Robust Heterogeneous Graph Neural Networks against Adversarial Attacks

Mengmei Zhang[1], Xiao Wang[1], Meiqi Zhu[1], Chuan Shi[1*], Zhiqiang Zhang[2], Jun Zhou[2]

[1]Beijing University of Posts and Telecommunications
[2]Ant Group, Hangzhou, China
{zhangmm, xiaowang, zhumeiqi, shichuan}@bupt.edu.cn, {lingyao.zzq, jun.zhoujun}@antfin.com

Abstract

Heterogeneous Graph Neural Networks (HGNNs) have drawn increasing attention in recent years and achieved outstanding performance in many tasks. However, despite their wide use, there is currently no understanding of their robustness to adversarial attacks. In this work, we first systematically study the robustness of HGNNs and show that they can be easily fooled by adding the adversarial edge between the target node and large-degree node (i.e., hub). Furthermore, we show two key reasons for such vulnerabilities of HGNNs: one is *perturbation enlargement effect*, i.e., HGNNs, failing to encode transiting probability, will enlarge the effect of the adversarial hub in comparison of GCNs, and the other is *soft attention mechanism*, i.e., such mechanism assigns positive attention values to obviously unreliable neighbors. Based on the two facts, we propose a novel robust HGNN framework *RoHe* against topology adversarial attacks by equipping an attention purifier, which can prune malicious neighbors based on topology and feature. Specifically, to eliminate the perturbation enlargement, we introduce the metapath-based transiting probability as the prior criterion of the purifier, restraining the confidence of malicious neighbors from adversarial hub. Then the purifier learns to mask out neighbors with low confidence, thus can effectively alleviate the negative effect of malicious neighbors in the soft attention mechanism. Extensive experiments on different benchmark datasets for multiple HGNNs are conducted, where the considerable improvement of HGNNs under adversarial attacks will demonstrate the effectiveness and generalization ability of our defense framework.

1 Introduction

Many real-world datasets are usually modeled with Heterogeneous Graphs (HGs) (Shi et al. 2017), which contain diverse types of objects and relations. An example of ACM citation network characterized by HG is given in Figure 1(a), consisting of three types of objects (Author (A), Paper (P), Subject (S)), and two types of relations (P-A and P-S). Since HGs contain rich high-order structural information, metapath (sequence of relation types between two node types) is widely used as a basic tool to capture such information (Shi et al. 2017), such as P-A-P (papers written by the same author) and P-S-P (papers attached to the same subject). In recent years, with deep learning employed on HGs, there is a surge of Heterogeneous Graph Neural Networks (HGNNs) (Wang et al. 2019b; Yun et al. 2019; Fu et al. 2020), which often adopt a hierarchical aggregation (including node-level and semantic-level) to capture the information from metapath-based neighbors, and have achieved state-of-the-art performance on a wide range of tasks, e.g., node classification and link prediction.

Despite the great success of HGNNs, there is no systematic understanding of the adversarial robustness of HGNNs, i.e., *whether the HGNNs can be easily fooled by slight perturbations of the input topology*. This is especially important for HGNN models since they are widely applied to many real-world applications, e.g., e-commerce (Zhang et al. 2019a; Hu et al. 2019) and cyber security (Zhang et al. 2019c; Zhong et al. 2020). So far, most works focus on adversarial vulnerabilities of homogeneous GNNs (Sun et al. 2018), but the robustness for HGNNs is indeed not foreseeable due to unique metapath-based aggregation.

To answer this question, we introduce the first study of adversarial robustness of HGNNs through evaluating their performance on dataset ACM under the same evasion adversarial attacks [1], which perturb topology in the test phase, and the attack results are shown in Figure 1(b). Surprisingly, compared to the drop of GCN by about 3 points, HGNNs, i.e., HAN, MAGNN, and GTN, dramatically decrease by an average of 28 points. Obviously, HGNNs have significantly different adversarial robustness from GCNs, which motivates us to further investigate the differences of architectures between GCNs and HGNNs.

In the further analysis of attack results, we observe that the attackers tend to maliciously link the target node to the large-degree node (i.e., hub). Taking HAN as an example, the attacker injects an adversarial edge (p_1, a_3) in Figure 1(a), which will lead malicious (red) papers $p_4 \cdots p_{66}$ to be the direct neighbors of p_1 under metapath PAP. And even they are assigned small attention values, they can still dominate the receptive field of HAN in Figure 1(c). We argue such vulnerabilities of HGNNs can be attributable to two key reasons: (1) **Perturbation enlargement effect**.

*Corresponding author

[1]Attack algorithm and the implementation details of the compared GCN (Kipf and Welling 2017) can be found in Appendix A.

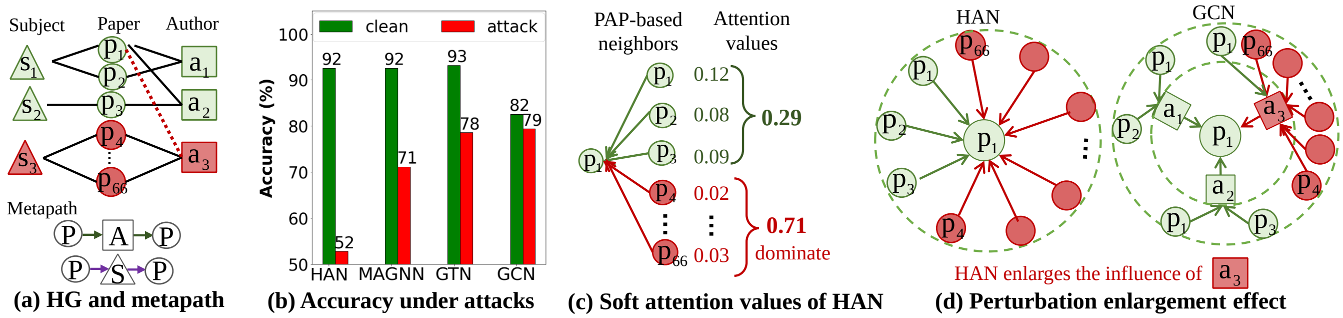


Figure 1: The illustrative example for adversarial attack against HGNNs on ACM dataset. (a) Basic concepts of HG (HG, metapath). (b) The a robustness evaluation of HGNNs and GCN. (c) A toy example of the soft attention values of HAN under adversarial edge (p_1, a_3). (d) The comparison of the influence of adversarial link (p_1, a_3) to HAN and GCN.

We will prove that HGNNs will rapidly enlarge the effect of the adversarial hub. While GCNs will not enlarge it, since compared to HGNNs, as shown in Figure 1(d), GCNs will not regard the malicious $p_4 \cdots p_{66}$ as the direct neighbors of p_1, and thus the malicious two-hop neighbors $p_4 \cdots p_{66}$ can only influence p_1 through one-hop neighbor a_3. However, HAN directly aggregates all neighbors under PAP with equal weights $\frac{1}{66}$, and thus enlarges the effect of adversarial (p_1, a_3) to $\frac{63}{66}$ (i.e., the total weights of malicious $p_4 \cdots p_{66}$). (2) **Soft attention mechanism**. Conventional attention mechanisms assume all neighbors are reliable and aggregate them with soft (i.e., positive) values. This soft attention mechanism may hurt the performance of GNNs when existing adversarial/noisy/disassortative neighbors (Zhang and Zitnik 2020; Bo et al. 2021). It will cause more serious damage to HGNN when injecting adversarial hub as shown in Figure 1 (c).

Once the vulnerabilities of HGNNs are identified, there is a strong need for further improving the adversarial robustness of HGNNs. Thus, in this paper, we propose a **Ro**bust **He**terogeneous GNN framework (**RoHe**) against topology adversarial attacks by designing an attention purifier, which can prune malicious neighbors based on topology and feature. More specifically, for the problem of perturbation enlargement, we introduce the metapath-based transiting probability as the prior criterion of the purifier, restraining the confidence of malicious neighbors from the adversarial hub. Then the purifier learns a differentiable mask vector to remove the unreliable neighbors in the soft attention mechanism.

