

SFR-GNN: Simple and Fast Robust GNNs against Structural Attacks

Xing Ai¹, Guanyu Zhu¹, Yulin Zhu¹, Yu Zheng^{1,2}, Gaolei Li³, Jianhua Li³, Kai Zhou¹

¹The Hong Kong Polytechnic University, HKSAR

²The Chinese University of Hong Kong, HKSAR

³Shanghai Jiao Tong University, Shanghai, China

Abstract

Graph Neural Networks (GNNs) have demonstrated commendable performance for graph-structured data. Yet, GNNs are often vulnerable to adversarial structural attacks as embedding generation relies on graph topology. Existing efforts are dedicated to purifying the maliciously modified structure or applying adaptive aggregation, thereby enhancing the robustness against adversarial structural attacks. It is inevitable for a defender to consume heavy computational costs due to lacking prior knowledge about modified structures. To this end, we propose an efficient defense method, called Simple and Fast Robust Graph Neural Network (SFR-GNN), supported by mutual information theory. The SFR-GNN first pre-trains a GNN model using node attributes and then fine-tunes it over the modified graph in the manner of contrastive learning, which is free of purifying modified structures and adaptive aggregation, thus achieving great efficiency gains. Consequently, SFR-GNN exhibits a 24%–162% speedup compared to advanced robust models, demonstrating superior robustness for node classification tasks.

Introduction

Graph Neural Networks (GNNs) have emerged as the leading approach for graph learning tasks across various domains, including recommender system (Zhang and Gan 2024), social networks (Hu et al. 2023), and bioinformatics (Liu et al. 2023). However, numerous studies (Zügner, Akbarnejad, and Günnemann 2018; Xu et al. 2019; Hussain et al. 2021; Zhu et al. 2022a) have demonstrated the vulnerability of GNNs under *adversarial attacks*, where an attacker can deliberately modify the graph data to cause the misprediction of GNNs. Among them, *structural attacks* (Zügner, Akbarnejad, and Günnemann 2018; Zügner and Günnemann 2019; Liu et al. 2022) have gained prominence due to the unique structural nature of graph data. Specifically, by solely modifying the edges in a graph, structural attacks hold practical significance in application scenarios where attackers have limited access to the relationships among entities rather than the attributes of the entities themselves.

To defend against structural attacks, numerous robust GNN models (Jin et al. 2020; Entezari et al. 2020a; Jin et al. 2021; Zhu et al. 2023) have been proposed recently. The

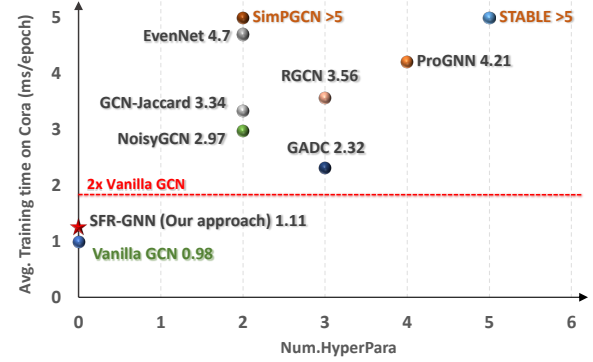


Figure 1: Computational Complexity and Hyperparameter Complexity Comparison of Existing Robust GNNs. Our method SFR-GNN is highlighted with a red star.

main ideas behind these approaches involve purifying the modified graph structure or designing adaptive aggregation mechanisms to avoid aggregating messages through poisoned structures. Despite these efforts, existing robust GNNs still encounter significant scalability challenges, which hinder their application in practical scenarios. These scalability issues are mainly attributed to two factors: *computational complexity* and *hyper-parameter complexity*.

Specifically, recent research (Zhu et al. 2021; Lei et al. 2024; Ennadir et al. 2024) reveals that current robust GNN models suffer from high computational complexity due to complex defense mechanisms, such as learning new graph structures or computing edge attention. Moreover, these robust models often introduce additional hyper-parameters (e.g. weighting coefficients and thresholds) beyond the basic ones (e.g. learning rates, dropout ratio, epochs). Unfortunately, effective hyper-parameter tuning often requires extensive background knowledge of the data, which may not always be available due to issues like distribution shifts. The interactions among multiple hyper-parameters compel developers to employ techniques such as grid search or cross-validation to ensure optimal values for all hyper-parameters, complicating model deployment in real-world scenarios (Wang et al. 2021; Chen et al. 2022).

Fig. 1 compares state-of-the-art robust GNNs with the

vanilla GCN in terms of training time (per epoch in milliseconds) on Cora dataset and the number of extra hyper-parameters. The results indicate that all robust GNNs require more than twice the runtime of the vanilla GCN and introduce at least two extra hyper-parameters (with our method as an exception). It demonstrates that while existing robust GNNs achieve adversarial robustness, it comes at a significant cost in training time and hyper-parameter complexity.

In response, a natural question emerges: *can we develop a GNN model that achieves adversarial robustness against structural attacks while also being simple and fast?*

In this work, we propose a Simple and Fast Robust Graph Neural Network (SFR-GNN) that employs a simple but effective learning strategy: pre-training on node attributes and then fine-tuning on structure information. Specifically, given a positioned graph $\mathcal{G}' = (\mathbf{X}, \mathbf{A}', \mathbf{Y})$ with manipulated structure \mathbf{A}' , SFR-GNN is initially pre-trained using only node attributes \mathbf{X} without structural information involved. Subsequently, the model is fine-tuned over \mathbf{A}' , devising a *specialized* graph contrastive learning scheme.

The idea behind this strategy is rooted in the analysis of the structural attack: the attacker meticulously generates a modified structure \mathbf{A}' based on the given node attributes \mathbf{X} , which is detrimental to the performance of GNN *with respect to the corresponding* \mathbf{X} . Theoretically, structural attacks contaminate the mutual information between \mathbf{A}' and \mathbf{Y} conditioned by \mathbf{X} : $I(\mathbf{A}'; \mathbf{Y}|\mathbf{X})$ to mislead GNN predictions. However, we indicate the “**paired effect**” of structural attacks: \mathbf{A}' is most effective alongside the given \mathbf{X} , and is not quite effective with any other $\mathbf{X}' \neq \mathbf{X}$ (detailed in Sec.). Therefore, our strategy achieves robustness against structural attacks by disrupting the “**paired effect**”. This is achieved by first pre-training on attributes \mathbf{X} to obtain node embedding \mathbf{Z}_p and then fine-tunes it paired with \mathbf{A}' to incorporate structural information, which actually pairs \mathbf{A}' with \mathbf{Z}_p instead of \mathbf{X} , thus mitigating the impact of the manipulated structure on model performance. Despite its simplicity, we provide theoretical support and insights through a mutual information perspective in Sec. .

As a result, SFR-GNN features a lightweight construction with no *additional* hyper-parameters, significantly alleviating the computational and hyper-parameter complexity associated with building robust GNNs. Fig. 1 illustrates that the computational and hyperparameter complexity of SFR-GNN is close to that of vanilla GCN and outperforms existing robust GNNs, highlighting the simplicity and ease of implementation of SFR-GNN. Datasets and codes of this paper are at the supplements.

Our major contributions are summarized as follows.

- 1) We propose a novel, simple and fast robust GNN model named SFR-GNN that employs an “attribute pre-training and structure fine-tuning” strategy to achieve robustness against structural attacks. This approach is efficient in that it requires no extra hyper-parameters and is free of time-consuming operations such as purification or attention mechanisms.
- 2) We offer a comprehensive theoretical analysis through mutual information theory to provide insights into the

proposed method and substantiate its effectiveness.

- 3) The comprehensive evaluation of SFR-GNN against state-of-the-art baselines on node classification benchmarks, encompassing large-scale graph datasets, reveals that it achieves comparable or superior robustness while significantly enhancing runtime efficiency by a range of 24% to 162% compared to state-of-the-art baselines.

Related Works

Structural Attacks in Graph Learning. Structural attacks are a popular form of attack covering a wide range of graph learning models beyond GNNs, including self-supervised learning (Bojchevski and Günnemann 2019; Zhang et al. 2022b), signed graph analysis (Zhou et al. 2023; Zhu et al. 2024), recommender systems (Lai et al. 2023), and so on. The primary idea is to utilize gradient-based methods to search for the optimal graph structure to degrade the performances of various tasks. For instance, Mettack (Zügner and Günnemann 2019) formulated the global structural poisoning attacks on GNNs as a bi-level optimization problem and leveraged a meta-learning framework to solve it. BinarizedAttack (Zhu et al. 2022b) simplified graph poisoning attacks against the graph-based anomaly detection to a one-level optimization problem. HRAT (Zhao et al. 2021) proposed a heuristic optimization model integrated with reinforcement learning to optimize the structural attacks against Android malware detection. GraD (Liu et al. 2022) proposes a reasonable budget allocation mechanism to enhance the effects of structural attacks.

Robust GNNs. To defend against structural attacks, a series of robust GNNs are proposed, which typically rely on purifying the modified structure or designing adaptive aggregation strategies. For example, GNNGUARD (Zhang and Zitnik 2020) removes the malicious links during training by considering the cosine similarity of node attributes. Zhao et al. (Zhao et al. 2023) used a conservative Hamiltonian flow to improve the model’s performance under adversarial attacks. However, common drawbacks of these approaches include high computational overhead and hyper-parameter complexity. More recently, few works have attempted to develop efficient robust GNNs. For example, NoisyGCN (Ennadir et al. 2024) defends against structural attacks by injecting random noise into the architecture of GCN, thereby avoiding complex strategies and improving runtime. Similarly, EvenNet (Lei et al. 2024) proposes an efficient strategy that ignores odd-hop neighbors of nodes, with a time complexity that is linear to the number of nodes and edges in the input graph. These efforts significantly reduce the time complexity of building robust GNNs but still introduce additional hyper-parameters. NoisyGCN requires careful selection of the ratio of injected noise and EvenNet requires the determination of both the order of the graph filter K and the initialization hyper-parameter α . This motivates us to develop even simpler while robust GNNs.

