Prompt-based Unifying Inference Attack on Graph Neural Networks

Yuecen Wei[1,2], Xingcheng Fu[3], Lingyun Liu[3], Qingyun Sun[2], Hao Peng[2], Chunming Hu[1,2]*

[1]School of Software, Beihang University, Beijing, China
[2]Beijing Advanced Innovation Center for Big Data and Brain Computing, Beihang University, Beijing, China
[3]Key Lab of Education Blockchain and Intelligent Technology, Ministry of Education, Guangxi Normal University, China
{weiyc,sunqy,penghao,hucm}@buaa.edu.cn, fuxc@gxnu.edu.cn, 1295713045@stu.gxnu.edu.cn

Abstract

Graph neural networks (GNNs) provide important prospective insights in applications such as social behavior analysis and financial risk analysis based on their powerful learning capabilities on graph data. Nevertheless, GNNs' predictive performance relies on the quality of task-specific node labels, so it is common practice to improve the model's generalization ability in the downstream execution of decision-making tasks through pre-training. Graph prompting is a prudent choice but risky without taking measures to prevent data leakage. In other words, in high-risk decision scenarios, prompt learning can infer private information by accessing model parameters trained on private data (publishing model parameters in pre-training, i.e., without directly leaking the raw data, is a tacitly accepted trend). However, myriad graph inference attacks necessitate tailored module design and processing to enhance inference capabilities due to variations in supervision signals. In this paper, we propose a novel **Pro**mpt-based unifying **I**nference **A**ttack framework on GNNs, named **ProIA**. Specifically, ProIA retains the crucial topological information of the graph during pre-training, enhancing the background knowledge of the inference attack model. It then utilizes a unified prompt and introduces additional disentanglement factors in downstream attacks to adapt to task-relevant knowledge. Finally, extensive experiments show that ProIA enhances attack capabilities and demonstrates remarkable adaptability to various inference attacks.

Introduction

Real-world data benefits from the modeling of graph structures, which more effectively captures the inherent interconnected properties of the data, thereby enhancing the training process of models for Graph Neural Network (GNN) message passing (Kipf and Welling 2017; Velickovic et al. 2018; Zhang et al. 2021). In practical applications, besides conventional scenarios such as social recommendation (Sharma et al. 2024) and traffic prediction (Zhang et al. 2024), GNNs have also demonstrated outstanding performance in high-risk scenarios such as fraud detection (Innan et al. 2024) and disease prediction (Boll et al. 2024). Despite this, a multitude of existing GNNs rely heavily on supervised learning,

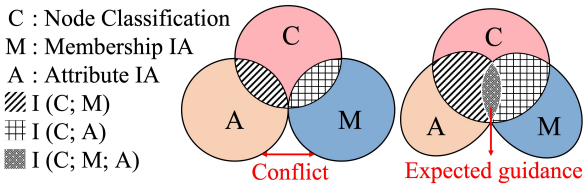


Figure 1: The Venn diagram depicts the respective optimization targets for the tasks with the intersections. $I(C; M)$: increases training-test data disparity, $I(C; A)$: improving data fitting, and $I(C; M; A)$: weakening model generalization and increasing model sensitivity to specific attributes.

and their representational capacity is significantly hampered by the challenges associated with obtaining high-quality labels for specific domains (Zhang, Zhang, and Yuan 2024; Liu et al. 2024c; Wu et al. 2022). Self-supervised learning has consistently achieved superior performance across an increasing range of tasks, enhancing the generalization ability of GNNs. Meanwhile, such graph learning models raise concerns about data security due to their ability to learn and extract knowledge. Many works in security have confirmed that pre-train models can lead to privacy leaks because of their inherent memory capabilities (Duan et al. 2024). For instance, in early question-answering systems, personal information utilized for model pre-training could be acquired through straightforward prompt-based queries (Carlini et al. 2021). Furthermore, even in modern systems equipped with security and ethical modules, private information can still be extracted by implementing jailbreak techniques (Li et al. 2023). Existing prompt attacks against pre-train models have only been defined within natural language processing (NLP) models. In contrast, attacks on graph data with non-Euclidean structures have yet to be discussed.

In the evolution of models, state-of-the-art GNNs (Zhiyao et al. 2024; Liu et al. 2023b; Yuan et al. 2024b) typically show positive, robust resistance and generalization capabilities when dealing with sparse and noisy data, thus effectively mitigating the issue of model overfitting. This results in privacy inference attacks that rely on model overfitting defects empirically failing to achieve satisfactory outcomes. However, even if the model avoids privacy leakage

*Corresponding author
Copyright © 2025, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

caused by overfitting, the adversary can still infer the target information by exploiting key and unique topological attributes of graph topology, such as scale-free and hierarchy (Carlini et al. 2021; Olatunji, Nejdl, and Khosla 2021; Wei et al. 2024). Nevertheless, existing privacy inference attack methods are usually designed to guide a single attack independently. The optimization targets for these classifications are different, especially when the adversary possesses multidimensional background knowledge, such as specific topologies. This makes it challenging to achieve satisfactory performance using uniformly guided attacks. As shown in Fig. 1, there is a conflict between simultaneously increasing the train-test distribution gap (achieving M) and improving data fitting capability (achieving A). In contrast, $I(C; M; A)$ illustrates the adversary's expected goal.

To investigate the capability of prompts in enhancing privacy inference attacks on graph data and to homogenize different attack tasks, it is non-trivial to address the following two challenges: (1) The vulnerability of graph structures means that models can easily crash when faced with harmful inputs or noise, thus revealing the attack. And the extent to which prompts can enhance graph data attacks remains unknown. Therefore, **it is vital to determine what kind of prompt queries can reveal private data while remaining inconspicuous to the target model**. (2) The most existing inference attacks necessitate the design of specific attack strategies. Integrating multifaceted information is challenging, as **extracting target information that matches downstream attack tasks to achieve adaptive inference attacks is difficult**.

Present work. To tackle these challenges, we propose a novel **Pro**mpt-based unifying **I**nference **A**ttack framework on GNNs, named **ProIA**, to enhance inference attacks and adaptive downstream attack tasks. Specifically, we design prompt queries for the target model to leak private information. Initially, we design information-theoretic principles during pre-training to guide the model in learning the essential structures of the graph, thereby enhancing the reachability of sensitive information. Subsequently, we construct prompt features through global-local contrastive learning to make the prompts appear as benign queries and make them unnoticeable to the target. Moreover, we design a disentanglement mechanism for the target model's output, allowing downstream tasks to infer the latent influence of corresponding variables from the enhanced posteriors provided by the prompt features, thereby adaptively achieving inference and further improving attack performance. Finally, extensive experiments validate the effectiveness of ProIA. Our contributions are summarized as:

- We explore the potential for prompt-induced graph data leakage and subsequently utilize feature embedding in graph pre-training to incorporate more useful structural information and camouflage malicious prompt features.
- Based on the prompt feature, we propose a novel **Pro**mpt-based unifying **I**nference **A**ttack framework on GNNs, named **ProIA**, that incorporates a disentanglement strategy to extract latent factors and adaptively guide various downstream attack tasks.
- We conduct extensive experiments on five public datasets, and the results demonstrate that ProIA exhibits superior privacy inference capabilities and has a disruptive effect on several common defense mechanisms.