The contributions of this work are three folds:

- We introduce the first systematic exploration and assessment of the robustness of HGNNs, and point out that the HGNNs are highly fragile to adversarial link to the hub, which can be attributed to the problems of perturbation enlargement effect and soft attention mechanism.
- Based on the above findings, we propose a novel robust HGNN framework (RoHe) against adversarial attacks by designing an attention purifier, which can constrain the enlargement perturbations by transiting probability and eliminate the negative impact of malicious neighbors through mask operation.
- We perform experiments on different benchmark datasets for multiple HGNNs. The effectiveness and generalization ability of our defense framework RoHe are well demonstrated by the considerable improvement of HGNNs under adversarial attacks.

2 Preliminaries

Definition 1 ***Heterogeneous Graph.*** *A heterogeneous graph, defined as $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, consists of an object set $\mathcal{V}$ and an edge set $\mathcal{E}$. $\mathcal{G}$ is also associated with a node type mapping function $\phi : \mathcal{V} \to \mathcal{A}$ and an edge type mapping function $\psi : \mathcal{E} \to \mathcal{R}$. $\mathcal{A}$ and $\mathcal{R}$ denote the predefined sets of node types and edge types, where $|\mathcal{A}| + |\mathcal{R}| > 2$. For each type $R \in \mathcal{R}$, $\mathbf{M}^R$ represents the corresponding binary adjacency matrix.*

Definition 2 ***Metapath.*** *A metapath Φ is defined as a path in the form of $\Phi = A_1 \xrightarrow{R_1} A_2 \xrightarrow{R_2} \cdots \xrightarrow{R_l} A_{l+1}$, which describes a composite relation $R = R_1 \odot R_2 \odot \cdots \odot R_l$ between node types A_1 and A_{l+1}.*

Definition 3 ***Metapath based Neighbors.*** *Given a node v and a metapath Φ in a heterogeneous graph, the metapath based neighbors $\mathcal{N}_v^\Phi$ are defined as the set of nodes that connect with v via metapath Φ.*

Metapath-based transiting probability. In this paper, we consider the metapath-based transiting probability denoted by $\mathbf{P}^\Phi_{vu}$ (from node v to neighbor u along metapath $\Phi = A_1 \xrightarrow{R_1} A_2 \xrightarrow{R_2} \cdots \xrightarrow{R_l} A_{l+1}$), which can be used to guide a metapath-based random walk for learning heterogeneous graph embedding, e.g., metapath2vec (Dong, Chawla, and Swami 2017). And the matrix $\mathbf{P}^\Phi$ can be calculated by

$$\mathbf{P}^\Phi = \mathbf{P}^{R_1} \cdots \mathbf{P}^{R_l}, \tag{1}$$

where $\mathbf{P}^{R_i} = (\mathbf{D}^{R_i})^{-1}\mathbf{M}^{R_i}$ for $i \in \{1 \cdots l\}$. This shows that given a metapath Φ, $\mathbf{P}^\Phi_{vu}$ is defined in terms of two parts: (1) their connectivity defined by the number of paths between v and u following Φ; and (2) the degree information of all nodes along paths.

Heterogeneous graph neural networks. HGNNs often adopt a hierarchical aggregation: the node-level one aims to merge the neighbors based on a specific metapath, and the semantic-level one can fuse the information of different metapaths. In this paper, we focus on three representative HGNNs, i.e., HAN (Wang et al. 2019b), MAGNN (Fu et al. 2020) and GTN (Yun et al. 2019). Taking HAN as an example, the metapath-based embedding of node v can be aggregated as follows:

$$\mathbf{z}_v^\Phi = \sigma(\sum_{u \in \mathcal{N}_v^\Phi} a_{vu}^\Phi \cdot \mathbf{h}_u), \tag{2}$$

where a_{vu}^Φ is the attention value for neighbor u, $\mathbf{h}_u$ is the projected feature of u, $\mathcal{N}_v^\Phi$ is the metapath-based neighbors.

To facility analysis, we provide more preliminaries about the mechanism of (asymmetrically normalized) GCN and MAGNN/GTN in Appendix. For clarity, we also formally provide the simplified structure-based weights of $u \in \mathcal{N}_v^\Phi$ for these models as shown in Table 1, by giving metapath $\Phi = A_1 \xrightarrow{R_1} A_2 \xrightarrow{R_2} A_3$ and supposing the features of nodes are the same.

Adversarial attacks on GNNs. In this paper, we focus on evasion attack, a typical type of adversarial attack that perturbs graph in the test phase and guides the model to misclassify the target node v. Specifically, given a homogeneous graph with adjacency matrix $\mathbf{M}$ and node features $\mathbf{X}$, the goal of an attacker is to find the optimal perturbed adjacency matrix $\mathbf{M}_\Delta$:

$$\underset{\mathbf{M}_\Delta}{\operatorname{argmax}} \mathcal{L}(f_{GNN}^*(\mathbf{M}_\Delta, \mathbf{X})_v, c_v), \tag{3}$$

where $f_{GNN}^*(\mathbf{M}_\Delta, \mathbf{X})_v$ is the prediction of trained GNN model f_{GNN}^* for node v, c_v is the label of v, Δ (named budget) is the maximum number of the perturbed edges, and $\mathcal{L}$ is the classification loss in this paper. The yielding optimal $\mathbf{M}_\Delta$ will lead to minimum test accuracy.

3 Adversarial Vulnerability Analysis

We perform adversarial attacks on HGNNs and GCN (details of attack method are in Appendix), and the results presented in Figure 1 (b) clearly show that compared to GCN, HGNNs are highly vulnerable to adversarial attacks, especially HAN. Here we further analyze the key reasons for such vulnerabilities.

3.1 Perturbation Enlargement Effect

We discover that HGNNs exist the phenomenon of perturbation enlargement, i.e., HGNNs will enlarge the effect of the adversarial hub. As shown in Figure 1 (d), the influence of adversarial hub a_3, expected to be less than $\frac{1}{3}$ (the inverse of the p_1's author neighbors), is enlarged to $\frac{63}{66}$ for HAN. Specifically, when the attacker injects an adversarial hub a_3 as the direct neighbor of p_1, the influence of a_3 to p_1 should be proportional to the inverse of the degree of target node p_1 (i.e., $\frac{1}{3}$) from the perspective of network science (i.e., transiting probability). While HGNNs can not satisfy it and enlarge the total weights to $\frac{63}{66}$ for HAN and $\frac{63}{68}$ for MAGNN/GTN, since they skip the intermediate layers (e.g., the layer for author in PAP), and directly aggregate multi-hop neighbors $p_1 \cdots p_{66}$, failing to encode transiting probability $\mathbf{P}^{R_1}\mathbf{P}^{R_2}$ in structural weight of u.

We also find that the perturbation enlargement effect is more significant in HAN than MAGNN and GTN. Taking Figure 1(d) as an example, we can see that p_1 is connected to p_1 more densely (by 3 paths) than p_4 (by 1 path). Thus the p_1 is expected to have larger weights than p_4 in transiting probability. MAGNN and GTN can satisfy it and assign higher weight to itself p_1 ($\frac{3}{68}$) than malicious p_4 ($\frac{1}{68}$), by encoding the total number of path instances (i.e., $(\mathbf{D}_v^\Phi)^{-1}\mathbf{M}_{vu}^\Phi$). While HAN equally treats all neighbors with same weights $\frac{1}{66}$, thus yields the larger total weights of malicious $p_4 \cdots p_{66}$ ($\frac{63}{66}$) than MAGNN/GTN ($\frac{63}{68}$).