Background

We consider the node classification task in a semi-supervised setting. Specifically, let $\mathcal{G} = (\mathbf{X}, \mathbf{A}, \mathbf{Y})$ be an input graph,

where $\mathbf{X} \in \mathbb{R}^{n \times d}$ denotes node attributes, $\mathbf{A} \in \{0, 1\}^{n \times n}$ is the adjacent matrix, and \mathbf{Y} represents the partially available labels of the nodes in the training set. A GNN model f_θ parameterized with θ is trained to predict the remaining node labels through minimizing the training loss \mathcal{L}_{tr} given the training node labels:

$$\theta^* = \arg \min_{\theta} \mathcal{L}_{tr}(f_\theta(\mathbf{X}, \mathbf{A}), \mathbf{Y}). \quad (1)$$

This training loss \mathcal{L}_{tr} is commonly employed classification loss such as the Negative Log-Likelihood.

Structural Attacks. Structural attacks against semi-supervised node classification naturally fit within a *poisoning attack setting*, where the GNN model is trained and makes predictions over a manipulated graph. In a worst-case scenario, it is assumed that the attacker can arbitrarily modify the graph structure (i.e., \mathbf{A}) with the goal of degrading classification performance. Specifically, the attacker seeks to find an optimal structural perturbation δ^* , resulting in a poisoned graph $\mathcal{G}' = (\mathbf{X}, \mathbf{A}' = \mathbf{A} + \delta^*, \mathbf{Y})$. Mathematically, a structural attack can be formulated as solving a bi-level optimization problem:

$$\begin{aligned} \delta^* &= \arg \min_{\delta} \mathcal{L}_{atk}(f_{\theta^*}(\mathbf{X}, (\mathbf{A} + \delta)), \mathbf{Y}) \\ \text{s.t. } \theta^* &= \arg \min_{\theta} \mathcal{L}_{tr}(f_\theta(\mathbf{X}, \mathbf{A} + \delta), \mathbf{Y}), \end{aligned} \quad (2)$$

where \mathcal{L}_{atk} quantifies the attack objective. The attacks (Zügner and Günnemann 2019; Liu et al. 2022; Xu et al. 2019) mainly differ in their specific algorithms to solve the optimization problem.

Robust GNNs as Defense. Training robust GNN models is a common defense strategy to mitigate structural attacks. In this paper, the defender’s goal is to train a robust GNN model from the poisoned graph to maintain node classification accuracy. We note that the defender only has access to the poisoned graph $\mathcal{G}' = (\mathbf{X}, \mathbf{A} + \delta^*, \mathbf{Y})$, not the clean graph \mathcal{G} . Additionally, the defender does not have prior knowledge about how the perturbation δ^* was generated.

Methodology

Design Intuition

We propose a novel framework for efficiently learning robust GNNs against structural attacks, which employs a straightforward strategy: *attribute pre-training and structure fine-tuning*, to alleviate computational and hyper-parameter complexity. We articulate this design and the intuition behind it through an information theoretical perspective. For completeness and better readability, we defer all theoretical analysis to Section .

Our intuition starts from a key observation of the essence of attacks: *structural attacks degrade GNN’s performance by contaminating the mutual information between \mathbf{A} and \mathbf{Y} conditioned on \mathbf{X}* , denoted as $I(\mathbf{A}; \mathbf{Y}|\mathbf{X})$ (see Lemma 1 for details). That is, given fixed attributes \mathbf{X} , the attacker can generate a poisoned structure \mathbf{A}' to effectively attack GNNs. Moreover, the structural attack has a “**paired effect**”: the

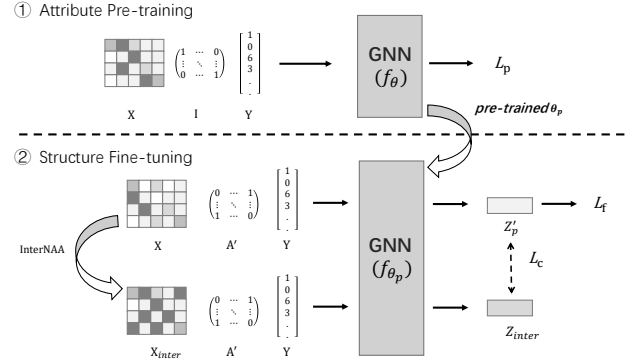


Figure 2: Framework of SFR-GNN.

poisoned structure \mathbf{A}' is effective with the given \mathbf{X} , and is not quite effective with any other $\mathbf{X}' \neq \mathbf{X}$ (see Lemma. 2).

The above analysis reveals the key to designing robust GNNs: create a mismatch between \mathbf{A}' and the attributes \mathbf{X} . Previous works did so by trying to purify \mathbf{A}' , however, with high computational complexity. We employ a totally different strategy: attribute pre-training and structure fine-tuning essentially involves obtaining a latent node embedding \mathbf{Z} through pre-train on \mathbf{X} , and then fine-tune \mathbf{Z} with \mathbf{A}' to incorporate structural information. This approach allows the model to learn from the *less harmful* $I(\mathbf{A}'; \mathbf{Y}|\mathbf{Z})$ instead of the contaminated $I(\mathbf{A}'; \mathbf{Y}|\mathbf{X})$ (see Theorem. 1).

However, since \mathbf{Z} is pre-trained from \mathbf{X} , there exists overlap between $I(\mathbf{A}'; \mathbf{Y}|\mathbf{Z})$ and $I(\mathbf{A}'; \mathbf{Y}|\mathbf{X})$, meaning that structural attacks still affects $I(\mathbf{A}'; \mathbf{Y}|\mathbf{Z})$. We thus further propose a novel contrastive learning approach to learn structural information from $I(\mathbf{A}'; \mathbf{Y}|\mathbf{Z})$ while mitigating the attack effect (see Theorem. 2). The detailed constructions are presented in the next section.

Detailed Construction

To implement the intuition in Sec. , we propose a novel method, namely Simple and Fast Robust Graph Neural Network (SFR-GNN) consisting of two main stages: attributes pre-training and structure fine-tuning as shown in Fig. 2 and Alg. 1. First, SFR-GNN pre-trains over the node attributes without structural information to generate node embeddings. Subsequently, SFR-GNN fine-tunes the node embeddings with the *modified* adjacency matrix to incorporate structural information.

Attributes Pre-training. Attributes pre-training is employed to learn node embeddings solely from node attributes \mathbf{X} . Specifically, a randomly initialized GNN model f_θ with parameters θ takes node attributes \mathbf{X} of the input graph $\mathcal{G}' = (\mathbf{X}, \mathbf{A}', \mathbf{Y})$ and an identity matrix \mathbf{I} as inputs and aims to minimize the pre-training loss \mathcal{L}_p to learn node embeddings \mathbf{Z}_p :

$$\theta_p = \arg \min_{\theta} \mathcal{L}_p(f_\theta(\mathbf{X}, \mathbf{I}), \mathbf{Y}), \quad \mathbf{Z}_p = f_{\theta_p}(\mathbf{X}, \mathbf{I}). \quad (3)$$

The choice of pre-training loss \mathcal{L}_p can be any common classification loss, such as the Negative Log-Likelihood

Loss function (NLL). Since the pre-training process completely excludes \mathbf{A}' and \mathbf{Z}_p is learned from \mathbf{X} without being modified by structural attacks, \mathbf{Z}_p is uncontaminated. Lines 3-7 of Alg. 1 shows the attributes pre-training stage.

Although the embeddings \mathbf{Z}_p learned through attribute pre-training are sufficiently “clean”, the lack of structural information makes \mathbf{Z}_p insufficient to predict labels accurately. Hence, we propose structure fine-tuning, which adjusts \mathbf{Z}_p using \mathbf{A}' to incorporate some structural information.

Structure Fine-tuning. In structure fine-tuning, the model is initialized by the pre-trained parameters θ_p and minimizes the fine-tuning loss function \mathcal{L}_f and contrastive loss function \mathcal{L}_c simultaneously:

$$\theta^* = \arg \min_{\theta_p} \mathcal{L}_f(\mathbf{Z}'_p, \mathbf{Y}) + \mathcal{L}_c(\mathbf{Z}'_p, \mathbf{Z}_{inter}),$$

$$\mathbf{Z}'_p = f_{\theta_p}(\mathbf{X}_{train}, \mathbf{A}'), \quad \mathbf{Z}_{inter} = f_{\theta_p}(\mathbf{X}_{inter}, \mathbf{A}'), \quad (4)$$

where the the fine-tuning loss function \mathcal{L}_f is as same as \mathcal{L}_p , \mathcal{L}_c is any typical contrastive function such as InfoNCE. \mathbf{X}_{inter} is generated by the proposed Inter-class Node Attributes Augmentation (InterNAA), which replaces the node feature of each node v in the training set with the average node feature of several nodes with the different class as v that are sampled randomly from the training set. The number of samples equals the degree of the node v . The process of InterNAA is shown in Lines 9-17 in Alg. 1.

The primary objectives of the structure fine-tuning stage are twofold: to ensure that \mathbf{Z}'_p contains structural information and to prevent it from being influenced by contaminated information in structure ($I(\mathbf{A}'; \mathbf{Y}|\mathbf{X})$). The former is achieved by minimizing fine-tuning loss \mathcal{L}_f , while the latter is achieved by minimizing contrastive loss \mathcal{L}_c . A detailed theoretical analysis is provided in Sec. . Here, we offer an intuitive explanation.

Firstly, the pre-trained parameters θ_p are used to initialize the model f . However, unlike the training stage, f receives \mathbf{A}' instead of \mathbf{I} as input, which means f fine-tunes the pre-trained embeddings \mathbf{Z}_p using structure information to obtain \mathbf{Z}'_p . Besides, by combining contrastive learning techniques to maximize the similarity between \mathbf{Z}'_p and \mathbf{Z}_{inter} , we effectively align \mathbf{Z}'_p with the less harmful $I(\mathbf{A}'; \mathbf{Y}|\mathbf{X}_{inter})$ rather than the contaminated $I(\mathbf{A}'; \mathbf{Y}|\mathbf{X})$. $I(\mathbf{A}'; \mathbf{Y}|\mathbf{X}_{inter})$ is less harmful because we replace \mathbf{X} with \mathbf{X}_{inter} generated by InterNAA, akin to reducing the lethality of a gun by providing it with mismatched bullets.