Related Work

Graph Inference Attack

Inference attacks have evolved from deep learning models (Zarifzadeh, Liu, and Shokri 2024; Tramèr et al. 2022; Shokri et al. 2017) to GNNs (Dai et al. 2022; Zhang et al. 2023) by incorporating graph structures (Yuan et al. 2024a). Due to the strong scalability of inference attacks in most applications, current research primarily focuses on transforming data formats (from Euclidean to non-Euclidean). Duddu, Boutet, and Shejwalkar (2020) and Olatunji, Nejdl, and Khosla (2021) are the first to implement MIA on GNNs, incorporating graph structures into both target and shadow models, thereby establishing the pipeline for conducting MIA on graphs. Conti et al. (2022) utilized only label information to carry out MIA, reducing the adversary's need for background knowledge. Subsequently, Zhang et al. (2022) and Wu et al. (2021) expanded node-level inference attacks to the graph level. Wang and Wang (2022) analyzed the adversary's ability to infer attributes from a group perspective. Olatunji et al. (2023) proposed an iterative query strategy based on the feature propagation algorithm to enhance attribute inference attacks. Moreover, numerous defensive (Dai et al. 2023; Hu et al. 2022) efforts actively address adversaries with increasingly extensive background knowledge.

However, most existing works require the proposal of a particular framework to enhance a specific inference attack, with little attention given to unified guidance strategies for different inference attacks.

Prompt Attack on Language Models

Prompting (Brown et al. 2020; Liu et al. 2023a) initially emerged in the field of NLP as directives designed for specific downstream tasks. It can serve as a substitute for fine-tuning to assist downstream tasks in retrieving relevant knowledge from pre-trained models. Privacy leakage during pre-training is typically associated with overfitting (Zhang et al. 2017; Deng et al. 2024). The erroneous association between overfitting and memorization has led many to believe that advanced models trained on large-scale data would not disclose information from their training data (Carlini et al. 2021). However, existing work has demonstrated the fallacy of this belief. They focus on enhancing the semantic information of structured prompt texts (Liu et al. 2024a; Yao et al. 2024; Niu et al. 2024), utilizing adversarial attacks to bypass Language Models (LMs) defense mechanisms (Li et al. 2024; Shi et al. 2024), and designing gradient-enhanced strategies to generate efficient prompts (Liu et al. 2024b).

However, the existing prompt designs for attacks focus on structured discrete data, making it challenging to adapt to the graph data structure. Moreover, graph prompts are an excellent alternative, yet they are not directly applicable to the pipeline of inference attacks.

Preliminary

Notations

We implement attribute inference attacks (AIA) and membership inference attacks (MIA) in GNNs with continuous and discrete attributes, respectively. Given a graph $G = (V, E, \mathbf{X})$ with node set $V = \{v_1, \ldots, v_N\}$, edge set $E \in V \times V$ and node attribute matrix $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$, where $\mathbf{x}_i \in \mathbb{R}^d$ denote the feature vector of node $v_i \in V$. Let $\mathbf{A}_{ij}$ denotes the adjacency matrix and $\mathbf{A}_{ij} \in \{0, 1\}$ represents the connection between node v_i and node v_j. For inference attack (IA), we focus on the node classification task, given the adversary's known labeled node set V^{tr} and their labels $\mathcal{Y}^{tr}$. We aim to train a node classifier $\mathcal{F}_\theta$ that distinguishes in AIA and MIA to predict the sensitive labels $\mathcal{Y}^{te}$ of remaining unlabeled nodes $V^{te} = V \setminus V^{tr}$.

Inference Attack

Attribute inference attack. The objective of attribute inference attacks is to train an attack model $\mathcal{F}_A$ by retraining the prediction results $\mathbf{h}_T$ of the target model $\mathcal{F}_T$ for specified queries $\mathbf{p}$, thereby inferring the sensitive attributes $\hat{\mathbf{x}}$ of nodes within the target training set G_T^{tr}. In real-world scenarios, such sensitive attributes might include social network users' gender, age, occupation, etc. Then, the attack model can be trained in the attack dataset G_A by

$$\min_{\theta_A} \frac{1}{|G_A|} \sum_{G_i \in G_A} \mathcal{L}\left(\mathcal{F}_A\left(\mathbf{x}_A^{tr}\right)_i^{G_i}, \mathbf{x}_i^{te}\right), \tag{1}$$

where $\mathbf{x}_A^{tr} = \mathbf{h}_T^{\mathbf{p}} = \mathcal{F}_T(\mathbf{p}^{G_i})$ is the train-set embedding of G_i from target model $\mathcal{F}_T$ in query $\mathbf{p}$, $\mathbf{x}^{te}$ is test set, G_i is k-hop subgraph centered at node v_i, and $\mathcal{L}(\cdot)$ can be MSE loss or cross-entropy loss for different types of attributes.

Membership inference attack. The objective is to infer whether a target individual was included in the training dataset of a target model $\mathcal{F}_A$, i.e., whether $\mathbf{x} \in G_T^{tr}$ or not, implying that MIA will lead to privacy leakage when $\mathcal{F}_A$ is trained on high-risk data.

MIA consists of a target model $\mathcal{F}_T$, a shadow model $\mathcal{F}_S$, and an attack model $\mathcal{F}_A$, distinguishing it from AIA by an additional shadow model. Specifically, the function of $\mathcal{F}_A$ remains consistent with that in AIA, but the output of the target model $\mathbf{h}_T^{tr}$ does not directly guide $\mathcal{F}_A$. Instead, it is utilized to instruct $\mathcal{F}_S$ to learn to generate the attack dataset G_A by mimicking the prediction behavior of $\mathcal{F}_T$, and its optimization objective is illustrated as:

$$\min_{\theta_S} \frac{1}{|G_S^{tr}|} \sum_{G_i \in G_S^{tr}} \mathcal{L}\left(\mathcal{F}_S\left(G_i\right), \mathcal{F}_T\left(G_i\right)\right), \tag{2}$$

where $\mathcal{F}(G_i)$ denotes the predicted label distribution of G_i. Finally, the attack can be implemented using a Multilayer Perceptron (MLP). It is noteworthy that MIA is a binary classification task and typically relies on the assumption that $\mathcal{F}_T$ overfits to achieve the attack. However, most competitive models are designed to mitigate this issue effectively.

Prompt attack. Prompt attacks macroscopically indicate that an adversary constructs specific inputs $\mathbf{p}$ to the target model $\mathcal{F}_T$, driving the model to search its memory θ_T for information related to the prompts and feed it back to the adversary $\mathcal{F}_A$. Specifically, the prompt attack (Dong et al. 2023) in NLP aims to bypass the security mechanisms of language models, potentially causing them to reveal private information like phone numbers and addresses. This paper then defines a similar concept called a "prompt-based inference attack", which adheres to the aforementioned attack principles. It utilizes message passing in GNNs to establish unique connections between users. Even erased individual information can be extracted through prompting inference.