Model	Weight of u	$\mathbf{p_1}$	$\mathbf{p_2/p_3}$	$\mathbf{p_{4\text{-}66}}$
TransP	$(\mathbf{P}^{R_1}\mathbf{P}^{R_2})_{vu}$	$\frac{2}{6} + \frac{1}{3\times64}$	$\frac{1}{6}$	$\frac{1}{3\times64}$
HAN	$\frac{1}{\lvert\mathcal{N}_v^\Phi\rvert}$	$\frac{1}{66}$	$\frac{1}{66}$	$\frac{1}{66}$
MAGNN	$(\mathbf{D}_v^\Phi)^{-1}\mathbf{M}_{vu}^\Phi$	$\frac{3}{68}$	$\frac{1}{68}$	$\frac{1}{68}$
GTN	$(\mathbf{D}_v^\Phi)^{-1}\mathbf{M}_{vu}^\Phi$	$\frac{3}{68}$	$\frac{1}{68}$	$\frac{1}{68}$

Table 1: The structural weights of $u \in \mathcal{N}_v^\Phi$ and their examples in HGNNs, TransP (short for Transiting Probability). Here $\mathbf{P}^{R_i} = (\mathbf{D}^{R_i})^{-1}\mathbf{M}^{R_i}$.

3.2 Soft Attention Mechanism

We argue the soft attention mechanism will especially hurt the generalization performance on adversarial attacks for HGNNs. As shown in Figure 1 (c), vast malicious neighbors $p_4 \cdots p_{66}$ can accumulate the smaller but positive attention values and finally dominate the receptive field of HGNNs, misleading the classification of p_1. Based on this fact, the power of assigning zero attention values to obviously unreliable neighbors is significant for HGNNs.

4 The Proposed Robust Heterogeneous GNN

This section depicts our proposed **Ro**bust **He**terogeneous GNNs (**RoHe**) against topology adversarial attacks. HGNNs often adopt a hierarchical aggregation (including node-level and semantic-level), and our RoHe is applied to purify the node-level aggregation. Figure 2 illustrates the overall architecture of RoHe. The node-level attention for metapath-based neighbors will be equipped by our purifier, which can constrain the enlargement perturbations by transiting prior and eliminate the negative impact of malicious neighbors through mask operation. Then the purified attention will be used for node-level aggregation, yielding the node embeddings for different metapath, which can be fused in semantic-level aggregation. Note that here we present our general framework based on HAN (Wang et al. 2019b) for simplicity.

4.1 Node-level Aggregation

Here we first detail the node-level attention mechanism and show that our purifier can eliminate the problems of perturbation enlargement and soft attention mechanism by transiting probability and purification mask.

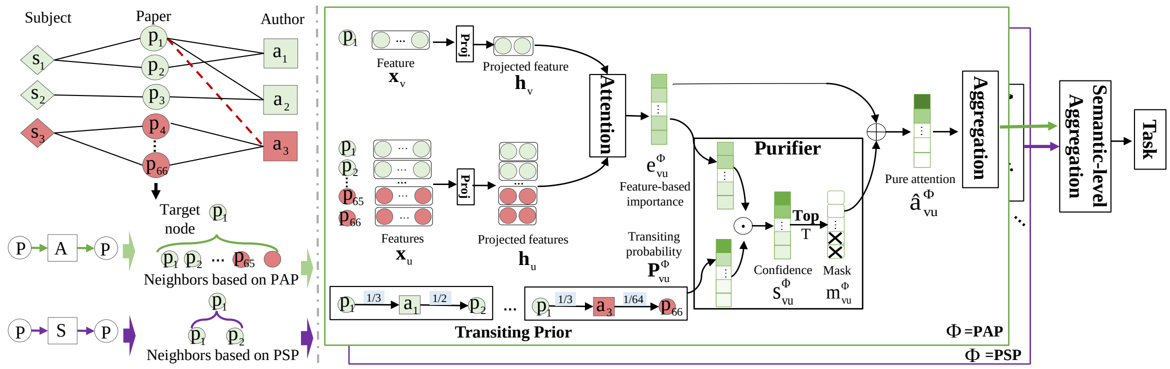


Figure 2: The overall framework of RoHe.

Node feature transformation. Since different node types may have unequal dimensions of feature vectors or lie in different feature spaces, HGNNs usually project the features of different types of nodes into the common space. Specifically, for the target node v with type $A \in \mathcal{A}$, we use a type-specific transformation matrix $\mathbf{W}_A$ to obtain the projected features $\mathbf{h}_v$ as follows:

$$\mathbf{h}_v = \mathbf{W}_A \mathbf{x}_v. \tag{4}$$

Feature-based importance. Given a metapath Φ, based on the hypothesis that the nodes with similar features are more likely to be important than dissimilar ones, we estimate the importance e^{Φ}_{vu} of neighbors u to target node v under Φ by dot-product similarity of features:

$$e^{\Phi}_{vu} = \mathbf{h}_v \cdot \mathbf{h}_u. \tag{5}$$

In conventional node-level attention mechanism, the feature-based importance e^{Φ}_{vu} will be directly normalized across $\mathcal{N}^{\Phi}_v$ with the softmax function, yielding the final soft attention values a^{Φ}_{vu}. We argue a^{Φ}_{vu} only considers the feature information of nodes, while equally treats the multi-hop neighbors $\mathcal{N}^{\Phi}_v$ from the perspective of topology, which will lead to enlarging the effect of adversarial hub neighbor as proved in Section 3.1. Besides, all neighbors in $\mathcal{N}^{\Phi}_v$ are assigned positive values after softmax function. Such soft attention mechanism has excellent differentiability in back propagation (Chaudhari et al. 2019), but fails to assign zero value to obviously malicious neighbors as showed in Section 3.2.

To solve above problems, we introduce a differentiable purifier to mask out the neighbor $u \in \mathcal{N}^{\Phi}_v$ with low confidence score s^{Φ}_{vu}. Specifically, we first utilize metapath-based transiting probability $\mathbf{P}^{\Phi}_{vu}$ as the prior of confidence of u to eliminate perturbation enlargement problem.

Transiting prior. Given metapath $\Phi = A_1 \xrightarrow{R_1} A_2 \xrightarrow{R_2} \cdots \xrightarrow{R_l} A_{l+1}$, to encode the probability of transiting along metapath Φ as a prior, we first calculate the transiting probability matrix $\mathbf{P}^{R_i} = (\mathbf{D}^{R_i})^{-1}\mathbf{M}^{R_i}$ for relation $R_i \in \{R_1, \cdots, R_l\}$. Each element $\mathbf{P}^{R_i}_{vu}$ represents the probability of transiting from node v to u in relation R_i. And powering it along Φ will lead to the metapath-based transiting probability $\mathbf{P}^{\Phi}$ as introduced in Section 2. Then we use the element $\mathbf{P}^{\Phi}_{vu}$ as the prior confidence of neighbor u for target node v in metapath Φ. We can see that the neighbor u is expected to obtain a small $\mathbf{P}^{\Phi}_{vu}$ for confidence, if u is indirectly connected to v passing through the hub node, which can solve the enlargement of adversarial hub as described in Section 3.1.

Confidence score. Based on the transiting prior $\mathbf{P}^{\Phi}_{vu}$, to determine the unreliable neighbors, we can calculate the confidence score vector $\mathbf{s}^{\Phi}_v \in \mathbb{R}^{|\mathcal{N}^{\Phi}_v|}$ for neighbors $\mathcal{N}^{\Phi}_v$ based on both feature and topology, by incorporating feature similarity e^{Φ}_{vu} and $\mathbf{P}^{\Phi}_{vu}$:

$$s^{\Phi}_{vu} = \sigma(\mathbf{P}^{\Phi}_{vu} \cdot e^{\Phi}_{vu}). \tag{6}$$

The notation s^{Φ}_{vu}, as an element of $\mathbf{s}^{\Phi}_v$, is the confidence score for neighbor $u \in \mathcal{N}^{\Phi}_v$, indicating that neighbors with similar features and high transiting probabilities are regarded to be reliable.

For the problem of soft attention, we design a mask operation, which can mask out neighbors with low confidence in a differentiable way.