Computational Complexity. The computational complexity consists of two parts: the attribute pre-training stage (Lines 3-7 in Alg. 1) and the structure fine-tuning stage (Lines 19-24 in Alg. 1). Assuming our network is composed of L layers of graph convolutional layers, with F hidden units per layer, N nodes and E edges in the graph, pretraining epochs e_p , and finetuning epochs e_f . Since the attributes pretraining does not utilize the adjacency matrix \mathbf{A}' , its computational complexity can be considered equivalent to that of a multi-layer perceptron (MLP), which is $O(e_p L N F^2)$.

As for structure fine-tuning, the nodes in the training set are traversed to generate \mathbf{X}_{inter} (Lines 9-17 in Alg. 1),

Algorithm 1: Simple and Fast Robust Graph Neural Network

```

1: Input: Input graph  $\mathcal{G}'(\mathbf{X}, \mathbf{A}')$ , includes modified adjacency matrix  $\mathbf{A}'$  and node attributes  $\mathbf{X}$ , identity matrix  $\mathbf{I}$  and GNN model  $f$  with parameter  $\theta$ , pre-training epoch  $pretrain\_epoch$ , finetune epoch  $finetune\_epoch$ , pre-training loss  $\mathcal{L}_p$  and fine-tuning loss  $\mathcal{L}_f$  and contrastive loss  $\mathcal{L}_c$ , node set  $V_{train}$ ,  $\mathbf{X}$ ,  $\mathbf{Y}$  for training
2: /*Attributes pre-training*/
3: for  $e = 1, 2, \dots, pretrain\_epoch$  do
4:    $\mathbf{Z}_\theta = f_\theta(\mathbf{X}, \mathbf{I})$ 
5:    $\theta = \theta + \nabla_\theta \mathcal{L}_p(\mathbf{Z}_\theta, \mathbf{Y})$ 
6: end for
7:  $\theta_p = \theta$ 
8: /*Inter-class Node Attributes Augmentation*/
9: empty set  $\mathbf{X}_{inter}$ .
10: for  $v \in V_{train}$  do
11:    $c = v.class$ 
12:    $InterClassSet = \{V_{train}.class \neq c\}$ 
13:    $num = v.degree$ 
14:    $inter\_class = \text{RandomChoice}(num, InterClassSet)$ 
15:    $\mathbf{X}_{v.inter} = \text{mean}(\mathbf{X}_{inter.class})$ 
16:   add  $\mathbf{X}_{v.inter}$  into  $\mathbf{X}_{inter}$ 
17: end for
18: /*Structure fine-tuning*/
19: for  $e = 1, 2, \dots, finetune\_epoch$  do
20:    $\mathbf{Z}'_p = f_{\theta_p}(\mathbf{X}, \mathbf{A}')$ ,  $\mathbf{Z}_{inter} = f_{\theta_p}(\mathbf{X}_{inter}, \mathbf{A}')$ 
21:    $\mathcal{L} = \mathcal{L}_f(\mathbf{Z}'_{\theta_p}, \mathbf{Y}) + \mathcal{L}_c(\mathbf{Z}'_p, \mathbf{Z}_{inter})$ 
22:    $\theta_p = \theta_p + \nabla_{\theta_p} \mathcal{L}$ 
23: end for
24:  $\theta^* = \theta_p$ 
25: return  $f_{\theta^*}$ 

```

with computational complexity of $O(\sigma N \bar{d} F)$, where σ is the training ratio and \bar{d} is the average degrees. The computational complexity of obtaining \mathbf{Z}'_p and \mathbf{Z}_{inter} , is equal to applying twice calculations of GCNs: $O(2 * (L N F^2 + L E F))$ (Chen et al. 2020). The computational complexity for computing the contrastive loss is $O(\sigma^2 N^2 F)$ (Zhang et al. 2022a; Alon et al. 2024).

Thus the overall computational complexity of SFR-GNN is $O(e_p L N F^2 + \sigma N \bar{d} F + e_f (2 L N F^2 + 2 L E F) + \sigma^2 N^2 F)$. In the worst case when the graph is fully connected, where $\bar{d} = N$ and $E = N^2$, the complexity is $O(e_p L N F^2 + \sigma N^2 F + e_f (2 L N F^2 + 2 L N^2 F + \sigma^2 N^2 F))$. Since the training ratio σ is less than 1, e_p and e_f are constants smaller than N , and their impact on the overall complexity is negligible. Hence, the total complexity of SFR-GNN is $O(L N F^2 + L N^2 F)$, which is on par with that of GCN, and significantly lower than that of existing robust GNNs. The experiments in Sec. substantiate this claim.

Theoretical Analysis

Our theoretical analysis serves two purposes: first, to analyze the essence of structural attacks and the paired effect from the perspective of mutual information, providing a theoretical explanation for our intuition; second, to theoretically prove the effectiveness of our proposed “attributes pre-training, structure fine-tuning” strategy.

Given the fundamental properties of mutual information, performance degradation of GNN can be attributed to the maliciously generated adjacency matrix over true node attributes. Accordingly, we provide the understanding of structural attacks from an information-theoretic perspective as in Lemma. 1.

Lemma 1 (Essence of Structural Attacks). *Structural attacks degrade GNNs’ performance through generating the modified adjacency matrix \mathbf{A}' to contaminate the mutual information between the labels \mathbf{Y} and \mathbf{A}' conditioned by \mathbf{X} , which essentially uses the mutual information $I(\mathbf{A}'; \mathbf{Y}|\mathbf{X})$.*

The significance of Lemma. 1 lies in highlighting that structural attacks essentially generate modified structure \mathbf{A}' according to corresponding node attributes \mathbf{X} , which implies the potential relationships between \mathbf{A}' and \mathbf{X} . Building on Lemma. 1, we propose Lemma. 2 blow to demonstrate the important correspondence between \mathbf{A}' and \mathbf{X} in $I(\mathbf{A}'; \mathbf{Y}|\mathbf{X})$. Namely, $I(\mathbf{A}'; \mathbf{Y}|\mathbf{X})$ can only maximally degrade GNN performance under the condition of \mathbf{X} .

Lemma 2 (Paired Effect of Structural Attacks). *For any $\mathbf{X}' \neq \mathbf{X}$, where $\mathbf{X}', \mathbf{X} \in \mathbb{R}^{n \times d}$, the mutual information $I(\mathbf{A}'; \mathbf{Y}|\mathbf{X}')$ is less harmful to GNNs than $I(\mathbf{A}'; \mathbf{Y}|\mathbf{X})$.*

Notably, Lemma. 2 implies a new defense strategy against structural attacks from the root cause. Unlike existing methods that focus on purifying the modified structure or employing adaptive aggregation, our approach does not require any operations on the modified structure. Instead, it replaces the corresponding attributes to disrupt the paired effect, thereby reducing the attack effectiveness of the modified structure on GNNs.

Motivated by Lemma. 2, SFR-GNN pre-trains node embeddings \mathbf{Z}_p solely on node attributes \mathbf{X} , and force the proposed model to learn information from $I(\mathbf{A}'; \mathbf{Y}|\mathbf{Z}_p)$, which actually replaces \mathbf{X} with \mathbf{Z}_p . We provide Theorem. 1 to demonstrate \mathbf{Z}_p shares mutual information with labels \mathbf{Y} and $I(\mathbf{A}'; \mathbf{Y}|\mathbf{Z}_p)$ is less harmful compared to $I(\mathbf{A}'; \mathbf{Y}|\mathbf{X})$.

Theorem 1. *SFR-GNN’s pre-training stage maximizes the mutual information $I(\mathbf{Z}_p; \mathbf{Y})$ between \mathbf{Z}_p and \mathbf{Y} , where $I(\mathbf{A}'; \mathbf{Y}|\mathbf{Z}_p)$ is less harmful to GNNs compared to $I(\mathbf{A}'; \mathbf{Y}|\mathbf{X})$.*

Although $I(\mathbf{A}'; \mathbf{Y}|\mathbf{Z}_p)$ is less harmful than $I(\mathbf{A}'; \mathbf{Y}|\mathbf{X})$, there may be an overlap between them since \mathbf{Z}_p is learned from \mathbf{X} , leading to the contamination of $I(\mathbf{A}'; \mathbf{Y}|\mathbf{Z}_p)$, as demonstrated in Lemma.3. Lemma.3 essentially explains the reason for employing contrastive learning, i.e., $I(\mathbf{A}'; \mathbf{Y}|\mathbf{Z}_p)$ may be contaminated. Based on Lemma.3, we introduce contrastive learning to align $I(\mathbf{A}'; \mathbf{Y}|\mathbf{Z}_p)$ to $I(\mathbf{A}'; \mathbf{Y}|\mathbf{X}_{inter})$ to prevent it from being contaminated, as demonstrated in Theorem. 2.

Lemma 3. *There exists an overlap between $I(\mathbf{A}'; \mathbf{Y}|\mathbf{Z}_p)$ and $I(\mathbf{A}'; \mathbf{Y}|\mathbf{X})$, which consequently leads to the contamination of $I(\mathbf{A}'; \mathbf{Y}|\mathbf{Z}_p)$.*

Theorem 2. *SFR-GNN’s structure fine-tuning stage maximizes $I(\mathbf{A}'; \mathbf{Y}|\mathbf{Z}_p)$ to learn structural information and align it to $I(\mathbf{A}'; \mathbf{Y}|\mathbf{X}_{inter})$ to prevent from being contaminated.*

Dataset	hyper-parameters	Range
SimP-GCN	λ	$\{0.1, 1, 10\}$
	γ	$\{0.01, 0.1, 1\}$
ProGNN	α	0.00025 to 0.064
	β	0 to 5
	γ	0.0625 to 16
	λ	1.25 to 320
STABLE	k	1 to 13
	α	-0.5 to 3
	t_1	0 to 0.05
	t_2	-1 to 0.5
	p	0.2
EvenNet	k	$\{4, 6, 8, 10\}$
	α	0 to 0.5
NoisyGCN	β	0.1 to 0.5
	ϵ	$\{0, 0.1\}$
GADC	ξ	0.01 to 1
	K	$\{4, 8, 16, 32\}$
	λ	$\{4, 8, 16, 32\}$

Table 1: Hyper-parameters of baselines and their choices.