Graph Prompting

Graph prompts are designed to rapidly adapt to downstream tasks by employing pre-training followed by prompts $\mathbf{p}$ for knowledge transfer. Specifically, to effectively extract task-specific prior knowledge from the pre-trained model, the node representation in the subgraph is generated using the ReadOut method as:

$$\mathbf{h}_{query} = \textsc{ReadOut}\left\{\mathbf{p} \odot \mathbf{h}\right\}, \tag{3}$$

where $\mathbf{h}$ is the node embeddings and $\odot$ denotes the element-wise multiplication. Additionally, node representations are fine-tuned during training to aid different downstream tasks. For different attacks, the core features of inference vary. For instance, AIA tends to focus on the expression of node attributes, whereas MIA emphasizes the overall data distribution.

Prompt-based Inference Attack

In this section, we will gradually introduce the detailed design of ProIA, including the overall framework, the pre-training prompts, attack data generation, and the adaptive inference attack. See the appendix for ProIA's algorithm.

Overview of ProIA

The overall framework of ProIA is illustrated in Fig. 2. It consists of three components: pre-training, data generation, and prompt attack. (1) Pre-training process, which aims to learn the critical topological structures and obscure malicious intents by contrasting local G_{lo} and global G_{gl} information, integrating rich information into the model to enhance its memory and attribute fitting capabilities. (2) Attack data generation, where $\mathbf{p}$ obtained from the pre-trained model readout serves as prompt queries, forcing $\mathcal{F}_T$ to output information beneficial to the attack, which is then used as feature input for $\mathcal{F}_A$. For $\mathcal{F}_S$, it specifically refers to the module used in MIA to mimic $\mathcal{F}_T$'s inference and generate attack dataset supervision signals. (3) Downstream inference attack, where an introduced disentangled representation module is used to further extract topology connections in $\mathbf{p}$ closely related to downstream tasks, enhancing information passed to guide adaptation to different attack tasks.

Pre-training for Leaking Information

ProIA attempts to approach the conflict mentioned above from a unified perspective, avoiding direct bias towards either task. This aims to establish a more feasible influence

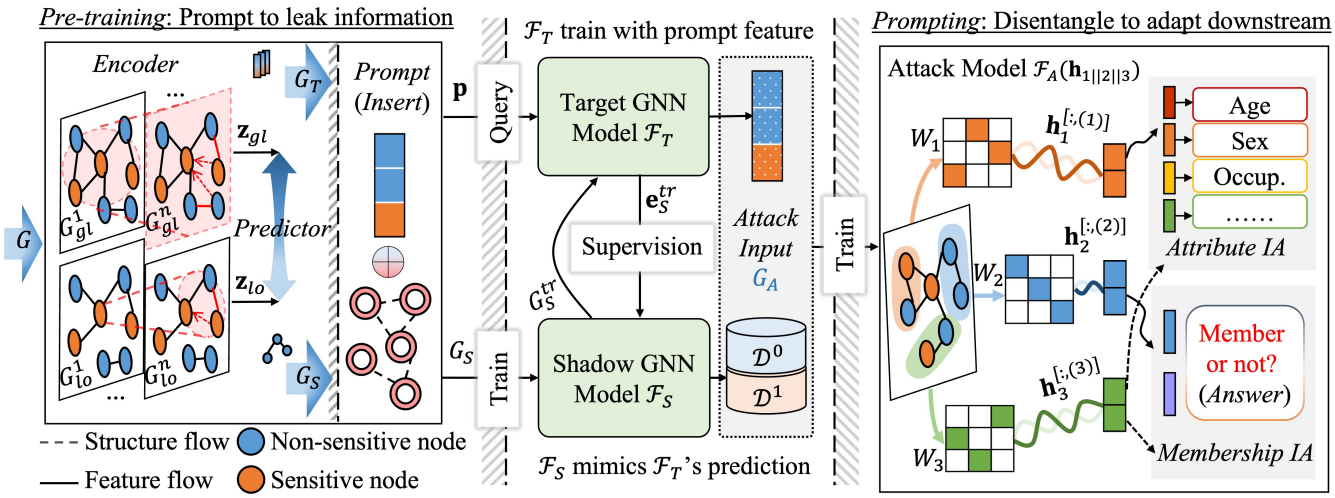


Figure 2: The overall framework of ProIA.

for the propagation of sensitive attributes and enhance the robustness of the training data.

Sample generation. We first quantify the similarity between nodes and their neighbors, which serves as a metric for initializing the construction of local subgraphs G_{lo}. Specifically, we set a threshold hyperparameter t to control the connectivity of edges in the sampled local subgraphs, i.e., let $\mathbf{s}(\mathbf{x}_i, \mathbf{x}_j)$ denote the Jaccard similarity score between node i and node j, if $\mathbf{s}(\mathbf{x}_i, \mathbf{x}_j) \geq t$, then there exists an edge between u and v ($\mathbf{A}_{i,j} = 1$). Otherwise, there is no edge ($\mathbf{A}_{i,j} = 0$). Subsequently, the subgraph is augmented by adding edges through random Bernoulli sampling. Therefore, the positive sample is denoted as $\mathbf{x}$, and the negative sample $\hat{\mathbf{x}}$ is generated by randomly permuting the node features of G_{gl}. In the joint learning process with the contrastive samples, G_{lo} enhances the model training's robustness against perturbations through reweighting.

Information bottleneck constraints. As mentioned above, a pre-trained model that captures the key structures can provide diversity for information extraction. Therefore, we utilize information-theoretic techniques to guide the aggregation of critical information, i.e., the information bottleneck (IB) principle (Wu et al. 2020).

Let a local subgraph be composed of node v_i and its k-hop neighbors containing the relevant data, assuming independence from the rest of the graph. Consequently, the IB constraint can approximate the globally optimal representation space within an iteratively refined Markov chain. Furthermore, to approximate the solution of $I(C, M, A)$ in Fig. 1, we need to optimize by:

$$\begin{aligned} &\min_{\theta} -I(\mathbf{Z}; \mathbf{Y}) + \beta_{A,M} I(G, \mathbf{Z}) \\ =&\min_{\theta} -I(\mathbf{Z}_{\mathbf{X}}; \mathbf{Y}^C) + \beta_{A,M} I(G; \mathbf{Z}_{At}, \mathbf{Z}_{Me}), \end{aligned} \tag{4}$$

where $\{\mathbf{Z}_{At}, \mathbf{Z}_{Me}\} \in \mathbf{Z}_{\mathbf{X}}$ denote the representations that facilitate attribute fitting and member information propagation, $I(\cdot)$ represents mutual information, β is the Lagrangian parameter to balance the terms, and $\mathbf{Y}^C$ is the label of node classification. According to those above, after instantiating the IB in the Markov chain, we only need to optimize distributions $\mathbb{P}(\mathbf{Z}^l_{At}|\mathbf{Z}^{l-1}_{At}, \mathbf{Z}^l_{Me})$ and $\mathbb{P}(\mathbf{Z}^l_{Me}|\mathbf{Z}^{l-1}_{At}, \mathbf{A}), l \in L$ to capture the local dependencies between nodes for easier facilitating parameterization and optimization.