Purification mask. We model the mask operation by constructing a mask vector $\mathbf{m}^{\Phi}_v \in \{1, -\infty\}^{|\mathcal{N}^{\Phi}_v|}$ for all the neighbors of target node v by

$$m^{\Phi}_{vu} = \begin{cases} 0 & \text{if } u \in \mathrm{Top}(\mathbf{s}^{\Phi}_v, T), \\ -\infty & otherwise, \end{cases} \tag{7}$$

where T is the number of neighbors to be kept, and $\mathrm{Top}(\cdot)$ returns the set of the T most reliable neighbors based on their confidence scores $\mathbf{s}^{\Phi}_v$. Then the other neighbors will be removed by setting their mask values as $-\infty$. When a softmax is applied to a sum of e^{Φ}_{vu} and $m^{\Phi}_{vu} = -\infty$, the node u will be effectively masked out, since the output of softmax for $-\infty$ is zero.

Thus, we can use $\mathbf{m}^{\Phi}_v$ to mask out the abundant adversarial/noisy neighbors, yielding purified attention $\hat{a}^{\Phi}_{vu}$ via softmax function:

$$\hat{a}^{\Phi}_{vu} = \frac{\exp(m^{\Phi}_{vu} + e^{\Phi}_{vu})}{\sum_{i \in \mathcal{N}^{\Phi}_v} \exp(m^{\Phi}_{vi} + e^{\Phi}_{vi})}. \tag{8}$$

In this way, node-level attention is enhanced to encode the transiting probability of metapath-based neighbors and only aggregate top-T reliable neighbors, alleviating the problems of perturbation enlargement and soft attention mechanism.

$$\hat{a}_{vu}^{\Phi} = \mathrm{softmax}_u(e_{vu}^{\Phi}) = \frac{\exp(e_{vu}^{\Phi})}{\sum_{i \in \mathcal{N}_v^{\Phi}} \exp(e_{vi}^{\Phi})}. \tag{9}$$

Aggregation of neighbors. Finally, the final purified attention $\hat{a}_{vu}^{\Phi}$ will be used to aggregate neighbors for semantic-specific embedding $\mathbf{z}_v^{\Phi}$ as

$$\mathbf{z}_v^{\Phi} = \sum_{u \in \mathcal{N}_v^{\Phi}} (\hat{a}_{vu}^{\Phi} \cdot \mathbf{h}_u). \tag{10}$$

4.2 Semantic-level Aggregation

Since different metapaths capture different semantics of the HG, HGNNs usually adopt semantic-level attention to calculate the importance of each metapath. Given the metapath set $\{\Phi_0, \Phi_1, \cdots, \Phi_P\}$, after node-level aggregation, we can obtain a group of semantic-specific node embeddings of v, denoted as $\{\mathbf{z}_v^{\Phi_0}, \mathbf{z}_v^{\Phi_1}, \cdots, \mathbf{z}_v^{\Phi_P}\}$. HAN further calculates the importance of metapath $\Phi \in \{\Phi_1, \cdots, \Phi_P\}$ by

$$w^{\Phi} = \frac{1}{|\mathcal{V}|} \sum_{v \in \mathcal{V}} \mathbf{q}^T \cdot tanh(\mathbf{W} \cdot \mathbf{z}_v^{\Phi} + \mathbf{b}), \tag{11}$$

where $\mathbf{W}$ and $\mathbf{b}$ denote the weight matrix and bias of the MLP, respectively. $\mathbf{q}$ is the semantic-level attention vector. Then HAN uses the softmax function to normalize the importance w^{Φ} to yield the attention value β^{Φ} for Φ. Hence, the final embedding $\mathbf{z}_v$ of v can be obtained by semantic-level aggregation:

$$\mathbf{z}_v = \sum_{\Phi \in \{\Phi_1, \cdots, \Phi_P\}} \beta^{\Phi} \cdot \mathbf{z}_v^{\Phi}. \tag{12}$$

Finally, the overall proposed model can be optimized by minimizing following loss:

$$\mathcal{L} = - \sum_{v \in \mathcal{V}_L} \ln(\mathbf{W}_{clf} \cdot \mathbf{z}_{v, c_v}), \tag{13}$$

where $\mathbf{W}_{clf}$ is the parameter of the classifier, c_v is the class of training node $v \in \mathcal{V}_L$. The overall process of our proposed RoHe is summarized in Algorithm 1.

5 Experiments

5.1 Experimental Setup

Datasets. RoHe is evaluated on three widely used HG datasets: (1) **ACM** (Wang et al. 2019a) consists of {Paper (P), Author (A), Subject (S)} and we employ metapath set {PAP, PSP} for paper classification. (2) **DBLP** (Fu et al. 2020) consists of {Author (A), Paper (P), Term (T), Conference (C)} and we use metapath set {APA, APCPA, APTPA} for author classification. (3) **Aminer** (Hu, Fang, and Shi 2019) consists of {Paper (P), Author (A), Reference (R)} and we employ metapaths set {PAP, PRP} for paper classification. Note that the features of ACM and DBLP are based on bag-of-words representations, and the features of Aminer are assigned one-hot id vectors to nodes. Details are in Appendix C.1.

Algorithm 1: RoHe: Robust heterogeneous HAN

Require: The heterogeneous graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$,
The node features $\{\mathbf{x}_v, v \in \mathcal{V}\}$,
The metapath set $\{\Phi_0, \Phi_1, \cdots, \Phi_P\}$,
The mask threshold T.
Ensure: The final node embeddings $\{\mathbf{z}_v, v \in \mathcal{V}\}$.
Pre-process transiting matrix via Eq. (1);
for node type $A \in \mathcal{A}$ **do**
  Type-specific transformation to obtain $\{\mathbf{h}_v, \in \mathcal{V}\}$;
end for
for $\Phi \in \{\Phi_0, \Phi_1, \cdots, \Phi_P\}$ **do**
  for $v \in \mathcal{V}$ **do**
    Find the metapath-based neighbors $\mathcal{N}_v^{\Phi}$
    for $u \in \mathcal{N}_v^{\Phi}$ **do**
      Calculate the feature-based importance e_{vu} for $u \in \mathcal{N}_v^{\Phi}$ via Eq. (5);
      Calculate confidence score via Eq. (6);
      Obtain purification mask vector $\mathbf{m}_v^{\Phi}$ via Eq. (7);
      Obtain the purified attention $\hat{a}_{vu}^{\Phi}$ via Eq. (9);
    end for
    Obtain the node embedding $\mathbf{z}_v^{\Phi}$ for Φ via Eq. (10);
  end for
end for
Calculate the semantic-level attention values $\{\beta^{\Phi}\}$ for $\Phi \in \{\Phi_0, \Phi_1, \cdots, \Phi_P\}$;
Obtain final node embeddings $\{\mathbf{z}_v, v \in \mathcal{V}\}$ by fusing the embeddings from different metapath via Eq. (12);

Setup. (1) **HGNNs:** We mainly evaluate the effectiveness of our RoHe on HAN, and we also generalize RoHe to MAGNN (Fu et al. 2020) and GTN (Yun et al. 2019). (2) **Baselines:** Since there are no existing robust HGNNs methods, we compare with the direct adaptations of following strategies: Jaccard (Wu et al. 2019), GGCL (Zhu et al. 2019) and SimP (Jin et al. 2021), and the variants of our RoHe: **RoHe$_\mathbf{T}$** (only keeping transiting probability) and **RoHe$_\mathbf{P}$** (only keeping pruning operation). (3) **Generating adversarial attack:** We employ FGSM-based attacks (Goodfellow, Shlens, and Szegedy 2015) to generate perturbation edges in experiments. Given a target node, we limit adversarial edges with budget $\Delta = \{1, 2, 3\}$ and edge types as P-A for ACM/DBLP and P-R for Aminer. We evaluate the performance with Micro-F1 metric over 500 target nodes, which are randomly sampled from the test set. More details about experimental settings are in Appendix C.2 and **??**.