Theorem.2 demonstrates the effectiveness of contrastive learning and the proposed InterNAA to mitigate the contamination. We provide complete proofs in the Appendix and present empirical experiments in Sec. to support the aforementioned claims.

Experiments

Datasets. We conduct experiments on three widely used benchmarks: Cora (McCallum et al. 2000), CiteSeer (Giles, Bollacker, and Lawrence 1998), Pubmed (Sen et al. 2008), and two large-scale graph datasets (ogbn-arxiv, and ogbn-products), with details presented in the Appendix. Furthermore, experimental results on two heterophilic graph datasets, demonstrating the robustness of the proposed method, are provided in the Appendix.

Implementation and Baselines. We conducted an empirical comparison against eight state-of-the-art baseline defense algorithms, including RGCN (Zhu et al. 2019), GCN-Jaccard (Entezari et al. 2020a), SimP-GCN (Jin et al. 2021), Pro-GNN (Jin et al. 2020), STABLE (Li et al. 2022), EvenNet (Lei et al. 2024), GADC (Liu et al. 2024), and NoisyGCN (Ennadir et al. 2024), which achieve remarkable performance in terms of structure attack defense. We select two representative structure attack methods, i.e., Mettack (Zügner and Günnemann 2019) and GraD (Liu et al. 2022), to verify the robustness of the proposed method and baselines. Source code and configuration of baselines are obtained from either the public implementation of DeepRobust (Li et al. 2020), or the official implementation of the authors. Detailed configurations are deferred to the Appendix.

Experiments Settings. Experiments are conducted on a device with 16 Gen Intel(R) Core(TM) i9-12900F cores and an NVIDIA L20 (48GB memory). On Cora, Citeseer and Pubmed, we follow the data splitting method of DeepRobust: randomly selecting 10% of the nodes for training, 10% for validation, and the remaining 80% for testing. As for

Datasets	Attacker	ptb(%)	RGCN	GCN-Jaccard	SimP-GCN	Pro-GNN	STABLE	EvenNet	GADC	NoisyGCN	SFR-GNN (Ours)
Cora	clean		83.4±0.2	82.2±0.5	82.1±0.6	83.0±0.2	81.1±0.5	83.1±0.4	79.0±0.3	82.9±0.6	83.4±0.5
	Mettack	0.05	72.0±0.5	78.8±0.6	80.5±1.7	<u>82.3±0.5</u>	81.4±0.6	80.0±0.8	78.7±0.3	77.4±1.2	82.6±0.6
		0.1	68.9±0.3	76.9±0.5	79.0±0.9	79.0±0.6	<u>81.0±0.4</u>	77.8±1.1	78.2±0.6	75.6±1.2	82.1±0.6
		0.2	62.8±1.2	75.2±0.7	76.1±2.0	73.3±1.6	<u>80.4±0.7</u>	78.2±0.9	77.1±0.6	74.5±1.3	81.1±0.8
	GraD	0.05	81.7±0.5	81.4±0.6	80.9±0.5	<u>81.8±0.5</u>	81.1±0.4	80.4±0.5	79.0±0.3	82.0±0.6	82.0±0.6
		0.1	79.9±0.4	80.3±0.4	80.9±0.5	80.9±0.3	80.2±0.5	78.7±0.7	78.7±0.3	<u>81.0±0.3</u>	81.1±0.8
		0.2	77.9±0.5	79.8±0.6	78.9±0.8	78.3±0.2	79.8±0.3	78.3±1.0	78.6±0.4	79.1±0.6	<u>79.6±0.9</u>
Citeseer	clean		71.8±0.6	72.6±0.6	<u>73.8±0.7</u>	73.3±0.7	<u>73.9±0.6</u>	73.8±0.5	73.4±0.5	72.3±0.4	75.1±0.4
	Mettack	0.05	70.5±1.0	72.0±0.4	73.0±0.7	72.9±0.6	72.6±0.3	<u>73.5±0.4</u>	73.0±0.8	71.3±0.3	74.7±0.5
		0.1	69.4±0.8	71.8±0.5	<u>74.1±0.7</u>	72.5±0.8	73.5±0.5	73.3±0.4	73.0±0.8	71.2±0.4	74.3±0.4
		0.2	67.7±0.5	70.6±0.3	70.9±0.5	70.0±2.3	72.8±0.7	<u>73.2±0.5</u>	72.9±0.8	70.2±0.7	73.7±0.4
	GraD	0.05	71.6±0.8	72.1±0.9	73.5±0.7	72.2±0.1	73.5±0.4	<u>73.8±0.8</u>	73.5±0.8	72.0±0.8	75.0±0.6
		0.1	70.7±0.6	72.3±0.7	<u>73.5±0.6</u>	72.1±0.1	72.5±0.6	73.0±0.5	73.4±0.8	71.8±0.4	74.5±0.6
		0.2	67.6±0.6	70.1±0.8	72.6±0.7	70.6±0.6	72.1±0.5	72.7±0.5	<u>73.2±0.7</u>	70.4±0.5	73.4±0.6
Pubmed	clean		85.4±0.1	86.2±0.1	<u>87.1±0.1</u>	87.3±0.2	85.0±0.1	86.7±0.1	86.4±0.1	85.0±0.0	85.4±0.4
	Mettack	0.05	83.0±0.2	83.6±0.5	<u>86.5±0.1</u>	87.2±0.1	81.3±0.1	86.0±0.2	86.3±0.1	79.7±0.2	85.3±0.4
		0.1	83.0±0.2	79.6±0.2	86.0±0.2	87.2±0.1	79.0±0.1	85.6±0.2	<u>86.3±0.2</u>	67.4±0.2	85.1±0.3
		0.2	81.4±0.2	70.5±0.4	<u>85.7±0.2</u>	87.2±0.2	78.4±0.1	85.3±0.2	86.1±0.1	56.5±0.4	84.5±0.4
	GraD	0.05	82.9±0.1	84.0±0.2	86.6±0.2	85.4±0.0	82.6±0.1	86.2±0.2	<u>86.3±0.1</u>	82.8±0.1	85.2±0.3
		0.1	81.8±0.1	82.7±0.1	<u>86.0±0.2</u>	85.0±0.2	81.5±0.1	85.9±0.2	86.1±0.1	81.7±0.1	84.5±0.5
		0.2	79.6±0.1	80.5±0.2	<u>85.3±0.4</u>	82.8±0.1	79.2±0.1	85.4±0.2	86.1±0.1	80.0±0.1	83.4±0.4

Table 2: Average classification accuracy (\pm standard deviation) of 10 runs under two structural attacks with different perturbation ratios (ptb). The best and second-best results are highlighted in bold and underlined, respectively.

ogbn-arxiv and ogbn-products, we follow dataset splits provided by OGB (Hu et al. 2020). As for hyper-parameters of baselines, we follow the authors’ suggestion to search for the optimal values. Table. 1 shows all hyper-parameters and their ranges. It can be observed that existing robust GNNs require multiple hyper-parameters, and some of them have a large search range. Consequently, existing robust GNNs require many training runs to determine the optimal values of all hyper-parameters.

Defense Performance. To showcase the effectiveness and efficiency of the proposed method, we compare its robustness (Table. 2) and training time (Table. 4) against two typical attack methods: Mettack and GraD, on three datasets with baselines. It’s worth noting that the proposed method always achieves the best performance or the second-best performance on Cora and Citeseer, highlighting its robustness, which is on par with or exceeds state-of-the-art baselines. For instance, the proposed method achieves 82.1% accuracy on Cora dataset under Mettack with 10% perturbation ratio while baselines’ accuracy ranges from 69% to 81%. Besides, for Citeseer, the proposed method achieves tiny but continuous improvements compared to baselines under all perturbation ratios. On the Pubmed dataset, while SFR-GNN does not surpass strong baselines like SimP-GCN and Pro-GNN, it still outperforms several other baselines. We speculate that the reason is more complex and larger models tend to have an advantage on larger datasets like Pubmed.

Besides the robustness improvements, the proposed method also achieves significant training time speedup compared to baselines as shown in Tabel. 4. Upon examining the table, we can observe that compared to the fastest existing methods in their respective categories, such as NoisyGCN and GADC, the proposed method achieves over a 100%

speedup on Cora, and Citeseer. Conversely, when compared to slower methods like SimP-GCN and STABLE, the proposed method’s speed is nearly 10 times that of theirs. The significant speedup achieved by the proposed method can be attributed to the elimination of time-consuming modified structure identification and processing operations.

Scalability to Large-scale Graph. We conduct experiments on two publicly available large-scale graph datasets, ogbn-arxiv and ogbn-products (Hu et al. 2020), to validate the scalability of the proposed method. Owing to memory overflow issues encountered by structure attack methods like Mettack on large-scale graphs, we employ PRBCD (Geisler et al. 2021) as the attack method for these settings, and compare its performance against four defense methods capable of scaling to large graphs. The tests are performed using the officially provided modified adjacency matrices, with the results presented in Table. 3. Given the substantial memory requirements of the contrastive learning component, to facilitate the scalability of SFR-GNN to large-scale graphs, we introduce a variant of our approach: SFR-GNN (w/o CL), which excludes contrastive learning during the structure fine-tuning stage, thereby enabling its effective application to large-scale graphs without encountering memory constraints.

