *Forty-second International Conference on Machine Learning*.

Wang, Y.; Zhao, Y.; Wang, Z.; Shan, W.; Li, L.; Li, Q.; Huang, M.; Wang, M.; Pan, S.; and Wang, X. 2025b. N2GON: Neural Networks for Graph-of-Net with Position Awareness. In *Forty-second International Conference on Machine Learning*.

Wu, B.; Zhang, H.; Yang, X.; Wang, S.; Xue, M.; Pan, S.; and Yuan, X. 2024. GraphGuard: Detecting and Counteracting Training Data Misuse in Graph Neural Networks. In *NDSS*. The Internet Society.

Wu, F.; Long, Y.; Zhang, C.; and Li, B. 2022a. LINKTELLER: Recovering Private Edges from Graph Neural Networks via Influence Analysis. In *SP*, 2005–2024. IEEE.

Wu, J.; Yang, Y.; Qian, Y.; Sui, Y.; Wang, X.; and He, X. 2023. GIF: A General Graph Unlearning Strategy via Influence Function. In *WWW*, 651–661. ACM.

Wu, L.; Cui, P.; Pei, J.; and Zhao, L. 2022b. *Graph Neural Networks: Foundations, Frontiers, and Applications*. Singapore: Springer Singapore.

Xu, K.; Hu, W.; Leskovec, J.; and Jegelka, S. 2019. How Powerful are Graph Neural Networks? In *ICLR*. OpenReview.net.

Yao, J.; Chien, E.; Du, M.; Niu, X.; Wang, T.; Cheng, Z.; and Yue, X. 2024. Machine Unlearning of Pre-trained Large Language Models. In *ACL (1)*, 8403–8419. Association for Computational Linguistics.

Zhang, B.; Noorbakhsh, S. L.; Dong, Y.; Hong, Y.; and Wang, B. 2025a. Learning Robust and Privacy-Preserving Representations via Information Theory. In *AAAI*, 22363–22371. AAAI Press.

Zhang, H.; Wu, B.; Wang, S.; Yang, X.; Xue, M.; Pan, S.; and Yuan, X. 2023. Demystifying Uneven Vulnerability of Link Stealing Attacks against Graph Neural Networks. In *ICML*, volume 202 of *Proceedings of Machine Learning Research*, 41737–41752. PMLR.

Zhang, H.; Wu, B.; Yang, X.; Yuan, X.; Liu, X.; and Yi, X. 2025b. Dynamic Graph Unlearning: A General and Efficient Post-Processing Method via Gradient Transformation. In *WWW*, 931–944. ACM.

Zhang, H.; Wu, B.; Yuan, X.; Pan, S.; Tong, H.; and Pei, J. 2024a. Trustworthy Graph Neural Networks: Aspects, Methods, and Trends. *Proc. IEEE*, 112(2): 97–139.

Zhang, S.; Zhou, C.; Liu, Y.; Zhang, P.; Lin, X.; and Pan, S. 2025c. Conformal anomaly detection in event sequences. In *Forty-second International Conference on Machine Learning*.

Zhang, Y.; Zhao, Y.; Li, Z.; Cheng, X.; Wang, Y.; Kotevska, O.; Yu, P. S.; and Derr, T. 2024b. A Survey on Privacy in Graph Neural Networks: Attacks, Preservation, and Applications. *IEEE Trans. Knowl. Data Eng.*, 36(12): 7497–7515.

Zheng, X.; Huang, W.; Zhou, C.; Li, M.; and Pan, S. 2025. Test-Time Graph Neural Dataset Search With Generative Projection. In *Forty-second International Conference on Machine Learning*.