5.2 Defense Effectiveness of RoHe

Here we evaluate the effectiveness of RoHe on HAN (i.e., HAN-RoHe) against all baselines, under two scenarios (Clean and Attack). The overall results are presented in Table 2, and results of more metrics are in Appendix **??**. Here we have the following observations:

(1) Attacker can dramatically decrease the performance of HAN by about 43% by adding one edge. However, the proposed HAN-RoHe successfully restores the performance of GNNs to the level comparable to when there is no attack. For example, with the increase of budget Δ, the per-

Data	Model	Clean	Attack		
			$\Delta=1$	$\Delta=3$	$\Delta=5$
ACM	HAN	0.926	0.528	0.330	0.240
	Jaccard	0.918	0.892	0.860	0.848
	SimP	0.898	0.746	0.476	0.358
	GGCL	0.902	0.260	0.084	0.084
	HAN-RoHe$_P$	0.924	0.780	0.868	0.870
	HAN-RoHe$_T$	**0.940**	0.900	0.564	0.304
	HAN-RoHe	0.920	**0.904**	**0.902**	**0.882**
DBLP	HAN	0.942	0.332	0.096	0.060
	Jaccard	0.934	0.816	0.812	0.802
	SimP	0.942	0.790	0.670	0.600
	GGCL	0.914	0.684	0.464	0.344
	HAN-RoHe$_P$	0.862	0.686	0.714	0.702
	HAN-RoHe$_T$	**0.944**	0.760	0.360	0.220
	HAN-RoHe	0.942	**0.936**	**0.864**	**0.808**
Aminer	HAN	**0.882**	0.346	0.134	0.102
	GGCL	0.808	0.276	0.056	0.042
	HAN-RoHe$_P$	0.840	0.772	0.772	0.774
	HAN-RoHe$_T$	0.842	0.788	0.668	0.562
	HAN-RoHe	0.838	**0.840**	**0.812**	**0.802**

Table 2: Results (Micro-F1) of HAN-RoHe. A higher value indicates better robustness.

formance of HAN-RoHe only drops by about 5% for ACM and Aminer. The reason is that HAN can greatly benefit from RoHe by equipping an attention purifier, which filters adversarial neighbors and retains the essential neighbors.

(2) The proposed HAN-RoHe consistently outperforms all defense methods in the Attack scenario. 1) Jaccard and SimP, pruning unreliable neighbors based on the feature similarity, will only alleviate the problem of soft attention mechanism and thus have limited improvement. 2) The Gaussian layer of GGCL also cannot completely absorb the vast adversarial neighbors, failing to defend against such attacks. 3) HAN-RoHe$_T$ and HAN-RoHe$_P$ only solve one of the problems of HGNNs respectively, and thus fail to achieve best adversarial robustness. In summary, the above observations prove the reasons for the adversarial vulnerabilities of HGNNs.

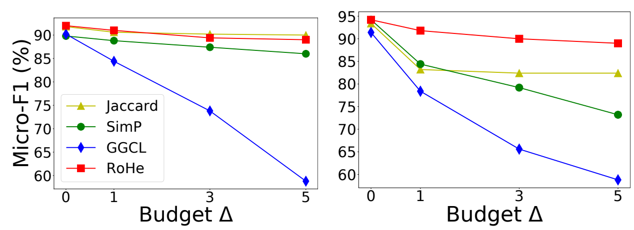
(3) HAN-RoHe also successfully defends the non-attributed HG Aminer. The defense model SimP and Jaccard, relying on the original feature (i.e., attribute), hence cannot be directly applied to Aminer. While our RoHe relies on node embedding rather than original features, thus can still enhance the robustness of HAN on Aminer.

5.3 Generalization Performance of RoHe

Generalization performance on random noise. We evaluate the robustness of the proposed RoHe under random noise by linking the target node to random nodes. The results are shown in Figure 3. Results of more metrics and datasets are in Appendix **??**. We can see HAN-RoHe achieves the best performance on most metrics and its performance only slightly drops with the increase of budget Δ, meanwhile, Jaccard and SimP can also achieve comparable performance in comparison with RoHe on some metrics. The reason is

Data	HGNNs	Clean	Attack		
			$\Delta=1$	$\Delta=3$	$\Delta=5$
ACM	HAN	**0.926**	0.528	0.330	0.240
	HAN-RoHe	0.920	**0.904**	**0.902**	**0.882**
	MAGNN	**0.926**	0.711	0.647	0.589
	MAGNN-RoHe	0.916	**0.901**	**0.907**	**0.909**
	GTN	0.932	0.786	0.466	0.302
	GTN-RoHe$_T$	**0.932**	**0.892**	**0.772**	**0.656**
DBLP	HAN	0.942	0.332	0.096	0.060
	HAN-RoHe	**0.942**	**0.936**	**0.864**	**0.808**
	MAGNN	**0.920**	0.620	0.494	0.416
	MAGNN-RoHe	0.898	**0.798**	**0.740**	**0.682**
	GTN	0.946	0.564	0.200	0.128
	GTN-RoHe$_T$	**0.950**	**0.644**	**0.334**	**0.172**

Table 3: Results (Micro-F1) of RoHe on different HGNNs.



(a) ACM (b) DBLP

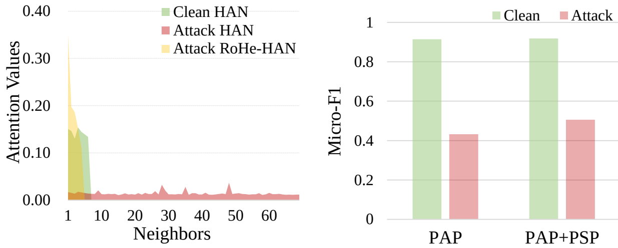
Figure 3: Results of HAN-RoHe under random noise.

that these noise neighbors possibly have different features with the target node, and can be filtered well by the defense models based on feature similarity.

Generalization performance on different HGNNs. To demonstrate that our proposed defense framework is generic to other HGNNs, we generalize RoHe to MAGNN and GTN. The results are presented in Table **??**, and results of more metrics are in Appendix **??**. We first observe that the performance of all HGNNs dramatically drops under adversarial attacks, which demonstrates their common limitations. And RoHe can significantly improve the robustness of diverse HGNNs, especially for HAN and MAGNN. The reason is that the memory-consuming GTN can only be equipped by variant RoHe$_T$, yielding limited improvement. Additionally, GTN and MAGNN show better robustness than HAN, since they can better encode structural information as explained in Section 3.1. We also find that all HGNNs are more vulnerable on DBLP than ACM, since the perturbations on P-A can be relieved by metapath PSP in ACM, which can be shown in Figure 4. But all metapaths in DBLP (APA, APCPA and APTPA) contain the perturbed relation type P-A, leading to less robustness.

5.4 Robustness of Aggregations

Analysis of node-level aggregation. To verify whether RoHe can learn robust node-level attention values, here we take a paper node P3143 about "Wireless Communication" in ACM as an example. For clean data (Clean), the P3143 is connected to 6 neighbors with PAP metapath. For perturbed data (Attack), the attacker just injects one perturbation edge



(a) Node-level (b) Semantic-level

Figure 4: Analysis of node/semantic-level aggregations. (a) Node-level attention values of HAN(-RoHe) for paper P3143 in ACM under Clean/Attack. (b) Results of different metapaths in ACM under Clean/Attack (only attacking P-A edges).