To achieve more precise attainment of the final optimization objective through local optimization, we introduce variational boundaries to optimize the model (Alemi et al. 2016). The lower bound of $I(\mathbf{Z}_{\mathbf{X}}; \mathbf{Y}^C)$ and the upper bound of $I(G; \mathbf{Z}_{At}, \mathbf{Z}_{Me})$ are inspired by Wu et al. (2020).

Proposition 1 *(The lower bound of $I(\mathbf{Z}_{\mathbf{X}}; \mathbf{Y}^C)$). For any variational distributions $\mathbb{Q}_1\left(\mathbf{Y}^C \mid \mathbf{Z}^L_{\mathbf{X}}\right)$ and $\mathbb{Q}_2(\mathbf{Y}^C)$:*

$$\begin{aligned} I\left(\mathbf{Y}^C; \mathbf{Z}^L_{\mathbf{X}}\right) \geq 1 + \mathbb{E}&\left[\log \frac{\mathbb{Q}_1\left(\mathbf{Y}^C \mid \mathbf{Z}^L_{\mathbf{X}}\right)}{\mathbb{Q}_2(\mathbf{Y}^C)}\right] \\ -\mathbb{E}_{\mathbb{P}(\mathbf{Y}^C)\mathbb{P}(\mathbf{Z}^L_{\mathbf{X}})}&\left[\frac{\mathbb{Q}_1\left(\mathbf{Y}^C \mid \mathbf{Z}^L_{\mathbf{X}}\right)}{\mathbb{Q}_2(\mathbf{Y}^C)}\right] \end{aligned} \tag{5}$$

Proposition 2 *(The upper bound of $I(G; \mathbf{Z}_{At}, \mathbf{Z}_{Me})$). Two randomly selected indices $\mathcal{I}_a, \mathcal{I}_m \subset L$, are independent ($G \perp \mathbf{Z}^L_{At} \mid \left\{\mathbf{Z}^{l_a}_{At}\right\} \cup \left\{\mathbf{Z}^{l_m}_{Me}\right\}$) based on the aforementioned Markov chain, where $l_a \in \mathcal{I}_a$, $l_m \in \mathcal{I}_m$. Then for any variational distributions $\mathbb{Q}\left(\mathbf{Z}^l_{At}\right)$ and $\mathbb{Q}\left(\mathbf{Z}^l_{Me}\right)$:*

$$\begin{aligned} I(G; \mathbf{Z}_{At}, \mathbf{Z}_{Me}) &\leq I\left(G; \left\{\mathbf{Z}^{l_a}_{At}\right\} \cup \left\{\mathbf{Z}^{l_m}_{Me}\right\}\right) \\ &\leq \sum_{l \in \mathcal{I}_a} \mathcal{T}^l_{At} + \sum_{l \in \mathcal{I}_m} \mathcal{T}^l_{Me}, \end{aligned} \tag{6}$$

$$\text{where } \mathcal{T}^l_{At} = \mathbb{E}\left[\log \frac{\mathbb{P}\left(\mathbf{Z}^l_{At} \mid \mathbf{Z}^{l-1}_{At}, \mathbf{Z}^l_{Me}\right)}{\mathbb{Q}\left(\mathbf{Z}^l_{At}\right)}\right], \tag{7}$$

$$\mathcal{T}^l_{Me} = \mathbb{E}\left[\log \frac{\mathbb{P}\left(\mathbf{Z}^l_{Me} \mid \mathbf{A}, \mathbf{Z}^{l-1}_{At}\right)}{\mathbb{Q}\left(\mathbf{Z}^l_{Me}\right)}\right]. \tag{8}$$

The proofs are provided in the appendix.

To estimate $\mathcal{T}_{At}^l$, let $\mathbb{Q}\left(\mathbf{Z}_{At}^l\right)$ be a Gaussians distribution $q(\mathbf{Z}_{At})$, $p(\mathbf{Z}_{At}, \mathbf{Z}_{Me})$ uses the sampled by $\mathbf{Z}_{At}$:

$$\mathcal{L}_A = \sum_{v \in V} \mathrm{KL}\left[(p(\mathbf{Z}_{At,v}, \mathbf{Z}_{Me,v} | \mathbf{Z}_{\mathbf{x}}) \| p(\mathbf{Z}_{At,v}))\right]. \quad (9)$$

Similarly, the estimation of $\mathcal{T}_{Me}^l$:

$$\mathcal{L}_M = \sum_{v \in V} \mathrm{KL}\left[(p(\mathbf{Z}_{Me,v}, \mathbf{Z}_{At,v} | \mathbf{Z}_{\mathbf{x}}) \| p(\mathbf{Z}_{Me,v}))\right]. \quad (10)$$

Then, the objective function of IB is:

$$\mathcal{L}_{IB} = -\sum_{v \in V} \mathrm{CE}\big(f(\mathbf{Z}_{\mathbf{X},v}^L); \mathbf{Y}^C\big) + \beta_A \mathcal{L}_A + \beta_M \mathcal{L}_M, \quad (11)$$

where $f(\cdot)$ is a classifier, CE represents the cross-entropy loss about the upper bound.

Contrastive training. We employ a graph convolutional network as the encoder, resulting in the node representation $\mathbf{z}$ with IB. For the representations G_{lo} and G_{gl} denoted as $\mathbf{z}_{lo}$ and $\mathbf{z}_{gl}$, to capture the intricate interactions between the features of the local and global representation spaces, we obtain a unified representation via a Bilinear transformation layer:

$$\mathcal{P} = \sigma(\mathbf{z}_{lo1}^T)\mathbf{W}\mathbf{z}_{gl} + \mathbf{b}, \quad (12)$$

$$\hat{\mathcal{P}} = \sigma(\mathbf{z}_{lo2}^T)\mathbf{W}\hat{\mathbf{z}}_{gl} + \mathbf{b}, \quad (13)$$

where $\mathbf{W}$ is a learned weight matrix and $\mathbf{b}$ represents a learned bias vector. $\sigma(\cdot)$ is a nonlinear activation function. The model parameter ϕ is trained to maximize the mutual information between the global and local representations. Consequently, the objective function of the binary cross-entropy loss is expressed as:

$$\mathcal{L}_{CL} = -\frac{1}{N}\sum_{i=1}^{N}\left[\log(\mathcal{P}) + \log(1 - \hat{\mathcal{P}})\right]. \quad (14)$$

Finally, the loss function for the entire pre-training is:

$$\mathcal{L}_1 = \alpha \mathcal{L}_{CL} + (1 - \alpha)\mathcal{L}_{IB}, \quad (15)$$

where α is a trade-off hyperparameter.