Results in Table. 3 demonstrate that SFR-GNN consistently achieves either the top or second-best performance across various perturbation ratios on two large-scale datasets. Notably, it also exhibits the fastest runtime, surpassing even GCN in speed. This efficiency stems from the attributes pre-training stage of SFR-GNN, which is free from structure information related computations. Additionally, EvenNet and Soft Medoid GDC encountered out-of-memory (OOM) issues on ogbn-products. This is attributed

Dataset	ptb(%)	GCN	GADC	EvenNet	Soft Medoid GDC	Soft Median GDC	SFR-GNN (w/o CL)
ogbn-arxiv	clean	66.9±0.32	64.1±0.15	63.2±1.3	57.5±0.24	64.1±0.15	67.0±0.22
	1%	54.8±0.29	55.6±0.13	36.4±7.92	52.2±0.22	56.9±0.19	58.8±0.16
	5%	34.6±0.32	45.2±0.17	32.4±4.94	48.0±0.27	47.1±0.21	48.5±0.21
	10%	29.5±0.58	36.4±0.19	29.2±2.45	45.4±0.31	40.8±0.33	41.5±0.21
	speed(ms/epoch)	109.15	119.8	89.9	145.3	132.7	53.0
ogbn-products	clean	73.5±0.08	73.0±0.05	OOM	OOM	64.3±0.	74.0±0.28
	1%	63.6±0.10	66.6±0.12			63.0±0.08	69.9±0.30
	5%	49.5±0.09	58.7±0.17			59.0±0.11	59.6±0.3
	10%	46.3±0.11	52.5±0.26			56.9±0.14	54.2±0.26
	speed(ms/epoch)	367.7	355.2			413.0	190.2

Table 3: Average classification accuracy (\pm standard deviation) and average training speed (in milliseconds per epoch) of 10 runs on large-scale graph datasets under PRBCD attacks.

Dataset	Cora	Citeseer	Pubmed
RGCN	3.56	3.57	41.39
GCN-Jaccard	3.34	3.28	13.66
SimP-GCN	10.42	9.89	37.68
ProGNN	4.21	4.89	>1,000
STABLE	7.26	5.52	64.10
EvenNet	4.70	4.65	4.75
NoisyGCN	3.19	3.55	4.47
GADC	2.32	3.25	6.65
SFR-GNN	1.11 (\uparrow 109%)	1.24 (\uparrow 162%)	3.61 (\uparrow 24%)

Table 4: Average Training Time Comparison (ms/epoch).

to the fact that Soft Medoid GDC incorporates diffusion computations, rendering it less scalable (Geisler et al. 2021), while EvenNet demands a minimum of 70GB of GPU memory, exceeding the capacity of our experiment device, which is limited to 48GB. It’s worth noting that the speed advantage of SFR-GNN is particularly pronounced in large-scale graphs compared to tiny graphs, which demonstrates the simplicity and effectiveness of SFR-GNN.

Ablation Study. To validate the effectiveness of the proposed method, we propose two variants: 1) **SFR-GNN w/o CL**: This variant lacks the contrastive learning technique and directly fine-tunes the model using the modified adjacency matrix. 2) **SFR-GNN w/o Fin**: This variant lacks the whole structure fine-tuning stage, thereby degenerating into a Multilayer Perceptron (MLP). The results in Fig. 3 (a) and Fig. 3 (b) show that the accuracies of both variants are consistently lower than that of SFR-GNN across all attack ratios. SFR-GNN w/o CL can not performer SFR-GNN and achieves suboptimal results which proves the Lemma. 3 and Theorem. 2 in Sec. . SFR-GNN w/o Fin achieved the worst results because the learned representations only contained attribute information without structure information.

Additionally, to validate the effectiveness of the proposed InterNAA, we replace it with commonly used augmentations: Node Dropping (SFR-GNN w/ND), Edge Removing (SFR-GNN w/ER), and Feature Masking (SFR-GNN w/FM). Besides, we provide a variant SFR-GNN w/Ran which replaces InterNAA with random node attributes sampling. The comparison results are shown in Fig. 3 (a) and

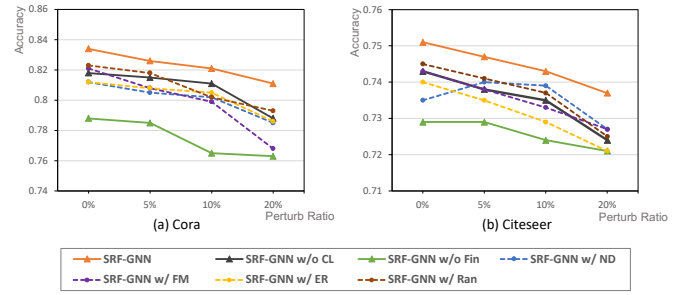


Figure 3: Ablation studies of SFR-GNN.

Fig. 3 (b). Clearly, InterNAA outperforms other augmentations. We believe this is because other augmentations randomly perturb the elements of the graph and cannot prevent the inflect of contaminated mutual information during the fine-tuning stage. Additionally, to validate the effectiveness of the proposed InterNAA, we replace it with commonly used augmentations: Node Dropping (SFR-GNN w/ND), Edge Removing (SFR-GNN w/ER), Feature Masking (SFR-GNN w/FM) and random node attribute sampling (SFR-GNN w/Ran). The comparison results are shown in Fig. 3 (a) and Fig. 3 (b). Clearly, InterNAA outperforms the other augmentations. We believe this is because other augmentations randomly perturb the elements of the graph and cannot prevent the influence of contaminated mutual information during the fine-tuning stage.

Conclusion

In this paper, we propose a novel robust GNN: Simple and Fast Robust Graph Neural Network (SFR-GNN) against structural attacks. SFR-GNN utilizes the proposed “attributes pre-training and structure fine-tuning” strategy, without the need for purification of the modified structures, thus significantly reducing computational overhead and avoiding the introduction of additional hyper-parameters. We conduct both theoretical analysis and numerical experiments to validate the effectiveness of SFR-GNN. Experimental results demonstrate that SFR-GNN achieves robustness comparable to state-of-the-art baselines while delivering a 50%-136% improvement in runtime speed. Addition-

ally, it exhibits superior scalability on large-scale datasets. This makes SFR-GNN a promising solution for applications requiring reliable and efficient GNNs in adversarial settings.

A. Theoretical Analyses

A.1 Proof of Lemma. 1

Lemma 4. *Structural attacks degrade GNNs' performance through generating the modified adjacency matrix \mathbf{A}' to contaminate the mutual information between the labels \mathbf{Y} and \mathbf{A}' conditioned by \mathbf{X} , which essentially uses the mutual information $I(\mathbf{A}'; \mathbf{Y}|\mathbf{X})$.*

Proof. For a GNN model f_θ parameterized by θ , the objective is to predict labels \mathbf{Y} as accurately as possible by taking node features \mathbf{X} and adjacency matrix \mathbf{A} as inputs. From the perspective of information theory, this objective can be viewed as minimizing the conditional entropy $H(\mathbf{Y}|f_\theta(\mathbf{X}, \mathbf{A}))$:

$$\min \mathcal{L}_{tr}(f_\theta(\mathbf{X}, \mathbf{A}), \mathbf{Y}) \Rightarrow \min H(\mathbf{Y}|f_\theta(\mathbf{X}, \mathbf{A})). \quad (5)$$

The conditional entropy $H(\mathbf{Y}|f_\theta(\mathbf{X}, \mathbf{A}))$ measures the uncertainty of \mathbf{Y} given the $f_\theta(\mathbf{X}, \mathbf{A})$. An effective $f_\theta(\mathbf{X}, \mathbf{A})$ should be able to predict \mathbf{Y} with high probability, meaning uncertainty is low. Thus the above equation holds.

According to the principles of mutual information, we have:

$$I(f_\theta(\mathbf{X}, \mathbf{A}); \mathbf{Y}) = H(\mathbf{Y}) - H(\mathbf{Y}|f_\theta(\mathbf{X}, \mathbf{A})). \quad (6)$$

$H(\mathbf{Y})$ is the information entropy of labels which is determined by \mathbf{Y} and fixed. Thus minimizing $H(\mathbf{Y}|f_\theta(\mathbf{X}, \mathbf{A}))$ is actually maximizing $I(f_\theta(\mathbf{X}, \mathbf{A}); \mathbf{Y})$:

$$\min H(\mathbf{Y}|f_\theta(\mathbf{X}, \mathbf{A})) \Rightarrow \max I(f_\theta(\mathbf{X}, \mathbf{A}); \mathbf{Y}). \quad (7)$$

Therefore, the learning objective of GNN is actually maximizing the mutual information $I(f_\theta(\mathbf{X}, \mathbf{A}); \mathbf{Y})$:

$$\min \mathcal{L}_{tr}(f_\theta(\mathbf{X}, \mathbf{A}), \mathbf{Y}) \Rightarrow \max I(f_\theta(\mathbf{X}, \mathbf{A}); \mathbf{Y}). \quad (8)$$

maximizing the mutual information between the labels \mathbf{Y} and the model output $f_\theta(\mathbf{X}, \mathbf{A})$, which is actually learning information from the mutual information between the labels \mathbf{Y} and the joint distribution (\mathbf{X}, \mathbf{A}) because most of GNNs have been demonstrated that satisfy the injective property (Xu et al. 2018) or linear assumption (Wu et al. 2019; Zhu and Koniusz 2021):

$$\max I(f_\theta(\mathbf{X}, \mathbf{A}); \mathbf{Y}) \Rightarrow \max I((\mathbf{X}, \mathbf{A}); \mathbf{Y}). \quad (9)$$

Furthermore, according to the properties of mutual information, we can decompose $I((\mathbf{X}, \mathbf{A}); \mathbf{Y})$ into $I(\mathbf{X}; \mathbf{Y})$ and $I(\mathbf{A}; \mathbf{Y}|\mathbf{X})$:

$$I((\mathbf{X}, \mathbf{A}); \mathbf{Y}) = I(\mathbf{X}; \mathbf{Y}) + I(\mathbf{A}; \mathbf{Y}|\mathbf{X}). \quad (10)$$

Thus GNNs' learning objective is actually maximizing $I((\mathbf{X}, \mathbf{A}); \mathbf{Y})$ and can be decomposed into the maximization of $I(\mathbf{X}; \mathbf{Y})$ and the maximization of $I(\mathbf{A}; \mathbf{Y}|\mathbf{X})$.