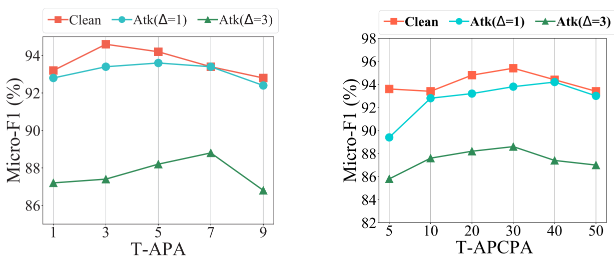


Figure 5: Analysis (Micro-F1) of parameter T.

by linking P3143 with author "Jiawei Han" who published 63 papers mainly about "Data Mining". And the node-level attention values are shown in Figure 4 (a). Obviously, under attack, the original soft attention mechanism of HAN has to aggregate the 63 adversarial neighbors with positive values, leading to the distortion of P3143 embedding. While RoHe successfully filters these perturbations and assigns high confidence scores for true neighbors.

Analysis of semantic-level aggregation. To evaluate whether the rich semantics conveyed by metapaths can enhance the adversarial robustness of HGNNs by semantic-level aggregation, as shown in Figure 4 (b), we report the performance of different metapath sets in ACM dataset under Clean and Attack, where the type of adversarial edges are constrained to P-A. Obviously, the HAN under full metapaths achieves better robustness, since the perturbations on P-A can be relieved by the information within P-S for ACM.

5.5 Parameter Study

We analyze hyper-parameters T which is the number of neighbors to be kept in purifier of HAN-RoHe, under scenarios of Clean and Attack with $\Delta = \{1, 3\}$. Here, we take metapaths APA and APCPA in DBLP dataset as examples. Figure 5 demonstrates how performance responds when threshold T increases. There exists an optimal T that delivers the best performance. When T is small, RoHe can only make use of little relevant neighbor information, which leads to inferior performance. When T increases, the purified receptive field involves more noise, leading to a higher chance of incorporating harmful neighbors, which negatively impacts the classification performance.

6 Related Work

Heterogeneous graph neural networks. Recently, HGNNs showed outstanding performance in various tasks. Roughly speaking, HGNNs fell into two categories: (1) Directly aggregating metapath-based neighbors. HAN (Wang et al. 2019b) proposed directly aggregated metapath-based neighbors with node-level attention. Then MAGNN (Fu et al. 2020) extended HAN by considering the intermediate nodes along metapath. GTN (Yun et al. 2019) further automatically identified the useful metapaths in the process of learning node embeddings. (2) Indirectly aggregating multi-hop neighbors. HGT (Hu et al. 2020) and R-GCN (Schlichtkrull et al. 2018) indirectly incorporated long-range neighbors through message passing across layers. Here we focus on the former type, which is widely used in many safety-related tasks (Hu et al. 2019; Zhong et al. 2020; Zhang et al. 2019c).

Adversarial robust on graphs. Recently, a magnitude of adversarial attacks were introduced for homogeneous graphs (Zügner, Akbarnejad, and Günnemann 2018; Li et al. 2020; Ma, Ding, and Mei 2020; Sun et al. 2018), pointing out their sensitivity regarding such attacks. However, there are few existing investigations on the adversarial attacks for HGs (Hou et al. 2019; Pezeshkpour, Tian, and Singh 2019; Zhang et al. 2019b), and they all focused on non-GNN based methods (e.g., metapath2vec (Hou et al. 2019)). This paper sheds the first light on this important problem. On the other side, these adversarial attacks works also triggered the research on adversarial defenses on GNNs (Wu et al. 2019; Jin et al. 2020). With a unique aggregation mechanism, HGNNs show different adversarial vulnerabilities from GCNs and need additional specially designed defense solutions.

7 Conclusion

In this paper, we introduce the first study on the adversarial robustness of HGNNs. Our extensive experiments show that HGNNs are highly fragile to topology adversarial attacks in comparison with GCNs, which can be attributed to the facts of perturbation enlargement and soft attention values. To address them, we propose an effective robust HGNN framework RoHe by equipping an attention purifier, which can prune unreliable neighbors based on topology and feature, alleviating the above vulnerabilities of HGNNs. Experiments on various datasets and multiple HGNNs show the effectiveness of RoHe. In future work, we will explore how to make full use of multiple aspects of information based on metapath to further improve robustness.

8 Acknowledgments

This work is supported in part by the National Natural Science Foundation of China (No. U20B2045, 62172052, 61772082, 61702296, 62002029), the Fundamental Research Funds for the Central Universities 2021RC28, and BUPT Excellent Ph.D. Students Foundation (No. CX2021202). This work is also sponsored by CCF-Ant Group Research Fund.

References

Bo, D.; Wang, X.; Shi, C.; and Shen, H. 2021. Beyond Low-frequency Information in Graph Convolutional Networks. 3950–3957. AAAI Press.

Chaudhari, S.; Polatkan, G.; Ramanath, R.; and Mithal, V. 2019. An Attentive Survey of Attention Models. *CoRR*, abs/1904.02874.

Dong, Y.; Chawla, N. V.; and Swami, A. 2017. metapath2vec: Scalable Representation Learning for Heterogeneous Networks. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Halifax, NS, Canada, August 13 - 17, 2017*, 135–144. ACM.

Fu, X.; Zhang, J.; Meng, Z.; and King, I. 2020. MAGNN: Metapath Aggregated Graph Neural Network for Heterogeneous Graph Embedding. In *WWW*, 2331–2341.

Goodfellow, I. J.; Shlens, J.; and Szegedy, C. 2015. Explaining and Harnessing Adversarial Examples. In *ICLR*.

Hou, S.; Fan, Y.; Zhang, Y.; Ye, Y.; Lei, J.; Wan, W.; Wang, J.; Xiong, Q.; and Shao, F. 2019. α*Cyber*: Enhancing Robustness of Android Malware Detection System against Adversarial Attacks on Heterogeneous Graph based Model. In *CIKM*, 609–618.

Hu, B.; Fang, Y.; and Shi, C. 2019. Adversarial Learning on Heterogeneous Information Networks. In *KDD*, 120–129.

Hu, B.; Zhang, Z.; Shi, C.; Zhou, J.; Li, X.; and Qi, Y. 2019. Cash-Out User Detection Based on Attributed Heterogeneous Information Network with a Hierarchical Attention Mechanism. In *AAAI*, 946–953.

Hu, Z.; Dong, Y.; Wang, K.; and Sun, Y. 2020. Heterogeneous Graph Transformer. In *WWW*, 2704–2710.

Jin, W.; Derr, T.; Wang, Y.; Ma, Y.; Liu, Z.; and Tang, J. 2021. Node Similarity Preserving Graph Convolutional Networks. In *WSDM*.

Jin, W.; Li, Y.; Xu, H.; Wang, Y.; and Tang, J. 2020. Adversarial Attacks and Defenses on Graphs: A Review and Empirical Study. *CoRR*.

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

Li, J.; Zhang, H.; Han, Z.; Rong, Y.; Cheng, H.; and Huang, J. 2020. Adversarial Attack on Community Detection by Hiding Individuals. In *WWW*, 917–927.

Ma, J.; Ding, S.; and Mei, Q. 2020. Towards More Practical Adversarial Attacks on Graph Neural Networks. In *NeurIPS*.

Pezeshkpour, P.; Tian, Y.; and Singh, S. 2019. Investigating Robustness and Interpretability of Link Prediction via Adversarial Modifications. In *NAACL-HLT*, 3336–3347.

Schlichtkrull, M. S.; Kipf, T. N.; Bloem, P.; van den Berg, R.; Titov, I.; and Welling, M. 2018. Modeling Relational Data with Graph Convolutional Networks. In *ESWC*, 593–607.

Shi, C.; Li, Y.; Zhang, J.; Sun, Y.; and Yu, P. S. 2017. A Survey of Heterogeneous Information Network Analysis. *IEEE Trans. Knowl. Data Eng.*, 29(1): 17–37.

Sun, L.; Dou, Y.; Yang, C.; Wang, J.; Yu, P. S.; and Li, B. 2018. Adversarial Attack and Defense on Graph Data: A Survey. *CoRR*.