Attack Data Generation

We aim to design a unified prompt feature to enhance attack capabilities. The attack processes for both attribute inference and membership inference follow established methodologies from previous research, and the manifestation forms of the prompt features are shown below.

Attribute Inference. The training set G_A^{tr} for $\mathcal{F}_A$ is derived from the adversary's background knowledge, while the test set G_A^{te} is obtained from the output of $\mathcal{F}_T$. Specifically, given the known parameters ϕ of the pre-trained model, the adversary designs prompt features $\mathbf{p}$ for $\mathcal{F}_T$ based on Eq. (3) and initiates queries to the model. The output posteriors $\mathbf{h}$ are then used as input for $\mathcal{F}_A$.

Membership Inference. In addition to $\mathcal{F}_T$, extra shadow models $\mathcal{F}_S$ in MIA are utilized to construct the training set G_A^{tr} used for training $\mathcal{F}_A$. Specifically, the adversary leverages background data $G_S(\mathbf{p})$ to train a model $\mathcal{F}_S$ that has a similar architecture to $\mathcal{F}_T$. The node representations are obtained from $\mathcal{F}_S$, with G_S^{tr} labeled as 1 and G_S^{te} labeled as 0, thereby forming $G_A^{tr} = \mathbf{h}$. Furthermore, $G_A^{te} = G_T$.

Prompting for Adapting Inference Attack with Disentanglement

From the previous phase, AIA obtains a vector matrix about the node representation optimized on the attack model, and MIA gets an extra set of datasets to train the attack model. Then ProIA proposes a task-specific learnable prompt method for more effective knowledge transfer.

Prompt disentanglement. To ensure that $\mathcal{F}_A$ obtains guidance information that is more directly relevant to downstream tasks during inference, ProIA integrate a disentanglement mechanism into $\mathcal{F}_A$ to isolate key factors affecting the structure in prompt features. Specifically, in AIA, nodes are influenced by various attributes and establish potential connections. We aim to set up k fixed virtual channels for the inter-attribute connections and learn the disentangled representation of latent factors. First, prompt features are pre-corrected for offsets through transformation and then mapped into channels, forming neighbors based on single/-multiple attributes:

$$\mathbf{z}_{i,k} = \frac{\sigma(\mathbf{W}_k^T(\mathrm{MLP}(\mathbf{p}_i) \odot \mathbf{h}_i) + \mathbf{b}_k)}{\left\|\sigma(\mathbf{W}_k^T(\mathrm{MLP}(\mathbf{p}_i) \odot \mathbf{h}_i) + \mathbf{b}_k)\right\|_2}, \quad (16)$$

where $\mathbf{W}_k$ and $\mathbf{b}_k$ are the parameters of channel k. After the initialization above, the similarity weights between nodes i and their neighbors j are calculated within each channel, forming the edge probability between nodes and reweighting the nodes:

$$\hat{\mathbf{z}}_{j,k} = \mathbf{z}_{j,k} \odot \sum_{j:(i,j) \in G_A} \textsc{Softmax}(\mathbf{z}_{j,k}^T \mathbf{z}_{i,k}^{(t-1)})/\tau, \quad (17)$$

where τ is a hyperparameter that regulates the tightness of the cluster centers. Subsequently, the latent factor representations within the channel are updated to the nodes, resulting in disentangled representations:

$$\mathbf{d}_k^t = \frac{\mathbf{z}_{i,k} + \hat{\mathbf{z}}_{j,k}}{\left\|\mathbf{z}_{i,k} + \hat{\mathbf{z}}_{j,k}\right\|_2}, \quad (18)$$

where t is a hyperparameter for the number of iterations in Eq. (17) and (18), aiming to search for the largest cluster in each subspace iteratively. Each neighbor is approximated to belong to only one subspace cluster.

Attack implementation. After multiple iterations, a multi-level disentangled representation $\mathbf{d}$ is formed. This representation, once normalized, is used as prior guidance for training $\mathcal{F}_A$ in the MLP. The objective function is as follows:

$$\mathcal{L}_2 = \frac{1}{N}\sum_{i=1}^{N}\left(-\log p_i^A - \log q_i^d\right) + \lambda \mathcal{D}_{\mathrm{KL}}(\mathbb{P}^A || \mathbb{Q}^d), \quad (19)$$

where $p_i^A = \mathcal{F}_A(\tilde{\mathbf{d}}_i \mathbf{h}_i)$ and $q_i^d = \textsc{Softmax}(\mathbf{W}^T \mathbf{d}_i + b)$, their KL divergence is employed further to constrain the impact of disentanglement factors on attack inference.

Experiment

In this section, we conduct extensive experiments on five real-world datasets to validate the attacking capability of ProIA[1]. We introduce the experimental setup, present the results, and provide a detailed analysis.

[1]Code available: https://github.com/RingBDStack/ProIA

	Model	Cora		Facebook		Lastfm		Imp.	Bail		Pokec-n		Imp.
		ACC	F1	ACC	F1	ACC	F1	Avg.	ACC	F1	ACC	F1	Avg.
Vanilla	GCN	82.03	81.60	59.02	58.91	61.84	60.40		51.81	51.80	70.76	59.55	
	GAT	85.88	85.70	60.72	55.17	64.84	64.11	-	50.61	34.01	71.24	59.27	-
	SAGE	73.77	73.94	60.01	45.22	61.51	51.73		52.96	52.40	70.86	60.10	
Pre-tr.	GCN	56.14	56.17	55.22	54.25	42.55	41.72	-16.29	55.14	**53.61**	70.71	58.57	↑1.4
	STABLE	63.10	57.45	61.58	61.44	67.13	64.56	-4.76	54.29	53.39	NA	NA	-4.63
	ProIA$_{\mathbf{p}}$	89.47	89.54	62.88	62.81	70.40	68.63	↑6.66	54.41	53.31	70.51	59.02	↑ 0.85
Prompt	**ProIA$_{\mathbf{d}}$**	81.42	81.09	61.42	61.34	65.91	64.77	↑2.03	52.84	51.79	70.69	58.56	↑0.01
	GCN	80.95	80.68	53.80	51.94	43.15	43.28	-8.33	53.67	52.23	72.93	68.75	↑3.43
	STABLE	62.80	56.56	61.09	60.92	66.56	63.32	-5.43	54.10	53.43	NA	NA	-4.70
ProIA	**GCN**	90.18	90.01	**62.89**	**62.85**	70.45	68.61		**55.19**	52.81	74.85	70.48	
	GAT	**97.62**	**97.60**	60.99	60.72	66.09	59.83	↑7.74	52.03	44.37	**78.44**	**77.06**	↑5.60
	SAGE	90.62	90.63	53.48	51.79	**80.05**	**71.28**		52.60	48.35	74.92	71.36	

Table 1: Summary results of accuracy, Weighted-F1 and average improvement performance. (NA represents memory limitation)