The goal of the structural attacker is degrading the prediction accuracy of f_θ as much as possible. To achieve this goal, the structural attacker employs perturbation δ^* to divert the outputs of the victim GNN from the true labels \mathbf{Y} to erroneous predictions \mathbf{Y}' , which can be formulated as the minimization of $I((\mathbf{X}, (\mathbf{A} + \delta)); \mathbf{Y})$ and the maximization of $I((\mathbf{X}, (\mathbf{A} + \delta)); \mathbf{Y}')$:

$$\mathbf{A}' = \mathbf{A} + \delta^*, \quad (11)$$

$$\delta^* = \arg \min_{\delta} I((\mathbf{X}, (\mathbf{A} + \delta)); \mathbf{Y}). \quad (12)$$

According to Eq. (9) and Eq. (10), the above goal can be rewritten as:

$$\delta^* = \arg \min_{\delta} I(\mathbf{X}; \mathbf{Y}) + I((\mathbf{A} + \delta); \mathbf{Y}|\mathbf{X}). \quad (13)$$

Due to $I(\mathbf{X}; \mathbf{Y})$ is irrelevant and independent to the structure perturbation δ thus the actual goal of the attacker is:

$$\delta^* = \arg \min_{\delta} I((\mathbf{A} + \delta); \mathbf{Y}|\mathbf{X}). \quad (14)$$

Hence, the structural attacker essentially aims to minimize $I(\mathbf{A}'; \mathbf{Y}|\mathbf{X})$ to hinder the victim GNN from extracting adequate information from this mutual information and compel victim GNN to make wrong predictions. We describe this scenario as the contamination of mutual information $I(\mathbf{A}'; \mathbf{Y}|\mathbf{X})$. \square

A.2 Proof of Lemma. 2

Lemma 5. For any $\mathbf{X}' \neq \mathbf{X}$, where $\mathbf{X}', \mathbf{X} \in \mathbb{R}^{n \times d}$, the mutual information $I(\mathbf{A}'; \mathbf{Y}|\mathbf{X}')$ is less harmful to GNNs than $I(\mathbf{A}'; \mathbf{Y}|\mathbf{X})$.

Proof. According to properties of mutual information, $I(\mathbf{A}'; \mathbf{Y}|\mathbf{X})$ can be reformulated as:

$$I(\mathbf{A}'; \mathbf{Y}|\mathbf{X}) = I(\mathbf{A}'; \mathbf{Y}) - I(\mathbf{A}'; \mathbf{Y}; \mathbf{X}), \quad (15)$$

where $I(\mathbf{A}'; \mathbf{Y}; \mathbf{X})$ is the mutual information between \mathbf{A}' , \mathbf{Y} and \mathbf{X} .

Lemma. 1 indicates attackers degrade victim GNNs' performances by minimizing $I(\mathbf{A}'; \mathbf{Y}|\mathbf{X})$. Eq. (15) indicates minimizing $I(\mathbf{A}'; \mathbf{Y}|\mathbf{X})$ can be achieved through minimizing $I(\mathbf{A}'; \mathbf{Y})$ and maximizing $I(\mathbf{A}'; \mathbf{Y}; \mathbf{X})$. Assuming the upper bound of $I(\mathbf{A}'; \mathbf{Y}; \mathbf{X})$ is donated as τ , an ideal attacker, in pursuit of minimizing $I(\mathbf{A}'; \mathbf{Y}|\mathbf{X})$, fulfills $I(\mathbf{A}'; \mathbf{Y}; \mathbf{X}) = \tau$.

Similarly, for $I(\mathbf{A}'; \mathbf{Y}|\mathbf{X}')$, we have:

$$I(\mathbf{A}'; \mathbf{Y}|\mathbf{X}') = I(\mathbf{A}'; \mathbf{Y}) - I(\mathbf{A}'; \mathbf{Y}; \mathbf{X}'). \quad (16)$$

Therefore, the difference between $I(\mathbf{A}'; \mathbf{Y}|\mathbf{X}')$ and $I(\mathbf{A}'; \mathbf{Y}|\mathbf{X})$ is referred as:

$$\begin{aligned} & I(\mathbf{A}'; \mathbf{Y}|\mathbf{X}') - I(\mathbf{A}'; \mathbf{Y}|\mathbf{X}) \\ &= I(\mathbf{A}'; \mathbf{Y}) - I(\mathbf{A}'; \mathbf{Y}; \mathbf{X}') - I(\mathbf{A}'; \mathbf{Y}) + I(\mathbf{A}'; \mathbf{Y}; \mathbf{X}) \\ &= I(\mathbf{A}'; \mathbf{Y}; \mathbf{X}) - I(\mathbf{A}'; \mathbf{Y}; \mathbf{X}') \\ &= \tau - I(\mathbf{A}'; \mathbf{Y}; \mathbf{X}') \geq 0, \end{aligned} \quad (17)$$

because $I(\mathbf{A}'; \mathbf{Y}; \mathbf{X}')$ is less than the upper bound τ . The above equation demonstrates $I(\mathbf{A}'; \mathbf{Y}|\mathbf{X}') \geq I(\mathbf{A}'; \mathbf{Y}|\mathbf{X})$. That implies $I(\mathbf{A}'; \mathbf{Y}|\mathbf{X}')$ retains more information that could potentially be exploited by GNNs. Thus $I(\mathbf{A}'; \mathbf{Y}|\mathbf{X}')$ is less harmful to GNNs. \square

A.3 Proof of Theorem. 1

Theorem 3. SFR-GNN's pre-training stage maximizes the mutual information $I(\mathbf{Z}_p; \mathbf{Y})$ between \mathbf{Z}_p and \mathbf{Y} , where $I(\mathbf{A}'; \mathbf{Y}|\mathbf{Z}_p)$ is less harmful to GNNs compared to $I(\mathbf{A}'; \mathbf{Y}|\mathbf{X})$.

Proof. \mathbf{Z}_p is solely learned from node attributes \mathbf{X} , thus

Minimizing the pre-training loss \mathcal{L}_p is actually pursuing the maximizing

Minimizing the pre-training loss function \mathcal{L}_p aims at ensuring accurate predictions for all nodes, which is maximizing the conditional probability $P(\mathbf{Y}_v = c_v | \mathbf{Z}_{p(v)})$ for any node v . It equals minimizing the information entropy $H(\mathbf{Y}|\mathbf{Z}_p)$. Due to the properties of mutual information, we have:

$$I(\mathbf{Y}; \mathbf{Z}_p) = H(\mathbf{Y}) - H(\mathbf{Y}|\mathbf{Z}_p). \quad (18)$$

Therefore, minimizing $H(\mathbf{Y}|\mathbf{Z}_p)$ equals to maximize $I(\mathbf{Y}; \mathbf{Z}_p)$, and naturally, minimizing \mathcal{L}_p is equal to maximize $I(\mathbf{Y}; \mathbf{Z}_p)$. Additionally, according to the Lemma. 2, we have:

$$I(\mathbf{A}'; \mathbf{Y}|\mathbf{Z}_p) \geq I(\mathbf{A}'; \mathbf{Y}|\mathbf{X}) = \tau, \quad (19)$$

which proves $I(\mathbf{A}'; \mathbf{Y}|\mathbf{Z}_p)$ is less harmful to GNNs compared to $I(\mathbf{A}'; \mathbf{Y}|\mathbf{X})$. \square

A.4 Proof of Lemma. 3

Lemma 6. There exists an overlap between $I(\mathbf{A}'; \mathbf{Y}|\mathbf{Z}_p)$ and $I(\mathbf{A}'; \mathbf{Y}|\mathbf{X})$, which consequently leads to the contamination of $I(\mathbf{A}'; \mathbf{Y}|\mathbf{Z}_p)$.

Proof. We first demonstrate the existence of overlap between $I(\mathbf{A}'; \mathbf{Y}|\mathbf{Z}_p)$ and $I(\mathbf{A}'; \mathbf{Y}|\mathbf{X})$. Due to the properties of mutual information, we have:

$$\begin{aligned} & I(\mathbf{A}'; \mathbf{Y}|\mathbf{Z}_p) \subset I(\mathbf{A}'; \mathbf{Y}), \quad I(\mathbf{A}; \mathbf{Y}|\mathbf{Z}_p) \subset I(\mathbf{A}'; \mathbf{Y}) \\ & I(\mathbf{A}'; \mathbf{Y}|\mathbf{X}) = I(\mathbf{A}'; \mathbf{Y}) - I(\mathbf{A}'; \mathbf{Y}; \mathbf{X}), \end{aligned} \quad (20)$$

thus $I(\mathbf{A}'; \mathbf{Y}|\mathbf{Z}_p) \cup I(\mathbf{A}'; \mathbf{Y}|\mathbf{X}) = 0$ if and only if $I(\mathbf{A}'; \mathbf{Y}|\mathbf{Z}_p) = I(\mathbf{A}'; \mathbf{Y}; \mathbf{X})$, which is not always feasible to guarantee in practice. Consequently, we have:

$$I(\mathbf{A}'; \mathbf{Y}|\mathbf{Z}_p) \cup I(\mathbf{A}'; \mathbf{Y}|\mathbf{X}) \neq 0. \quad (21)$$

To demonstrate the possible contamination of $I(\mathbf{A}'; \mathbf{Y}|\mathbf{Z}_p)$, we abstract the f_{θ} as a Simplified Graph Convolution (SGC), where f_{θ} is a linearized function with parameter W_{θ} :

$$f_{\theta}(\mathbf{X}) = \mathbf{X} \cdot W_{\theta}. \quad (22)$$

In structure fine-tuning, the SGC model with K layers convolution can be formulated as:

$$f_{\theta_p}(\mathbf{X}) = (\mathbf{A}')^k \cdot \mathbf{X} \cdot W_{\theta_p}, \quad \mathbf{Z}_p = \mathbf{X} \cdot W_{\theta_p}. \quad (23)$$

Suppose the parameter update during the fine-tuning process is denoted as ΔW , the output of the function f_{θ_p} after fine-tuning is:

$$\begin{aligned} f_{\theta_p}(\mathbf{X}) &= (\mathbf{A}')^k \cdot \mathbf{X} \cdot (W_{\theta_p} + \Delta W) \\ &= (\mathbf{A}')^k \cdot \mathbf{X} \cdot W_{\theta_p} + (\mathbf{A}')^k \cdot \mathbf{X} \cdot \Delta W \\ &= (\mathbf{A}')^k \cdot \mathbf{Z}_p + (\mathbf{A}')^k \cdot \mathbf{X} \cdot \Delta W. \end{aligned} \quad (24)$$