Wang, M.; Zheng, D.; Ye, Z.; Gan, Q.; Li, M.; Song, X.; Zhou, J.; Ma, C.; Yu, L.; Gai, Y.; Xiao, T.; He, T.; Karypis, G.; Li, J.; and Zhang, Z. 2019a. Deep Graph Library: A Graph-Centric, Highly-Performant Package for Graph Neural Networks. *CoRR*.

Wang, X.; Ji, H.; Shi, C.; Wang, B.; Ye, Y.; Cui, P.; and Yu, P. S. 2019b. Heterogeneous Graph Attention Network. In *WWW*, 2022–2032.

Wu, H.; Wang, C.; Tyshetskiy, Y.; Docherty, A.; Lu, K.; and Zhu, L. 2019. Adversarial Examples for Graph Data: Deep Insights into Attack and Defense. In *IJCAI*, 4816–4823.

Yun, S.; Jeong, M.; Kim, R.; Kang, J.; and Kim, H. J. 2019. Graph Transformer Networks. In *NeurIPS*, 11960–11970.

Zhang, C.; Song, D.; Huang, C.; Swami, A.; and Chawla, N. V. 2019a. Heterogeneous Graph Neural Network. In *KDD*, 793–803.

Zhang, H.; Zheng, T.; Gao, J.; Miao, C.; Su, L.; Li, Y.; and Ren, K. 2019b. Data Poisoning Attack against Knowledge Graph Embedding. In *IJCAI*, 4853–4859.

Zhang, X.; and Zitnik, M. 2020. GNNGuard: Defending Graph Neural Networks against Adversarial Attacks. In *NeurIPS*.

Zhang, Y.; Fan, Y.; Ye, Y.; Zhao, L.; and Shi, C. 2019c. Key Player Identification in Underground Forums over Attributed Heterogeneous Information Network Embedding Framework. In *CIKM*, 549–558.

Zhong, Q.; Liu, Y.; Ao, X.; Hu, B.; Feng, J.; Tang, J.; and He, Q. 2020. Financial Defaulter Detection on Online Credit Payment via Multi-view Attributed Heterogeneous Information Network. In *WWW*, 785–795.

Zhu, D.; Zhang, Z.; Cui, P.; and Zhu, W. 2019. Robust Graph Convolutional Networks Against Adversarial Attacks. In *KDD*, 1399–1407.

Zügner, D.; Akbarnejad, A.; and Günnemann, S. 2018. Adversarial Attacks on Neural Networks for Graph Data. In *KDD*, 2847–2856.

This supplementary material is organized as follows: Sec. A provides the detailed model description and algorithm of adversarial attack used in this paper; Sec. B presents the neighbor aggregation mechanisms of HGNNs and GCN, and compares them with transiting probability; Sec. C shows more experiments details including datasets, baselines, experiments settings and the additional experimental results on more metrics.

A Adversarial Attacks on Heterogeneous Graph

In this section, we will describe how we generate adversarial attacks for HGNNs.

Given a target node v and an HGNN model f_{HGNN}, we constrain the edge type of perturbations to type R_i. The adversarial edges aim to mislead the classification of the trained f^*_{HGNN} for node v in the test phase. We adopt the representative FGSM (Goodfellow, Shlens, and Szegedy 2015) as the adversarial attack method.

Training agent model. Specifically, considering the memory occupation and complexity of HGNNs, here we need to use a lighter graph convolutional model (e.g.,GCN f_{GCN}) as the agent of HGNN model f_{HGNN} to generate attacks. However, GCN, which is originally proposed for homogeneous graph, fails to capture the metapath-based neighbor information as HGNNs, thus cannot be directly applied here. To tackle this problem, we first extract homogeneous sub-graph (i.e., $\mathbf{M}^{\Phi}$) based on a specified metapath Φ via the multiplications:

$$\mathbf{M}^{\Phi} = \mathbf{1}_A(\mathbf{M}^{R_1}\mathbf{M}^{R_2}\cdots\mathbf{M}^{R_l}), \tag{14}$$

where $\mathbf{1}_A$ is an indicator function defined for obtaining binary adjacency matrix:

$$\mathbf{1}_A(x) = \begin{cases} 1 & \text{if} \quad x > 0, \\ 0 & \text{if} \quad x = 0. \end{cases} \tag{15}$$

Thus GCN model, with the extracted homogeneous subgraph $\mathbf{M}^{\Phi}$ as input, can directly aggregate metapath based neighbors, similar to HAN. And the generated adversarial attacks based on such GCN agent are expected to decrease the performance of HAN. Typically, we consider a two-layer GCN model as the agent and the output probability of GCN model f_{GCN} can be defined as following simple form:

$$f_{GCN}(\mathbf{M}^{\Phi}, \mathbf{X}) = \mathrm{Softmax}(\hat{\mathbf{M}}^{\Phi}\mathrm{ReLU}(\hat{\mathbf{M}}^{\Phi}\mathbf{X}\mathbf{W}^{(1)})\mathbf{W}^{(2)}), \tag{16}$$

where $\hat{\mathbf{M}}^{\Phi}$ is the normalized adjacency matrix, given by $\hat{\mathbf{M}}^{\Phi} = (\tilde{\mathbf{D}}^{\Phi})^{-\frac{1}{2}}\tilde{\mathbf{M}}^{\Phi}(\tilde{\mathbf{D}}^{\Phi})^{-\frac{1}{2}}$, $\tilde{\mathbf{D}}^{\Phi}_{ii} = \Sigma_j\tilde{\mathbf{M}}^{\Phi}_{ij}$ and $\tilde{\mathbf{M}}^{\Phi} = \mathbf{M}^{\Phi} + \mathbf{I}^{\Phi}$. Given node v and the label c_v from the training set V_L, model can be trained by loss function $\mathcal{L}$ of node classification, which is defined as:

$$\mathcal{L}(f_{GCN}(\mathbf{M}^{\Phi}_v, \mathbf{X}_v), c_v) = -\ln f_{GCN}(\mathbf{M}^{\Phi}_v, \mathbf{X}_v)_{v,c_v}. \tag{17}$$

Generating adversarial attacks. With the well-trained agent model f^*_{GCN}, the gradient matrix $\mathbf{G}^{R_i}$ w.r.t. specified adjacency matrix $\mathbf{M}^{\mathbf{R_i}}$ can be calculated as follows:

$$\mathbf{G}^{R_i} = \frac{\partial\mathcal{L}(f^*_{GCN}(\mathbf{M}^{\Phi}, \mathbf{X})_v, c_v)}{\partial\mathbf{M}^{\Phi}} \cdot \frac{\partial\mathbf{M}^{\Phi}}{\partial\mathbf{M}^{R_i}}. \tag{18}$$

Algorithm 2: Adversarial edges generator

Require: The heterogeneous graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$,
The metapath Φ,
The specified edge type R,
The target node v,
The maximum number of perturbations Δ.
Ensure: The perturbed adjacency matrix $\mathbf{M}^{R}_{(\Delta)}$.

```
Extract homogeneous sub-graph M^Φ via Eq. (14);
Set M^Φ_(0) = M^Φ;
Train agent model f_GCN on M^Φ_(0);
for i = 1 to Δ do
    Calculate G^R_(i−1) based on M^Φ_(i−1) via Eq. (18);
    Select node pair (v, u) of the maximum absolute gradient in G^R_(i−1) for target node v;
    if G^R_(i−1)[v, u] > 0 then
        Update M^R_(i) by adding edge (v, u);
    else
        Update M^R_(i) by removing edge (v, u);
    end if
    Update M^Φ_(i) with M^R_(i) via Eq. (14);
end for
Return M^R_(Δ);
```

Based on the intuition that the gradients characterize how edge changes affect the model output, the FGSM attack algorithm picks the maximum absolute value of $\mathbf{G}^{R_i}$ as the adversarial edge for $\mathbf{M}^{R_i}$. After repeating above process for Δ times, we can obtain the perturbed adjacency matrix $\mathbf{M}^{R_i}_{\Delta}$, the pseudo-code of whole process is given in Algorithm 2. It is worth noting that we only consider the case that attacker has no access to true labels of target nodes, which is common in practical attack scenarios. Alternatively, attacker uses the prediction of HAN on target node as a substitution of true label.