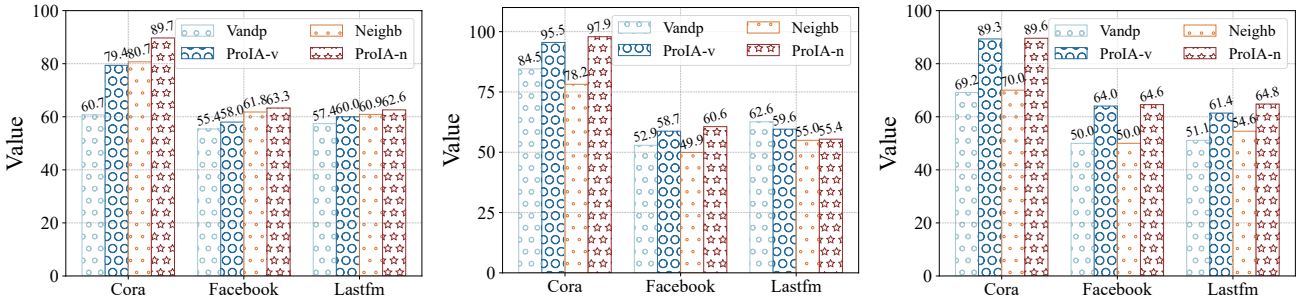


Figure 3: Attack AUC-ROC scores of GCN, GAT, and SAGE (from left to right) against defended models for MIA.

Experimental Settings

Datasets. *For the MIA*, we employ three widely-used graph datasets. **Cora** (Kipf and Welling 2017) is a citation network of academic papers. **Facebook** (Leskovec and Mcauley 2012) describes the social network of relationships between social media pages. **Lastfm** (Rozemberczki and Sarkar 2020) is a social network reflecting users' musical interests. *For the AIA*, we utilize two datasets that are labeled with sensitive attributes. **Bail** (Agarwal, Lakkaraju, and Zitnik 2021) is a U.S. bail dataset. **Pokec-n** (Takac and Zabovsky 2012) is a social network extracted from the Slovakian social network Pokec. See the appendix for more details.

Baselines. We established three backbone models (GCN, GAT, and SAGE) to test the expressiveness of ProIA in different scenarios. **STABLE** (Li et al. 2022), an additional model, shows ProIA's utility in extracting hints from robust models using a contrastive learning framework. Furthermore, we validated ProIA's capability in defense models (**Vandp** (Olatunji, Nejdl, and Khosla 2021), **Neigbh** (Olatunji, Nejdl, and Khosla 2021), **PPGL** (Hu et al. 2022)). See the appendix for more details.

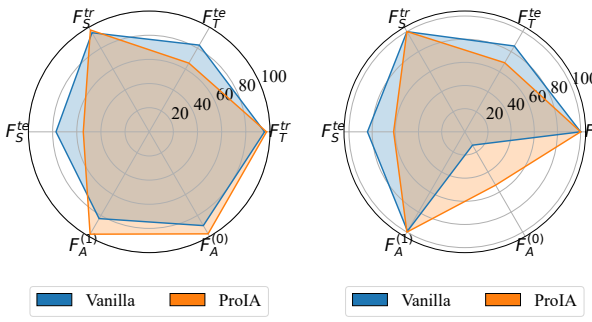
Based on the backbones and attack backgrounds, the methods are divided into four categories: (1) **Vanilla** represents models used by both $\mathcal{F}_T$ and $\mathcal{F}_S$, with $\mathcal{F}_A$ being an MLP. (2) **Pre-tr.** refers to obtaining hint features through pre-training on different models, applied to the training of $\mathcal{F}_T$ or $\mathcal{F}_S$ with GCN as the backbone, using an MLP as $\mathcal{F}_A$. (3) **Prompt** introduces a disengagement mechanism for $\mathcal{F}_A$ in pre-training. (4) **ProIA** demonstrates the proposed method's attack capability on targets in different scenarios.

Settings. To maximize its advantages, we set the number of baseline layers to 2 and the disentangled mechanism layers in ProIA to 5. The attack model's learning rate and iteration number are set to 0.01 and 100, and the remaining modules and methods are set to $1e-4$ and 200. Common parameters include a representation dimension of 256 and the Adam optimizer. The privacy-preserving model employs a unified privacy budget of 0.2. All other settings use default optimal values. See the appendix for more details.

Performance Evaluation

Based on the categorization above, we uniformly conducted comprehensive performance verification and ablation experiments for ProIA. We then compared the inference capability of ProIA against commonly used privacy protection models. Finally, we design the case study to demonstrate the information extraction ability of ProIA during the pre-training and the downstream attack.

Overall performance and ablation study. As shown in



(a) Cora on GAT. (b) Cora on Sage.

Figure 4: Case study.

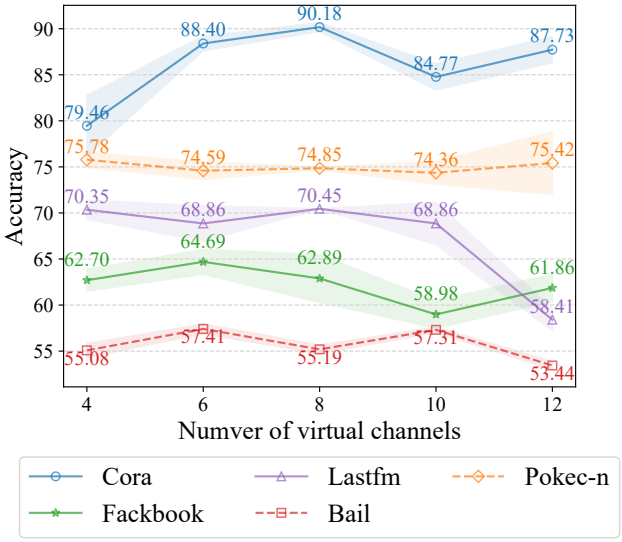


Figure 5: Hyperparameter analysis.

Tab. 1, we employed multiple metrics and devised various method variants. The average increase in ProIA's attack accuracy indicates superior performance in both MIA and AIA. Specifically, compared to Vanilla, ProIA achieved a maximum accuracy improvement of 18.54% for Lastfm in SAGE. Although ProIA is slightly disadvantaged under a few backbones, it maintains a substantial advantage by flexibly adapting the attack scenarios.

For the ablation studies involving Pre-tr and Prompt, our method ProIA$_{\mathbf{p}}$ represents removing the disentanglement mechanism, and ProIA$_{\mathbf{d}}$ indicates that prompt features are not used as background knowledge, as shown in Tab. 1 and appendix. Both approaches show varying degrees of attack improvement compared to Vanilla, with single modules sometimes achieving sub-optimal results. This demonstrates their effectiveness and that not all prompt features used as queries can assist in the attack. Furthermore, data issues like label bias and noise within the dataset may understandably lead to varying module performance. Overall, ProIA successfully extracts critical information for inference attacks and provides adaptive guidance for downstream attacks.