The term following the addition can be regarded as the embedding output by a GNN with parameters ΔW , taking \mathbf{X} and \mathbf{A}' as inputs. It is actually equal to a victim GNN, whose embedding is contaminated by $I(\mathbf{A}'; \mathbf{Y}|\mathbf{X})$. That implies that during the fine-tuning stage, \mathbf{Z}_p unavoidably incorporates the influence of $I(\mathbf{A}'; \mathbf{Y}|\mathbf{X})$. \square

A.5 Proof of Theorem. 2

Theorem 4. *SFR-GNN's structure fine-tuning stage maximizes $I(\mathbf{A}'; \mathbf{Y}|\mathbf{Z}_p)$ to learn structural information and align it to $I(\mathbf{A}'; \mathbf{Y}|\mathbf{X}_{\text{inter}})$ to prevent from being contaminated.*

Proof. Following the principles of mutual information, we hold:

$$\begin{aligned} I((\mathbf{X}_{\text{inter}}, \mathbf{A}'); \mathbf{Y}) &= I(\mathbf{A}'; \mathbf{Y}) + I(\mathbf{X}_{\text{inter}}; \mathbf{Y}|\mathbf{A}') \\ &\leq I(\mathbf{A}'; \mathbf{Y}) + I(\mathbf{X}_{\text{inter}}; \mathbf{Y}) \\ &\leq I(\mathbf{A}'; \mathbf{Y}) + H(\mathbf{Y}) - H(\mathbf{Y}|\mathbf{X}_{\text{inter}}), \end{aligned} \quad (25)$$

where $H(\cdot)$ is information entropy. $H(\mathbf{Y})$ is the information entropy of labels which is solely determined by \mathbf{Y} and independent of \mathbf{X} . As for the conditional entropy $H(\mathbf{Y}|\mathbf{X}_{\text{inter}})$ which measures the uncertainty of \mathbf{Y} given the $\mathbf{X}_{\text{inter}}$. An effective $\mathbf{X}_{\text{inter}}$ should be able to predict \mathbf{Y} well, meaning that knowing $\mathbf{X}_{\text{inter}}$ allows us to determine the value of \mathbf{Y} with great certainty. However, InterNAA intentionally replaces the node features contained in $\mathbf{X}_{\text{inter}}$ with features from nodes of different classes, leading to an inability to accurately predict \mathbf{Y} through $\mathbf{X}_{\text{inter}}$, leading to a large $H(\mathbf{Y}|\mathbf{X}_{\text{inter}})$. Due to the non-negative characteristic of mutual information:

$$I(\mathbf{X}_{\text{inter}}; \mathbf{Y}) = H(\mathbf{Y}) - H(\mathbf{Y}|\mathbf{X}_{\text{inter}}) \geq 0. \quad (26)$$

When the conditional entropy is sufficiently large, the mutual information $I(\mathbf{X}_{\text{inter}}; \mathbf{Y})$ tends to be 0, at which point the above equation fulfills:

$$I((\mathbf{X}_{\text{inter}}, \mathbf{A}'); \mathbf{Y}) \Rightarrow I(\mathbf{A}'; \mathbf{Y}). \quad (27)$$

which means $I(f(\mathbf{X}_{\text{inter}}, \mathbf{A}'); \mathbf{Y})$ is approximately equal to $I(\mathbf{A}'; \mathbf{Y})$, and align \mathbf{Z}_p with $f(\mathbf{X}_{\text{inter}}, \mathbf{A}')$ is actually let \mathbf{Z}_p combine structure information from $I(\mathbf{A}'; \mathbf{Y})$ instead of the contaminated $I(\mathbf{A}'; \mathbf{Y}|\mathbf{X})$. \square

B. Experimental Details

B.1 Dataset Details

In this section, we describe in detail the graph datasets used in this paper. We report statistics for these datasets in (Table. 5). Each dataset is described below:

Cora, Citeseer, and Pubmed are three benchmark datasets commonly used in graph neural network research, each representing different scenarios in scientific literature citation networks. In these networks, nodes represent papers; edges indicate citations of that paper by other papers, and node labels are the academic topics of the papers. Among them, Cora and Citeseer datasets use bag-of-words 0/1 vectors to represent terms present in the paper as node features, while PubMed uses TF-IDF vectors as its node features.

Chameleon and Squirrel are heterophilic graph datasets commonly used in GNN. Chameleon is a web page link network where the nodes represent web pages, and the edges represent hyperlinks between them. On the other hand, Squirrel is a Wikipedia article network, with nodes denoting the articles and edges indicating the hyperlinks between them. In contrast to the previously mentioned datasets, the node features in Chameleon and Squirrel are not based on text content but rather describe the properties of the nodes (web pages and Wikipedia articles).

B.2 Baseline Descriptions

- **GCN:** It is the most representative GNN model which utilizes the graph convolutional layer to propagate node features with the low-pass filter and smooth the features of connected node pairs.
- **RGCN:** It learns the Gaussian distributions for each node feature and employs an attention mechanism to alleviate the potential malicious influence of nodes with high variance.
- **GCN-Jaccard:** It sanitizes the graph data by pruning links that connect nodes with low values of Jaccard similarity of node attributes.
- **ProGNN:** It jointly learns a structural graph and a robust GNN model from the modified graph guided by the three properties: low-rank, sparsity and feature smoothness.
- **SimP-GCN:** It utilizes a kNN graph to capture the node similarity and enhance the node representation of the GNN.
- **STABLE:** It utilizes the homophily assumption to refine the modified structure and combine contrastive learning techniques to remove adversarial edges. Finally, an advanced GCN is used to predict node labels.
- **EvenNet:** By applying balance theory, it obtains a more robust spectral graph filter under homophily change by ignoring messages from odd-order neighbors and only using even-order terms.
- **GADC:** Inspired by graph diffusion convolution, it proposes a novel min-max optimization to perturb graph structure based on Laplacian distance.
- **NoisyGCN:** It injects random noise into the hidden states of the GCN to improve the robustness against structure attacks.

Datasets	Hom. Ratio	Nodes	Edges	Features	Classes
Cora	0.81	2,708	5,429	1,433	7
Cora-ml	0.80	2,810	7,981	2,879	7
Citeseer	0.74	3,327	4,732	3,703	6
Pubmed	0.80	19,717	44,338	500	3
Chameleon	0.23	2,277	36,101	2,325	4
Squirrel	0.22	5,201	198,353	2,089	5
ogbn-Arxiv	-	169,343	1,157,799	128	40
ogbn-Products	-	2,449,029	61,859,076	100	47

Table 5: Dataset Statistics.

Dataset	Chameleon				Squirrel			
Ptb	0	5	10	20	0	5	10	20
GCN	56.3±1.6	52.1±2.0	49.2±2.2	40.6±1.9	41.2±0.6	38.4±0.5	36.8±0.7	34.3±0.3
RGCN	54.7±1.5	53.8±1.4	51.0±1.7	41.3±1.8	40.5±0.2	38.6±0.4	34.3±0.6	32.9±0.2
GCN-Jaccard	-	-	-	-	-	-	-	-
Pro-GNN	56.1±0.4	54.8±0.9	50.2±1.3	48.1±1.0	42.0±0.1	39.7±0.5	37.6±0.9	36.1±1.0
SimP-GCN	55.6±1.4	54.0±0.9	50.5±1.7	46.4±1.8	40.9±0.2	38.3±0.6	35.1±0.5	32.3±0.7
EvenNet	57.3±1.2	<u>55.6±1.9</u>	52.5±2.0	49.0±2.2	41.3±0.9	40.0±1.1	38.8±0.7	<u>37.0±1.3</u>
STABLE	54.0±1.3	52.2±1.9	49.1±2.6	39.9±2.8	<u>41.4±0.7</u>	37.7±0.8	35.6±0.9	31.5±0.7
GADC	56.8±2.2	54.7±1.9	51.0±2.4	<u>48.8±1.6</u>	41.1±0.9	38.9±1.1	37.7±0.5	36.9±1.1
Noisy-GCN	56.5±1.3	53.0±1.1	50.1±1.6	46.2±1.8	40.0±0.7	37.3±0.8	36.1±0.9	34.7±1.3
SFR-GNN	<u>57.0±3.7</u>	55.9±2.5	<u>52.0±1.8</u>	<u>48.8±1.7</u>	40.9±1.0	<u>39.7±1.3</u>	<u>38.1±0.9</u>	37.4±0.9

Table 6: Robustness Comparison Mettack.

B.3 Configuration for Baselines and Datasets

The attack ratios of the two methods are set to 5%, 10%, 15%, 20%, 25%. Our baseline methods include representatives of vanilla GNNs, such as GCN (Kipf and Welling 2017), as well as typical adaptive aggregation methods like RGCN (Zhu et al. 2019) and SimP-GCN (Jin et al. 2021), purifying methods including GCN-SVD (Entezari et al. 2020b), GCN-Jaccard (Entezari et al. 2020a), Pro-GNN (Jin et al. 2020), and the most recent method, HANG-quad (Zhao et al. 2023). Detailed information about baselines can be found in the Appendix. .

We follow the data splitting method of DeepRobust: randomly selecting 10% of the nodes for training, 10% for validation, and the remaining 80% for testing. In accordance with the victim GNN configuration of DeepRobust, SFR-GNN is configured with 2 layers, 16 hidden units and a dropout ratio of 0.5, while the learning rate was set to 0.01. We search the training epoch of SFR-GNN over 200,300,400 and fine-tuning epoch over 3, 10, 20, 50. For PubMed, to mitigate the computational overhead associated with InterNAA sampling, we random select 20% of nodes in the training set to replace their attributes with those from nodes belonging to different classes.

C. Defense Performance in Heterophilic Graphs

Table. 6 demonstrates the performance of the proposed model and baselines on two typical heterophilic graph datasets. We report the average classification accuracy (\pm standard deviation) of 10 runs with different perturbation ra-

tios (ptb). The best and second-best results are highlighted in bold and underlined, respectively. Due to errors reported by GCN-Jaccard on these two datasets, its results are represented by "-". The proposed model achieves the best or second-best performance in most cases, demonstrating the robustness of SFR-GNN on heterophilic graphs.