The compared GCN model for heterogeneous graph. In Figure 1, we compare the adversarial robustness of GCN and HGNNs under heterogeneous graphs. To adapt the heterogeneous node features for homogeneous GCN, we set these features as one-hot matrix. The input adjacency matrix contains all types of nodes and relations.

B Neighbor Aggregation Mechanisms

To facilitate the analysis of the vulnerabilities of HGNNs, here we compare the neighbor aggregation mechanisms of HGNNs and GCN, then analyze their relationships with the transiting probability. For clarity, here we give a two-hop metapath $\Phi = A_1 \xrightarrow{R_1} A_2 \xrightarrow{R_2} A_3$ as an example.

HAN (Wang et al. 2019b) can aggregate neighbors $u \in \mathcal{N}^{\Phi}_v$ of node v in node-level aggregation as:

$$\mathbf{z}^{\Phi}_v = \sigma(\sum_{u\in\mathcal{N}^{\Phi}_v} a^{\Phi}_{vu} \cdot \mathbf{h}_u), \tag{19}$$

where a^{Φ}_{vu} is the attention value for neighbor u, $\mathbf{h}_u$ is the projected feature of u, $\mathcal{N}^{\Phi}_v$ is the metapath-based neighbors.

Intuitively, the importance of u to v is totally determined by the attention value a_{vu}^{Φ}, which is calculated only based on node features and neglects graph structure information.

MAGNN (Fu et al. 2020) improves HAN by further considering the intermediate nodes along metapath:

$$\mathbf{z}_v^{\Phi} = \sigma(\sum_{\Phi(v,u), u \in \mathcal{N}_v^{\Phi}} a_{v\Phi(v,u)} \cdot \mathbf{h}_{\Phi(v,u)}), \quad (20)$$

where $\Phi(v, u)$ is the metapath instance (i.e., node sequence connecting v with metapath-based neighbor $u \in \mathcal{N}_v^{\Phi}$), $\mathbf{h}_{\Phi(v,u)}$ is the embedding of instance $\Phi(v, u)$ by encoding all nodes features along $\Phi(v, u)$, $a_{v\Phi(v,u)}$ is the importance of u to v based on the metapath instance $\Phi(v, u)$. Note that v and u may be connected by multiple metapath instances. Clearly, the total weight of neighbor u is proportional to the number of the metapath instances between v and u.

GTN (Yun et al. 2019) further improves HAN by learning optimal metapath but here we fix the leaned metapath to Φ for simplicity:

$$\mathbf{z}_v = \sigma(\sum_{u \in \mathcal{N}_v^{\Phi}} (\mathbf{D}_v^{\Phi})^{-1} \mathbf{M}_{vu}^{\Phi} \mathbf{h}_u), \quad (21)$$

where adjacency matrix $\mathbf{M}^{\Phi}$ is obtained by $\mathbf{M}^{\Phi} = \mathbf{M}^{R_1}\mathbf{M}^{R_2}$ and then normalized by its degree matrix $\mathbf{D}^{\Phi}$. As we can see, neighbor u will have a large weight for aggregation, if the number of paths between u and v (i.e., $\mathbf{M}_{vu}^{\Phi}$) is large.

GCN (Kipf and Welling 2017) is the most typical GNN model for homogeneous graph. To align with the above transiting probability, we use the asymmetrical $\mathbf{D}^{-1}\mathbf{M}$ normalization for adjacency $\mathbf{M}$ instead of the symmetrical one $\mathbf{D}^{-\frac{1}{2}}\mathbf{M}\mathbf{D}^{-\frac{1}{2}}$, and the obtained propagation rule of GCN for $l-$th layer is

$$\mathbf{X}^{(l+1)} = \sigma(\mathbf{D}^{-1}\mathbf{M}\mathbf{X}^{(l)}\mathbf{W}^{(l)}). \quad (22)$$

Then we further simplify a two-layer GCN by removing the non-linearities between GCN layers as (Jin et al. 2021):

$$\mathbf{X}^{(2)} = \mathrm{Softmax}(\mathbf{D}^{-1}\mathbf{M}(\mathbf{D}^{-1}\mathbf{M}\mathbf{X}^{(0)}\mathbf{W}^{(0)})\mathbf{W}^{(1)}). \quad (23)$$

Clearly, the total weight of 2-hop neighbor u is proportional to the number of paths between u and v, and also proportional to the inverse of degree information of all nodes along paths.

C Experimental Details

C.1 Datasets

RoHe is evaluated on three benchmark datasets: (1)**ACM:** We adopt a subset of ACM dataset extracted by (Wang et al. 2019a) and all the papers are divided into 3 classes. The features are the bag-of-words representations of paper keywords. (2)**DBLP:** We adopt a subset of DBLP extracted by (Fu et al. 2020) and the authors are divided into 4 classes. The features of authors are the bag-of-words representations of their published paper keywords. (3) **Aminer:** We further extract a subset of Aminer in (Hu, Fang, and Shi 2019) and the papers are divided into three classes. Following (Hu, Fang, and Shi 2019), we assign one-hot id vectors to nodes as their dummy input features. The dataset statistics are provided in Table 4.

These datasets can be found as the following URLs:

- ACM: https://github.com/dmlc/dgl
- DBLP: https://github.com/cynricfu/MAGNN
- Aminer: https://github.com/librahu/HIN-Datasets-for-Recommendation-and-Network-Embedding

C.2 Baselines

Since there are no existing robust HGNNs methods, we compare with following strategies: (1) **Jaccard** (Wu et al. 2019): It calculates the Jaccard Similarity with features between nodes and their metapath-based neighbors, then removes the edges with low similarity as the preprocessing of HAN. (2) **GGCL** (Zhu et al. 2019): The Gaussian-based Graph Convolution (GGCL), using variance-based attention mechanism to remedy the propagation of attacks, is especially designed for homogeneous GCN model, thus we need to transfer it to heterogeneous graph data. Here we directly adapt it by extracting homogeneous graph based on specific metapath as its input. (3) **SimP** (Jin et al. 2021): It can alleviate the adversarial edges by preserving feature similarity in training. While it is also especially designed for homogeneous GCN model, here we transfer it as GGCL. (4) **RoHe$_{\mathbf{T}}$**: It directly adopts transiting probability as node-level attention values, which aims to only handle the perturbation enlargement problem. (5) **RoHe$_{\mathbf{P}}$**: It adopts attention purifier for node-level attention without transiting prior, which aims to only handle the limitations of soft attention mechanism. Note that SimP and Jaccard, relying on node features, cannot be used for non-attributed Aminer dataset.

The publicly available implementations of used models can be found at the following URLs:

- HGNNs:
  - HAN: https://github.com/dmlc/dgl
  - MAGNN: https://github.com/cynricfu/MAGNN
  - GTN: https://github.com/seongjunyun/Graph_Transformer_Networks
- Attack model
  - FGSM: https://github.com/DSE-MSU/DeepRobust
- Defense model
  - GCCL/SimP: https://github.com/DSE-MSU/DeepRobust
  - Jaccard/RoHe/RoHe$_{\mathrm{T}}$/RoHe$_{\mathrm{P}}$ based on: https://github.com/dmlc/dgl

Table 4: Dataset statistics.

Dataset	Relations(A-B)	# of A	# of B	# of A-B	Feature	Training	Validation	Test	Metapaths
ACM	Paper-Author	4025	17431	13407	1903	808	401	2816	PAP
	Paper-Subject	4025	73	4025					PSP
DBLP	Paper-Author	14328	4057	19645	334	400	400	3257	APA
	Paper-Conf	14328	20	14328					APCPA
	Paper-Term	14328	8693	86506					APTPA
Aminer	Paper-Author	6564	13329	18007	-	1314	655	4595	PAP
	Paper-Reference	6564	35890	58831					PRP