Attack performance in defended models. To ensure fairness in model settings, we evaluated the impact of inserting defense methods into Vanilla and ProIA on attacks against the protected target model, with the results illustrated in Fig. 3 and appendix. For MIA, we applied the Vandp and Neighb, denoted as ProIA-v and ProIA-n when inserted into our method. For AIA, we introduced the PPGL, similarly represented as ProIA-p in ProIA. It is evident that most ProIA variants disrupt existing defense mechanisms to varying extents. In particular, MIA achieves an increase in attack AUC-ROC of up to 18.7% on the Cora dataset using GCN. Notably, ProIA's AIA demonstrates exceptional attacking power on the sparsely distributed Pokec-n dataset, while Vanilla can only maintain random guess results.

Case study. We aim to answer: Does ProIA successfully obfuscate itself so the target model is unaware during training? We utilized a radar chart, as illustrated in Fig. 4, to comprehensively demonstrate the impact of ProIA on $\mathcal{F}_T$ during MIA. ProIA distorts the originally even distribution of the polygon into a conical shape, revealing the differences in training accuracy ($\mathcal{F}_T^{tr}$, $\mathcal{F}_S^{tr}$) and testing accuracy ($\mathcal{F}_T^{te}$, $\mathcal{S}_T^{te}$) of $\mathcal{F}_T$ and $\mathcal{F}_S$ under attack. This indicates that ProIA obfuscates the target's training process and increases the target's overfitting. Additionally, we use $\mathcal{F}_A^{(0)}$ and $\mathcal{F}_A^{(1)}$ to denote the true negative and true positive of $\mathcal{F}_A$'s binary classification. Notably, ProIA decreases the misclassification rate.

Hyperparameter analysis. We conduct the hyperparameter analysis for virtual channels k in the disentanglement mechanism to verify the role of latent factors as key elements affecting the isolation of prompt feature structures. The results are shown in Fig. 5, indicating that the sensitivity to attack accuracy when setting k varies across different datasets. The Cora has higher feature dimensions, and the Facebook has a more complex topology. During learning latent factors, the decision boundaries are affected by more interference sources, necessitating attention to the setting and adjustment of channel numbers. In contrast, the attack accuracy of other datasets is less affected. In summary, the hyperparameter settings can be further optimized according to the characteristics of different datasets to improve the attack model's overall performance.

Conclusion

In this paper, we present a novel framework named ProIA to explore the role of prompts in inference attacks and enhance the adaptiveness of such attacks under unified prompt guidance. ProIA initially extracts critical topological information from graph data during pre-training while concealing the malicious intent of prompt features. Subsequently, it generates posterior supervised attack data by querying target and shadow models with prompts. Finally, ProIA improves the localization of prompt knowledge through a disentanglement mechanism in downstream tasks. Extensive experiments have demonstrated that ProIA exhibits superior inference attack capabilities and adaptability. Future optimizations will focus on reducing the computational overhead in pre-training structures.

Acknowledgments

The corresponding author is Chunming Hu. This paper is supported by the STI 2030-Major Projects under Grant No. 2022ZD0120203 and the National Natural Science Foundation of China through Grant No. U21A20474, No. 62462007, and No. 62302023. We owe sincere thanks to all authors for their valuable efforts and contributions.

References

Agarwal, C.; Lakkaraju, H.; and Zitnik, M. 2021. Towards a unified framework for fair and stable graph representation learning. In *Uncertainty in Artificial Intelligence*, 2114–2124.

Alemi, A. A.; Fischer, I.; Dillon, J. V.; and Murphy, K. 2016. Deep variational information bottleneck. arXiv:1612.00410.

Boll, H. O.; Amirahmadi, A.; Ghazani, M. M.; de Morais, W. O.; de Freitas, E. P.; Soliman, A.; Etminani, K.; Byttner, S.; and Recamonde-Mendoza, M. 2024. Graph neural networks for clinical risk prediction based on electronic health records: A survey. *Journal of Biomedical Informatics*, 104616.

Brown, T.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J. D.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. 2020. Language models are few-shot learners. *NeurIPS*, 1877–1901.

Carlini, N.; Tramèr, F.; Wallace, E.; Jagielski, M.; Herbert-Voss, A.; Lee, K.; Roberts, A.; Brown, T. B.; Song, D.; Erlingsson, Ú.; Oprea, A.; and Raffel, C. 2021. Extracting Training Data from Large Language Models. In *USENIX Security*, 2633–2650.

Conti, M.; Li, J.; Picek, S.; and Xu, J. 2022. Label-only membership inference attack against node-level graph neural networks. In *Proceedings of the 15th ACM Workshop on Artificial Intelligence and Security*, 1–12.

Dai, E.; Cui, L.; Wang, Z.; Tang, X.; Wang, Y.; Cheng, M.; Yin, B.; and Wang, S. 2023. A unified framework of graph information bottleneck for robustness and membership privacy. In *KDD*, 368–379.

Dai, E.; Zhao, T.; Zhu, H.; Xu, J.; Guo, Z.; Liu, H.; Tang, J.; and Wang, S. 2022. A comprehensive survey on trustworthy graph neural networks: Privacy, robustness, fairness, and explainability. arXiv:2204.08570.

Deng, Y.; Zhang, W.; Pan, S. J.; and Bing, L. 2024. Multilingual jailbreak challenges in large language models. In *ICLR*.

Dong, X.; He, Y.; Zhu, Z.; and Caverlee, J. 2023. PromptAttack: Probing Dialogue State Trackers with Adversarial Prompts. In *ACL (Findings)*, 10651–10666.

Duan, M.; Suri, A.; Mireshghallah, N.; Min, S.; Shi, W.; Zettlemoyer, L.; Tsvetkov, Y.; Choi, Y.; Evans, D.; and Hajishirzi, H. 2024. Do membership inference attacks work on large language models? arXiv:2402.07841.

Duddu, V.; Boutet, A.; and Shejwalkar, V. 2020. Quantifying privacy leakage in graph embedding. In *MobiQuitous*, 76–85.

Hu, H.; Cheng, L.; Vap, J. P.; and Borowczak, M. 2022. Learning privacy-preserving graph convolutional network with partially observed sensitive attributes. In *WWW*, 3552–3561.

Innan, N.; Sawaika, A.; Dhor, A.; Dutta, S.; Thota, S.; Gokal, H.; Patel, N.; Khan, M. A.-Z.; Theodonis, I.; and Bennai, M. 2024. Financial fraud detection using quantum graph neural networks. *Quantum Machine Intelligence*, 7.

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

Leskovec, J.; and Mcauley, J. 2012. Learning to discover social circles in ego networks. In *NeurIPS*, volume 25.

Li, H.; Guo, D.; Fan, W.; Xu, M.; and Song, Y. 2023. Multi-step jailbreaking privacy attacks on chatgpt. arXiv:2304.05197.