References

- Alon, N.; Avdiukhin, D.; Elboim, D.; Fischer, O.; and Yaroslavtsev, G. 2024. Optimal Sample Complexity of Contrastive Learning. In *The Twelfth International Conference on Learning Representations*.
- Bojchevski, A.; and Günnemann, S. 2019. Adversarial attacks on node embeddings via graph poisoning. In *International Conference on Machine Learning*, 695–704. PMLR.
- Chen, M.; Wei, Z.; Ding, B.; Li, Y.; Yuan, Y.; Du, X.; and Wen, J.-R. 2020. Scalable graph neural networks via bidirectional propagation. *Advances in neural information processing systems*, 33: 14556–14566.
- Chen, T.; Zhou, K.; Duan, K.; Zheng, W.; Wang, P.; Hu, X.; and Wang, Z. 2022. Bag of tricks for training deeper graph neural networks: A comprehensive benchmark study. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(3): 2769–2781.
- Ennadir, S.; Abbahaddou, Y.; Lutzeyer, J. F.; Vazirgiannis, M.; and Boström, H. 2024. A Simple and Yet Fairly Effective Defense for Graph Neural Networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 21063–21071.

- Entezari, N.; Al-Sayouri, S. A.; Darvishzadeh, A.; and Papalexakis, E. E. 2020a. All you need is low (rank) defending against adversarial attacks on graphs. In *Proceedings of the 13th international conference on web search and data mining*, 169–177.
- Entezari, N.; Al-Sayouri, S. A.; Darvishzadeh, A.; and Papalexakis, E. E. 2020b. All you need is low (rank) defending against adversarial attacks on graphs. In *Proceedings of the 13th international conference on web search and data mining*, 169–177.
- Geisler, S.; Schmidt, T.; Şirin, H.; Zügner, D.; Bojchevski, A.; and Günnemann, S. 2021. Robustness of Graph Neural Networks at Scale. In *Neural Information Processing Systems, NeurIPS*.
- Giles, C. L.; Bollacker, K. D.; and Lawrence, S. 1998. CiteSeer: An automatic citation indexing system. In *Proceedings of the third ACM conference on Digital libraries*, 89–98.
- Hu, W.; Fey, M.; Zitnik, M.; Dong, Y.; Ren, H.; Liu, B.; Catasta, M.; and Leskovec, J. 2020. Open graph benchmark: Datasets for machine learning on graphs. *Advances in neural information processing systems*, 33: 22118–22133.
- Hu, X.; Chen, H.; Chen, H.; Liu, S.; Li, X.; Zhang, S.; Wang, Y.; and Xue, X. 2023. Cost-Sensitive GNN-Based Imbalanced Learning for Mobile Social Network Fraud Detection. *IEEE Transactions on Computational Social Systems*.
- Hussain, H.; Duricic, T.; Lex, E.; Helic, D.; Strohmaier, M.; and Kern, R. 2021. Structack: Structure-based adversarial attacks on graph neural networks. In *Proceedings of the 32nd ACM Conference on Hypertext and Social Media*, 111–120.
- Jin, W.; Derr, T.; Wang, Y.; Ma, Y.; Liu, Z.; and Tang, J. 2021. Node similarity preserving graph convolutional networks. In *Proceedings of the 14th ACM international conference on web search and data mining*, 148–156.
- Jin, W.; Ma, Y.; Liu, X.; Tang, X.; Wang, S.; and Tang, J. 2020. Graph structure learning for robust graph neural networks. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, 66–74.
- Kipf, T. N.; and Welling, M. 2017. Semi-supervised classification with graph convolutional networks. *International Conference on Learning Representations (ICLR)*.
- Lai, Y.; Zhu, Y.; Fan, W.; Zhang, X.; and Zhou, K. 2023. Towards adversarially robust recommendation from adaptive fraudster detection. *IEEE Transactions on Information Forensics and Security*.
- Lei, R.; Wang, Z.; Li, Y.; Ding, B.; and Wei, Z. 2024. EvenNet: ignoring odd-hop neighbors improves robustness of graph neural networks. In *Proceedings of the 36th International Conference on Neural Information Processing Systems, NIPS '22*. Red Hook, NY, USA: Curran Associates Inc. ISBN 9781713871088.
- 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 *Proceedings of the 28th ACM SIGKDD conference on knowledge discovery and data mining*, 925–935.
- Li, Y.; Jin, W.; Xu, H.; and Tang, J. 2020. Deeprobust: A pytorch library for adversarial attacks and defenses. *arXiv preprint arXiv:2005.06149*.
- Liu, S.; Chen, J.; Fu, T.; Lin, L.; Zitnik, M.; and Wu, D. 2024. Graph Adversarial Diffusion Convolution. In *International Conference on Machine Learning*.
- Liu, T.; Wang, Y.; Ying, R.; and Zhao, H. 2023. MuSe-GNN: Learning Unified Gene Representation From Multi-modal Biological Graph Data. In Oh, A.; Neumann, T.; Globerson, A.; Saenko, K.; Hardt, M.; and Levine, S., eds., *Advances in Neural Information Processing Systems*, volume 36, 24661–24677. Curran Associates, Inc.
- Liu, Z.; Luo, Y.; Wu, L.; Liu, Z.; and Li, S. Z. 2022. Towards Reasonable Budget Allocation in Untargeted Graph Structure Attacks via Gradient Debias. In *Advances in Neural Information Processing Systems*.
- McCallum, A. K.; Nigam, K.; Rennie, J.; and Seymore, K. 2000. Automating the construction of internet portals with machine learning. *Information Retrieval*, 3: 127–163.
- Sen, P.; Namata, G.; Bilgic, M.; Getoor, L.; Galligher, B.; and Eliassi-Rad, T. 2008. Collective classification in network data. *AI magazine*, 29(3): 93–93.
- Wang, Y.; Jin, J.; Zhang, W.; Yu, Y.; Zhang, Z.; and Wipf, D. 2021. Bag of tricks for node classification with graph neural networks. *arXiv preprint arXiv:2103.13355*.
- Wu, F.; Souza, A.; Zhang, T.; Fifty, C.; Yu, T.; and Weinberger, K. 2019. Simplifying graph convolutional networks. In *International conference on machine learning*, 6861–6871. PMLR.
- Xu, K.; Chen, H.; Liu, S.; Chen, P.-Y.; Weng, T.-W.; Hong, M.; and Lin, X. 2019. Topology attack and defense for graph neural networks: An optimization perspective. *arXiv preprint arXiv:1906.04214*.
- Xu, K.; Hu, W.; Leskovec, J.; and Jegelka, S. 2018. How Powerful are Graph Neural Networks? In *International Conference on Learning Representations*.
- Zhang, H.; Wu, Q.; Zhang, S.; Yan, J.; Wipf, D.; and Yu, P. S. 2022a. ESCo: Towards Provably Effective and Scalable Contrastive Representation Learning.
- Zhang, S.; Chen, H.; Sun, X.; Li, Y.; and Xu, G. 2022b. Unsupervised graph poisoning attack via contrastive loss back-propagation. In *Proceedings of the ACM Web Conference 2022*, 1322–1330.
- Zhang, X.; and Gan, M. 2024. Hi-GNN: hierarchical interactive graph neural networks for auxiliary information-enhanced recommendation. *Knowledge and Information Systems*, 66(1): 115–145.
- Zhang, X.; and Zitnik, M. 2020. GNNGUARD: Defending Graph Neural Networks against Adversarial Attacks. In *Proceedings of the 34th International Conference on Neural Information Processing Systems, NIPS'20*. Red Hook, NY, USA: Curran Associates Inc. ISBN 9781713829546.
- Zhao, K.; Kang, Q.; Song, Y.; She, R.; Wang, S.; and Tay, W. P. 2023. Adversarial Robustness in Graph Neural Networks: A Hamiltonian Energy Conservation Approach. In *Advances in Neural Information Processing Systems*. New Orleans, USA.

Zhao, K.; Zhou, H.; Zhu, Y.; Zhan, X.; Zhou, K.; Li, J.; Yu, L.; Yuan, W.; and Luo, X. 2021. Structural Attack against Graph Based Android Malware Detection. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security, CCS '21*, 3218–3235. New York, NY, USA: Association for Computing Machinery. ISBN 9781450384544.

Zhou, J.; Lai, Y.; Ren, J.; and Zhou, K. 2023. Black-Box Attacks against Signed Graph Analysis via Balance Poisoning. *arXiv preprint arXiv:2309.02396*.

Zhu, D.; Zhang, Z.; Cui, P.; and Zhu, W. 2019. Robust graph convolutional networks against adversarial attacks. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, 1399–1407.

Zhu, H.; and Koniusz, P. 2021. Simple spectral graph convolution. In *International conference on learning representations*.

Zhu, Y.; Lai, Y.; Zhao, K.; Luo, X.; Yuan, M.; Ren, J.; and Zhou, K. 2022a. Binarizedattack: Structural poisoning attacks to graph-based anomaly detection. In *2022 IEEE 38th International Conference on Data Engineering (ICDE)*, 14–26. IEEE.

Zhu, Y.; Lai, Y.; Zhao, K.; Luo, X.; Yuan, M.; Ren, J.; and Zhou, K. 2022b. BinarizedAttack: Structural Poisoning Attacks to Graph-based Anomaly Detection. In *2022 IEEE 38th International Conference on Data Engineering (ICDE)*, 14–26.

Zhu, Y.; Michalak, T.; Luo, X.; Zhang, X.; and Zhou, K. 2024. Towards Secrecy-Aware Attacks Against Trust Prediction in Signed Social Networks. *IEEE Transactions on Information Forensics and Security*.

Zhu, Y.; Tong, L.; Li, G.; Luo, X.; and Zhou, K. 2023. FocusedCleaner: Sanitizing Poisoned Graphs for Robust GNN-based Node Classification. *IEEE Transactions on Knowledge and Data Engineering*.

Zhu, Y.; Xu, W.; Zhang, J.; Liu, Q.; Wu, S.; and Wang, L. 2021. Deep graph structure learning for robust representations: A survey. *arXiv preprint arXiv:2103.03036*, 14: 1–1.

Zügner, D.; Akbarnejad, A.; and Günnemann, S. 2018. Adversarial attacks on neural networks for graph data. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, 2847–2856.

Zügner, D.; and Günnemann, S. 2019. Adversarial Attacks on Graph Neural Networks via Meta Learning. In *International Conference on Learning Representations*.