Li, K.; Liu, Y.; Ao, X.; Chi, J.; Feng, J.; Yang, H.; and He, Q. 2022. Reliable representations make a stronger defender: Unsupervised structure refinement for robust gnn. In *KDD*, 925–935.

Li, X.; Wang, R.; Cheng, M.; Zhou, T.; and Hsieh, C.-J. 2024. Drattack: Prompt decomposition and reconstruction makes powerful llm jailbreakers. arXiv:2402.16914.

Liu, P.; Yuan, W.; Fu, J.; Jiang, Z.; Hayashi, H.; and Neubig, G. 2023a. Pre-train, Prompt, and Predict: A Systematic Survey of Prompting Methods in Natural Language Processing. *ACM Comput. Surv.*, 195:1–195:35.

Liu, X.; Xu, N.; Chen, M.; and Xiao, C. 2024a. Autodan: Generating stealthy jailbreak prompts on aligned large language models. In *ICLR*.

Liu, X.; Yu, Z.; Zhang, Y.; Zhang, N.; and Xiao, C. 2024b. Automatic and universal prompt injection attacks against large language models. arXiv:2403.04957.

Liu, Y.; Wu, Z.; Lu, Z.; Nie, C.; Wen, G.; Hu, P.; and Zhu, X. 2024c. Noisy Node Classification by Bi-level Optimization based Multi-teacher Distillation. *arXiv preprint arXiv:2404.17875*.

Liu, Y.; Wu, Z.; Lu, Z.; Wen, G.; Ma, J.; Lu, G.; and Zhu, X. 2023b. Multi-teacher Self-training for Semi-supervised Node Classification with Noisy Labels. In *Proceedings of the 31st ACM International Conference on Multimedia*, 2946–2954.

Niu, Z.; Ren, H.; Gao, X.; Hua, G.; and Jin, R. 2024. Jailbreaking attack against multimodal large language model. In *ICLR*.

Olatunji, I. E.; Hizber, A.; Sihlovec, O.; and Khosla, M. 2023. Does black-box attribute inference attacks on graph neural networks constitute privacy risk? arXiv:2306.00578.

Olatunji, I. E.; Nejdl, W.; and Khosla, M. 2021. Membership Inference Attack on Graph Neural Networks. In *TPS-ISA*, 11–20.

Rozemberczki, B.; and Sarkar, R. 2020. Characteristic functions on graphs: Birds of a feather, from statistical descriptors to parametric models. In *CIKM*, 1325–1334.

Sharma, K.; Lee, Y.-C.; Nambi, S.; Salian, A.; Shah, S.; Kim, S.-W.; and Kumar, S. 2024. A survey of graph neural networks for social recommender systems. *ACM Computing Surveys*, 1–34.

Shi, J.; Yuan, Z.; Liu, Y.; Huang, Y.; Zhou, P.; Sun, L.; and Gong, N. Z. 2024. Optimization-based Prompt Injection Attack to LLM-as-a-Judge. arXiv:2403.17710.

Shokri, R.; Stronati, M.; Song, C.; and Shmatikov, V. 2017. Membership Inference Attacks Against Machine Learning Models. In *S & P*, 3–18.

Takac, L.; and Zabovsky, M. 2012. Data analysis in public social networks. In *International scientific conference and international workshop present day trends of innovations*.

Tramèr, F.; Shokri, R.; San Joaquin, A.; Le, H.; Jagielski, M.; Hong, S.; and Carlini, N. 2022. Truth serum: Poisoning machine learning models to reveal their secrets. In *CCS*, 2779–2792.

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

Wang, X.; and Wang, W. H. 2022. Group property inference attacks against graph neural networks. In *CCS*, 2871–2884.

Wei, Y.; Yuan, H.; Fu, X.; Sun, Q.; Peng, H.; Li, X.; and Hu, C. 2024. Poincaré Differential Privacy for Hierarchy-aware Graph Embedding. In *AAAI*, 9160–9168.

Wu, B.; Yang, X.; Pan, S.; and Yuan, X. 2021. Adapting membership inference attacks to GNN for graph classification: Approaches and implications. In *ICDM*, 1421–1426. IEEE.

Wu, T.; Ren, H.; Li, P.; and Leskovec, J. 2020. Graph information bottleneck. *NeurIPS*, 20437–20448.

Wu, Z.; Zhou, P.; Wen, G.; Wan, Y.; Ma, J.; Cheng, D.; and Zhu, X. 2022. Information Augmentation for Few-shot Node Classification. In *IJCAI*, 3601–3607.

Yao, D.; Zhang, J.; Harris, I. G.; and Carlsson, M. 2024. Fuzzllm: A novel and universal fuzzing framework for proactively discovering jailbreak vulnerabilities in large language models. In *ICASSP*, 4485–4489.

Yuan, H.; Sun, Q.; Fu, X.; Ji, C.; and Li, J. 2024a. Dynamic Graph Information Bottleneck. In *Proceedings of the ACM on Web Conference 2024*, 469–480.

Yuan, H.; Sun, Q.; Fu, X.; Zhang, Z.; Ji, C.; Peng, H.; and Li, J. 2024b. Environment-aware dynamic graph learning for out-of-distribution generalization. *Advances in Neural Information Processing Systems*, 36.

Zarifzadeh, S.; Liu, P.; and Shokri, R. 2024. Low-Cost High-Power Membership Inference Attacks. In *ICML*.

Zhang, C.; Bengio, S.; Hardt, M.; Recht, B.; and Vinyals, O. 2017. Understanding deep learning requires rethinking generalization. In *ICLR*.

Zhang, G.; Zhang, S.; and Yuan, G. 2024. Bayesian graph local extrema convolution with long-tail strategy for misinformation detection. *ACM Transactions on Knowledge Discovery from Data*, 18(4): 1–21.

Zhang, Q.; Wang, H.; Long, C.; Su, L.; He, X.; Chang, J.; Wu, T.; Yin, H.; Yiu, S.-M.; Tian, Q.; et al. 2024. A Survey of Generative Techniques for Spatial-Temporal Data Mining. arXiv:2405.09592.

Zhang, Y.; Zhao, Y.; Li, Z.; Cheng, X.; Wang, Y.; Kotevska, O.; Yu, P. S.; and Derr, T. 2023. A Survey on Privacy in Graph Neural Networks: Attacks, Preservation, and Applications. arXiv:2308.16375.

Zhang, Z.; Chen, M.; Backes, M.; Shen, Y.; and Zhang, Y. 2022. Inference Attacks Against Graph Neural Networks. In *USENIX Security Symposium*, 4543–4560.

Zhang, Z.; Liu, Q.; Wang, H.; Lu, C.; and Lee, C. 2021. Motif-based Graph Self-Supervised Learning for Molecular Property Prediction. In *NeurIPS*, 15870–15882.

Zhiyao, Z.; Zhou, S.; Mao, B.; Zhou, X.; Chen, J.; Tan, Q.; Zha, D.; Feng, Y.; Chen, C.; and Wang, C. 2024. Opengsl: A comprehensive benchmark for graph structure learning. In *NeurIPS*.